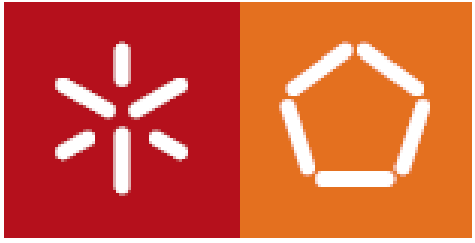

Universidade do Minho
Escola de Engenharia

Mickael Alexandre Silva Carvalho

Previsão em Tempo Real da Qualidade dos Efluentes de uma ETAR

Outubro de 2014

Universidade do Minho
Escola de Engenharia
Departamento de Informática

Mickael Alexandre Silva Carvalho

Previsão em Tempo Real da Qualidade dos Efluentes de uma ETAR

Dissertação de Mestrado

Mestrado em Engenharia Informática

Trabalho realizado sob orientação de

Professor Orlando Manuel de Oliveira Belo

Outubro de 2014

Agradecimentos

Antes de mais pretendo agradecer aos meus Pais, Irmão e Namorada, pelo suporte incondicional e pelo incentivo que demonstraram ao longo de toda a minha vida académica.

Aos meus amigos e colegas, pelos bons momentos passados juntos, pelo companheirismo, e pelo espírito de entreajuda, sem os quais este percurso académico não teria sido tão rico em boas recordações.

Aos professores do Departamento de Informática da Universidade do Minho que sempre se mostraram disponíveis para esclarecer qualquer tipo de dúvidas.

Em especial, Ao Professor Doutor Orlando Belo, um grande obrigado pela forma com cativa os alunos nas suas aulas. Pela orientação ao longo deste projeto, disponibilidade, e auxílio prestado nos momentos de dúvidas, e pelos conhecimentos transmitidos durante todo o Mestrado em Engenharia Informática que foram essenciais, tanto para a realização desta dissertação, como também o serão para um futuro profissional.

Outubro de 2014

Resumo

Uma análise ao desenvolvimento da sociedade, especialmente nas últimas décadas, permite verificar que é cada vez maior o número de informações geradas em todos os tipos de organizações. Esta quantidade de informação resulta um pouco da procura incessante pelo conhecimento. O surgimento das técnicas de *Data Mining* abriram novos horizontes nessa procura pelo conhecimento e permitem tornar uma organização mais competitiva e próspera. As técnicas de *Data Mining* permitem inúmeras atividades, desde a obtenção desse conhecimento, intrínseco e dificilmente obtido apenas com a observação dos dados, como também na monitorização e previsão de diversas situações nos processos envolvidos nas organizações. No contexto das ETAR, e no aperfeiçoamento do seu processo de tratamento, a utilização de técnicas de *Data Mining* revela-se uma atividade com bastante interesse. Atualmente uma das técnicas de *Data Mining* que mais tem chamado a atenção dos especialistas da área são as técnicas de *Support Vector Machines*, pela sua generalização e pelos resultados obtidos em trabalhos realizados no domínio. Num ambiente típico de uma ETAR são registados diariamente novos valores provenientes das diversas leituras realizadas por sensores de medição dos parâmetros físico-químicos, biológicos e microbiológicos das águas residuais. Estes sensores encontram-se situados ao longo das várias etapas do processo de tratamento das águas. Um dos parâmetros analisados e alvo de previsão neste projeto baseia-se na carência bioquímica de oxigénio, bastante importante para o processo de remoção de sólidos suspensos em ambientes de tratamento aeróbio e controlo do pH. Os constituintes dos efluentes que dão entrada diariamente numa ETAR possuem uma grande variabilidade em concentração e género. O surgimento diário de novos dados com uma grande variabilidade traz novas tendências e padrões que relacionam os diversos parâmetros das águas residuais. Neste projeto, procurou-se demonstrar que os modelos de previsão criados podem trazer um leque muito vasto de melhorias para o funcionamento de uma ETAR e, principalmente, para o seu processo de tratamento, monitorização e avaliação, aspetos muito importantes para a conservação do meio ambiente e da saúde pública. As ferramentas utilizadas para as várias tarefas de *Data Mining* que se realizaram foram o RapidMiner, o LIBLINEAR, e o TinySVM. Para tal, e seguindo a metodologia adotada, o CRISP-DM, a análise e a preparação dos dados foram fundamentais para a obtenção de resultados previsionais com alto índice de assertividade. Foram ainda utilizados métodos de avaliação para avaliar e comparar os modelos de previsão produzidos.

Palavras-chave: Descoberta de Conhecimento, CRISP-DM, *Data Mining*, Classificação, Regressão, Support Vector Machines, *Data Mining* Incremental, Estações de Tratamento de Águas Residuais, Carência Bioquímica de Oxigénio.

Abstract

An analysis of the development of society, especially in recent decades, shows that an increasing number of information is generated in all types of organizations. This amount of information is the result of the constant search for knowledge. The emergence of *Data Mining* techniques have opened new horizons in this quest for knowledge and the best method of making a more competitive organization and thrive. The *Data Mining* techniques allows numerous activities, from obtaining such knowledge, intrinsic and hardly obtained only with the observation of data, such as monitoring and forecasting various situations in the processes involved in organizations. In the context of the WWTP and of improvement of the treatment process, the use of *Data Mining* techniques proves to be an activity with great interest. Currently, one of *Data Mining* techniques that has most attracted the attention of specialists in the area are the Support Vector Machines techniques, by its capacity of generalization and by the obtained results in works done in the domain. In a typical environment of a WWTP are daily recorded new values from readings made by measuring sensors of the physical, chemical, biological, and microbiological parameters of the wastewater. These sensors are located throughout the various stages of the water treatment process. One of the analyzed parameters and target of prediction tasks of this project is based on the biochemical oxygen demand, fairly important to the process of removing suspended solids in the aerobic treatment and control of pH. The constituents of effluents that arrive daily in a WWTP have a large variability in concentration and gender. The daily emergence of new data with large variability brings new trends and patterns that relate the various parameters of wastewater. In this project, were sought to show that the forecast models created can bring many improvements to the overall operation of a wastewater treatment plant and especially for the treatment process and monitoring and evaluation important for the conservation of the environment and public health. The tools used for the various tasks of *Data Mining* that are performed were the RapidMiner, LIBLINEAR, and TinySVM. To that end, and following the methodology adopted, the CRISP-DM, the analysis and data preparation processes were essential for obtaining results of forecast with high assertiveness. In this project, were also used assessment methods to evaluate and compare the predictive produced models.

Keywords: Knowledge Discovery, CRISP-DM, *Data Mining*, Classification, Regression, Support Vector Machines, *Data Mining* Incremental, Wastewater Treatment Plants, Biochemical Oxygen Demand.

Indice

Indice de Figuras

Índice de tabelas

Siglas e Acrónimos

AM – Aprendizagem de Máquina

CBO - Carência Bioquímica de Oxigénio

CQO - Carência Química de Oxigénio

CRISP-DM - Cross Industry Standard Process for Data Mining

ETAR - Estações de Tratamento de Águas Residuais

KDD - Knowledge Discovery in Databases

KKT - Karush–Kuhn–Tucker

MAD – Erro Médio Absoluto (do Inglês – Mean Absolute Deviation)

MSE - Erro Quadratico Médio (do Inglês - Mean Squared Error)

RBF - Radial-Basis Function

RMSE - Raiz do Erro Quadrático Médio (do Inglês - Root Mean Squared Error)

SEMMA - Sample, Explore, Modify, Model, Assess

SGBD – Sistema de Gestão de Base de Dados

SP - Pontos de Amostragem (do Inglês – Sampling Points)

SS - Sólidos Suspensos

SSD - Sistemas de Suporte à Decisão

SSE - Soma do Erro Quadratico (do Inglês - Sum Squared Error)

SST - Sólidos Suspensos Totais

SSV - Sólidos Suspensos Voláteis

SVM - Máquinas de Vetores de Suporte (do Inglês – Support Vector Machines)

SVR - Support Vector Regression

VC - Vapnik e Chervonenkis

Capítulo 1

Introdução

1.1. Contextualização

Desde sempre, as organizações focaram-se em melhorar o seu funcionamento de forma a evoluir e tornarem-se mais competitivas. Um dos recursos fundamentais para essa evolução, e que sempre foi alvo de procura, é a obtenção e gestão do próprio conhecimento. Uma boa e eficiente gestão do conhecimento é uma caraterística sinónima de sucesso. Deste modo, a gestão do conhecimento surge como um recurso fundamental que vai além do processo de inovação, podendo determinar a vantagem competitiva de uma organização. A gestão do conhecimento é um dos principais objetivos das organizações com a finalidade de obter alguma vantagem competitiva no mercado inovando seus produtos e serviços.

Esta procura incessante pelo conhecimento leva as organizações a armazenar enormes quantidades de informação, que é acumulada durante vários anos nas suas bases de dados. Esta informação encontra-se na maior parte das vezes completamente desorganizada e sem grande potencial para utilização. Isto levou as organizações a focarem-se num outro recurso, os próprios sistemas tecnológicos. Os avanços tecnológicos, nomeadamente na área da administração de dados, têm-se focado num conceito relativamente recente capaz de solucionar o problema da informação inutilizável: a mineração de dados (*Data Mining*).

As técnicas de *Data Mining* têm superado as espectativas na obtenção de conhecimento e tiveram uma evolução enorme nas últimas décadas. O *Data Mining* é uma prática relativamente recente que utiliza técnicas de gestão de informação, inteligência artificial, reconhecimento de padrões e estatística para procurar correlações entre diferentes dados, permitindo adquirir conhecimento útil para promover a inovação e o bem-estar de uma organização. O *Data Mining* encontra-se envolvido num processo chamado descoberta de conhecimento em bases de dados (KDD –*Knowledge discovery in Databases*). Uma das caraterísticas fundamentais do *Data Mining* é a sua generalização e a capacidade de se adaptar a qualquer área do conhecimento. Estes conceitos englobam técnicas e ferramentas capazes de extrair informação em bases de dados das mais complexas e transformá-la em conhecimento útil para o negócio de organizações em qualquer domínio profissional. Alguns estudos têm-se focado na utilização de técnicas de *Data*

Mining para benefícios ambientais, nomeadamente no tratamento de águas residuais (Gallop et al., 2004).

Na atualidade, as *Estações de Tratamento de Águas Residuais* (ETAR) são infraestruturas muito importantes no tratamento de diversos cursos de água. Os efluentes que compõem as águas residuais são muitas vezes prejudiciais para o meio ambiente e para a própria saúde pública, devendo ser analisadas e tratadas segundo normas legais estipuladas no Diário da República. O processo de tratamento efetuado nas ETAR é bastante complexo e envolve várias etapas para a remoção de componentes nocivos, nomeadamente: os tratamentos preliminar, primário, secundário, e terciário. A necessidade de controlar a grande variedade de constituintes dos efluentes, e a preocupação em avaliar o processo de tratamento, levam as ETAR a proceder a diversas medições diárias de cada caraterística das águas residuais, usualmente através de sensores específicos para o efeito. As caraterísticas físicas, químicas e biológicas das águas residuais são assim medidas e registadas nas bases de dados. As leituras diárias realizadas pelos sensores de medição levam ao armazenamento de grandes volumes de dados, por vezes impossíveis de compreender sem um complexo processamento prévio. Com base nisto, a necessidade de melhorar e encontrar o processo de tratamento mais adequado para remover eficazmente todo o tipo de constituintes nocivos leva a perceber que existem muitas dificuldades no tratamento de águas residuais. Uma das formas mais eficazes de solucionar estas dificuldades, e tornar o processo de tratamento de águas residuais mais eficaz, encontra-se na aplicação de técnicas de *Data Mining*.

As técnicas de *Data Mining* para além de permitir a extração de padrões e tendências sobre os tratamentos das águas residuais possibilitam um eficaz auxílio no controlo e na previsão da qualidade do tratamento, e consequentemente da qualidade dos efluentes. As técnicas de previsão que o *Data Mining* envolve permitem prever valores das caraterísticas das águas residuais a partir de padrões e tendências descobertos através de das suas variadíssimas técnicas. Isso permite, por exemplo, a substituição de sensores avariados e a obtenção rápida de valores previsionais de parâmetros que normalmente demoram vários dias a serem conseguidos, como é o caso do parâmetro de *Carência Bioquímica em Oxigénio* (CBO) que necessita de 5 dias em laboratório para se apurar o seu valor medido (Meireles, 2011).

As técnicas de previsão podem ser de dois tipos: classificação ou regressão. Enquanto que a classificação atribui uma determinada classe a um registo, a regressão tenta prever o próprio valor de previsão. Uma das técnicas que mais tem evoluído e chamado a atenção de profissionais da área é a técnica *Support Vector Machines* (SVM). Esta técnica pode ser utilizada tanto para regressão como para classificação. Inúmeros autores têm-se dedicado no aperfeiçoamento desta técnica. Um dos focos de grande estudo sobre esta e outras técnicas de *Data Mining* é a atualização do modelo de previsão quando novos dados surgem. Usualmente, um modelo de previsão resulta da aprendizagem das tendências nos dados. Ora, quando surgem novos dados, torna-se necessário conhecer as novas tendências para que a previsão seja eficaz. Este facto obriga a atualização do modelo e, consequentemente, a realização de um novo processo de aprendizagem com todos os dados. Isso leva ao redescobrimento desnecessário e moroso de tendências já adquiridas. Uma nova forma de atualizar um modelo de previsão tem

sido adotada por alguns autores para diversas técnicas de *Data Mining*, entre as quais as técnicas de SVM. Esta vertente incremental do processo de atualização de um modelo de SVM permite extrair as novas tendências sobre os dados sem um reprocessamento completo da base de dados levando a uma grande diminuição do custo computacional que muitas vezes impedia uma eficiente utilização dos recursos (Cauwenberghs e Poggio, 2003).

Em infraestruturas como as ETAR, a variabilidade das caraterísticas das águas residuais é enorme. Todos os dias são registados novos dados, possivelmente portadores de novas tendências na composição dos efluentes. A utilização de técnicas de *Data Mining* sem uma atualização incremental dos modelos de previsão pode levar, por instantes, a inutilização dos sistemas o que não pode ser tolerável numa organização com a importância das ETAR.

1.2. Motivação e Objetivos

A importância dada a infraestruturas como as ETAR tem vindo a aumentar, devido a crescente preocupação na preservação dos ecossistemas e do meio ambiente. O processo de tratamento de águas residuais é alvo de estudos constantes por forma a melhorar a sua eficácia. Nas últimas décadas, a implementação de sensores capazes de realizar medições dos parâmetros físico-químicos, biológicos, e microbiológicos das águas residuais veio revolucionar o controlo e a avaliação da qualidade dos processos de tratamento e, consequentemente, das próprias águas residuais. A obtenção de conhecimento sobre os processos inerentes a cada organização sempre foi uma prioridade e um meio fundamental para o sucesso.

Em infraestruturas como as ETAR, o volume de informação registada diariamente pelas medições de controlo e avaliação das caraterísticas das águas, ao longo do processo de tratamento, é enorme. Existe uma necessidade óbvia de transformar toda essa informação em conhecimento útil para aperfeiçoar o processo de tratamento das águas residuais. Esta transformação é possível através das técnicas de *Data Mining*. A utilização de técnicas de *Data Mining* nas ETAR e as diversas melhorias que podem trazer para um processo de tratamento cada vez mais essencial na preservação do meio ambiente e da própria saúde pública é uma motivação acrescida para a realização deste projeto. Para além de permitir a extração de conhecimento, essencial para a tomada de decisões, a partir dos dados recolhidos ao longo dos anos. As técnicas de *Data Mining*, introduzidas no processo de tratamento de uma ETAR, possibilitam a previsão de situações de risco e ainda o auxílio nas decisões a tomar para a resolução de cada problema. Permitem, também, apoiar a monitorização do processo de tratamento substituindo parcialmente alguns sensores danificados permitindo a não interrupção momentânea de todo o funcionamento da ETAR.

O próprio contexto das ETAR, e o impacto que estas possuem na preservação da vida humana que se propõe tratar é, por si só, uma motivação para a realização desta dissertação. Por outro lado, temos a motivação de poder explorar novas tecnologias e expandir o conhecimento sobre estas. Outra motivação para a realização deste projeto é a possibilidade de aplicar os conhecimentos obtidos ao longo do percurso académico num caso real. Isso permite ter uma ideia das possíveis aplicações do nosso conhecimento e das aptidões detidas para a realização

de determinadas tarefas com vários níveis de dificuldades. Uma das motivações mais importantes para a realização desta dissertação foi a possibilidade de estudar e utilizar técnicas de *Data Mining* relativamente recentes, com poucos exemplos de aplicação, e com um potencial enorme, nomeadamente as técnicas de SVM incremental.

Objetivos de Negócio

Um projeto de *Data Mining* deve iniciar sempre pela análise e compreensão do negócio. A definição do problema de negócio e dos objetivos que se pretendem alcançar é essencial para se conhecer o objetivo real de qualquer projeto. O principal objetivo de negócio para uma infraestrutura como as ETAR, proposto na realização desta dissertação, é produzir um modelo de previsão capaz de trazer uma melhoria para o processo de tratamento e para o controlo e avaliação do mesmo. Deste modo, torna-se importante produzir um modelo de previsão eficiente que demonstre alguns dos benefícios que a sua implementação pode trazer para uma ETAR. Neste caso, realizar a previsão da qualidade dos efluentes de uma ETAR, ao longo do seu processo de tratamento, com um bom grau de precisão através das técnicas de *Data Mining* é o que se pretende verificar com este projeto. Outro objetivo de negócio é a produção de um meio de extração de conhecimento intrínseco nos dados das ETAR através das técnicas de *Data Mining*. Esses conhecimentos sobre os dados permitem compreender melhor os padrões de variabilidade das caraterísticas dos efluentes contidos nos dados e as tendências registadas ao longo dos anos.

Objetivos do projeto de *Data Mining*

Os objetivos de negócio permitem apresentar o que se pretende solucionar no contexto do problema do negócio. Os objetivos mais técnicos e aprofundados deste projeto de *Data Mining* envolvem o objetivo principal de produzir um modelo de previsão eficaz seguindo a metodologia de descoberta de conhecimento em base de dados (KDD), etapa por etapa. Como tal destacam-se os seguintes objetivos:

- A realização de uma eficiente preparação dos conjuntos de dados procedendo por exemplo à remoção de ruídos, de forma a possibilitar a produção de um modelo de previsão com elevados índices de assertividade. Para isso, é necessário analisar e testar as diversas técnicas de preparação de dados.
- A identificação da variável de previsão. Este passo necessita de um conhecimento prévio sobre os dados e sobre o negócio.
- A análise e utilização de técnicas que permitem reduzir a dimensionalidade dos dados, seleção de atributos. Pretende-se ainda nestas técnicas, identificar as características dos efluentes mais correlacionadas entre si com a variável de interesse que será objeto de previsão.
- A análise exaustiva dos algoritmos de SVM, comparando os algoritmos de classificação e de regressão e analisando e testando cada caraterística e parâmetro de configuração que permitem produzir modelos com grande eficácia.

- A análise e utilização de diversas ferramentas de *Data Mining* para a produção dos modelos de previsão. Os modelos de previsão produzidos neste projeto resultam da aprendizagem de modelos de SVM.
- A análise e utilização de técnicas de atualização dos modelos de previsão, nomeadamente técnicas de SVM incremental.
- A avaliação dos modelos de SVM produzidos utilizando os métodos de avaliação. Este objetivo pressupõe um estudo dos métodos de avaliação e das medidas de desempenho dos modelos de SVM. Pretende-se ainda uma comparação dos testes de avaliação realizados entre as ferramentas utilizadas e os modelos produzidos.

1.3. Estrutura do Documento

Para além do presente capítulo, este documento está estruturado em mais oito capítulos, nomeadamente:

- **Capítulo 2 – Estações de Tratamento de Águas Residuais** - Este capítulo apresenta o contexto em que se inseriu este projeto de *Data Mining*. Grande parte da análise de negócio é aqui realizada.
- **Capítulo 3 – Data Mining** - Neste capítulo são apresentados os conceitos de *Data Mining*, as suas vantagens, o seu propósito e objetivos. São ainda apresentadas as diversas técnicas de *Data Mining* existentes e enumeradas as diversas dificuldades normalmente encontradas em projetos de *Data Mining*.
- **Capítulo 4 – Support Vector Machines** - Neste capítulo são apresentadas as técnicas de SVM com mais detalhe, assim como toda a sua fundamentação teórica.
- **Capítulo 5 – Data Mining Incremental** - Neste capítulo é introduzido e apresentado detalhadamente o conceito de *Data Mining* incremental e as suas vantagens para a atualização de modelos de previsão. São ainda apresentadas e analisadas algumas técnicas de SVM incremental.
- **Capítulo 6 – Análise e Preparação dos dados** - Neste capítulo são apresentados alguns conceitos de análise e preparação de dados. Depois, segue-se uma análise dos conjuntos de dados utilizados neste projeto, incluindo uma caraterização dos dados e uma descrição de todo o processo de preparação realizado em cada conjunto de dados, nomeadamente o processo de seleção de atributos e a transformação de formatos de dados.
- **Capítulo 7 – Construção dos modelos de previsão** - Neste capítulo é apresentado o processo de modelação e as ferramentas adotadas para realizar as várias tarefas de previsão dos parâmetros dos efluentes das ETAR. São apresentados e analisados os vários suportes utilizados para a produção dos modelos de SVM.
- **Capítulo 8 – Testes e resultados -** Neste capítulo são apresentados os testes realizados e analisados os resultados obtidos. São ainda apresentadas algumas representações gráficas que permitem comparar os diferentes modelos de previsão produzidos.

- **Capítulo 9 – Conclusões e Trabalho Futuro -** Neste último capítulo são apresentadas algumas conclusões tiradas acerca dos testes realizados, bem como apresentadas algumas sugestões para trabalho futuro.

Capítulo 2

As Estações de Tratamento de Águas Residuais

2.1. A Importância da água

Atualmente, uma das principais preocupações das economias mundiais é a sustentação da vida e a preservação dos recursos vitais. Este sistema sociológico, demográfico, e económico que abrange o mundo inteiro não parece estar em condições de aguentar esta necessidade básica. Segundo diversos estudos realizados, o consumo de água no Planeta está a aumentar a um ritmo insustentável. Existem locais em que a quantidade de água consumida, seja na agricultura, indústria, ou no uso doméstico, é muito superior ao que a própria natureza consegue repor. Nas últimas décadas, o consumo de água aumentou aproximadamente 6 vezes, isto é, mais do dobro do crescimento da população mundial. Cerca de 70% dos meios hídricos disponíveis é utilizada na agricultura, 22% gasta no uso industrial e 8% para o uso doméstico. O consumo médio nos países desenvolvidos tem uma variação per capita de 500 e 800 litros de água por dia. Nos países em desenvolvimento são apenas consumidos por pessoa em média 60 a 150 litros. Nos dias que correm, 40% da população vive em regiões sujeitas à escassez hídrica, prevendo-se que em 2025 este número se aproxime dos 65%, isto é, cerca de5,5 mil milhões de pessoas. Estima-se, também, que nesse mesmo ano cerca de mil milhões de pessoas no mundo não tenham acesso a água potável e mais de 2 mil milhões vivam sem condições sanitárias básicas. Consequentemente, mais de 250 milhões de pessoas por ano são afetadas por doenças relacionadas com o consumo impróprio de água considerada não potável. (Lopes, P., 2009).

O investimento anual em infraestruturas hídricas representa atualmente pouco mais de 70 mil milhões de euros, quando o necessário seria cerca de 180 mil milhões. Concisamente, o desenvolvimento e crescimento da ocupação humana e das cidades, assim como, as alterações climáticas têm vindo a reforçar as influências que o homem exerce sobre a utilização da água. Generalizando, pode-se afirmar que existe uma disponibilidade de água doce suficiente. No entanto esta encontra-se mal repartida no tempo e no espaço, sendo utilizada em muitos países de forma abusiva e insustentável.

2.2. Propriedades das águas residuais

A qualidade das massas hídricas para qualquer tipo de utilização é um indicador fundamental para avaliar o desenvolvimento de um país e do bem-estar dos seus cidadãos. Em Portugal tem-se verificado uma evolução bastante satisfatória, quer relativamente à qualidade da água distribuída, quer na realização de análises para o seu controlo. Os últimos dados nacionais conhecidos não deixam quaisquer dúvidas, comprovando uma grande melhoria no controlo e avaliação da qualidade da água nos últimos 10 anos. A avaliação da qualidade das massas de águas encontra-se prevista num enquadramento legal europeu. Este normativo baseia-se num conjunto de parâmetros físico-químicos, microbiológicos e biológicos. Em alguns casos, como por exemplos das águas balneares e de recreio, a qualidade microbiológica tem uma maior importância devido a constante exposição dos cidadãos a agentes patogénicos. Em termos legais, controlar e avaliar a qualidade microbiológica das águas é uma prática que baseada no indicador de contaminação fecal. Este conceito descreve o pressuposto básico de que a sua deteção indica a presença de contaminação fecal. Os indicadores clássicos habitualmente usados são:

- O grupo dos Coliformes (e.g. coliformes totais, coliformes fecais e *Escherichia coli).*
- Streptococci (e.g. Enterococci e streptococci fecal).
- Formadores de esporos (e.g. Cloctridium perfringens).

Assim, e para preservar a saúde pública, é necessário proceder ao tratamento das águas, sobretudo das águas residuais. As águas residuais são águas que resultam de vários tipos de utilização ligados à atividade humana, nomeadamente água que foi utilizada nas habitações ou na indústria. Estas massas hídricas possuem geralmente na sua composição componentes prejudiciais para o meio ambiente e para o ser humano.

2.3. ETAR

2.3.1 Tratamento das Águas Residuais

A água contém várias impurezas que definem a sua qualidade (Tabela 1). Isso deve-se às suas caraterísticas como solvente e à sua capacidade de transportar substâncias. A qualidade é essencialmente afetada pelos fenómenos naturais e pela intervenção do ser humano. O homem é o principal agente poluidor das massas hídricas por consequência, por exemplo, das evacuações de águas residuais provenientes das habitações e industria, e também da poluição dos solos com pesticidas e fertilizantes (Sperling, 2007). O tratamento de águas residuais é realizado para se evitar riscos para a saúde pública e remover os poluentes dos afluentes para os quais são descarregados efluentes, nomeadamente a rede hidrográfica, os lagos e o mar. A contaminação dos cursos de água e a acumulação de águas residuais não tratados levam à extinção da flora e da fauna aquática, limita atividades económicas, sociais e de recreio normais,

levam a poluição do ambiente em geral sob a forma de odores desagradáveis, de paisagem alterada, e de contaminação das águas subterrâneas (Ministério do Ambiente, 2004).

Os métodos usados nas ETAR na deteção e quantificação das medições dos indicadores não são, por vezes, compatíveis com sistemas de alerta previsionais, particularmente pelo tempo que é preciso para a obtenção analítica dos valores, e pelos recursos materiais e humanos necessários. Para além dos parâmetros microbiológicos também devem ser considerados os parâmetros físico-químicos da água, tais como, a sua temperatura, o nível de pH, a quantidade de oxigénio dissolvido, a condutividade elétrica, associados às fontes de carbono e energia e turvação originada pelos sólidos em suspensão. Incluem-se ainda em medições úteis, mas menos importantes, os parâmetros relacionados com o ambiente tais como a precipitação, radiação, direção e velocidade do vento, nível da maré, população de aves e caraterísticas do solo. Estes são todos os fatores que podem ser relevantes na avaliação da qualidade da água.

Componente	Exemplo	Efeitos negativos
Materiais sólidos de grandes dimensões	Papéis, trapos, sacos de plástico, entre outros	Os seus efeitos prejudiciais são a poluição visual provocada pela sua acumulação nos meios hídricos e o risco para a saúde pública provocada pela proliferação de agentes infeciosos.
Matéria orgânica	Proteínas, hidratos de carbono, gorduras, organismos biodegradáveis	São normalmente medidos como CBO (carência bioquímica de oxigénio) e CQO (carência química de oxigénio). Se descarregados sem tratamento, podem levar à redução do oxigénio nos recursos naturais e ao desenvolvimento de condições sépticas.
Óleos e gorduras	Óleos automóveis, azeite, entre outros	Provoca a formação de espuma nas superfícies do meio hídrico com consequente degradação da paisagem e consequências negativas para toda a biosfera do meio e a formação de uma película impermeável na superfície líquida, reduzindo a transferência de oxigénio da atmosfera para o meio líquido, com as inerentes consequências que essa redução acarreta.
Nutrientes	Azoto e o Fósforo	Funcionam como fertilizantes e estimulam o crescimento de algas, tóxicas ou não, e outras plantas aquáticas que obstruem os cursos de água e poluem as margens dos meios hídricos com material em decomposição, que eventualmente se transforma em resíduos orgânicos, com as consequências nefastas daí decorrentes.
Bactérias e vírus	Coliformes, Estreptococia, entre outros	As suas presenças em recursos hídricos onde se faz captação de água para consumo humano ou irrigação de culturas, que serão depois ingeridas cruas por pessoas ou animais, originam riscos graves para a saúde pública podendo provocar doenças, tais como, cólera, febre tifoide e salmonelas.
Substâncias tóxicas	Efluentes industriais, tintas, vestígios petrolíferos, fertilizantes, entre outros.	Podem danificar ou destruir a vida aquática e/ou serem acumuladas ao longo da cadeia alimentar até chegarem ao ser humano

Tabela 1 - Componentes das águas residuais e efeitos negativos.

As políticas de saúde pública reservam uma grande atenção à garantia da qualidade da água como um elemento essencial para a sociedade. A monitorização da qualidade dos efluentes das ETAR assenta em métodos analíticos, cada vez mais sofisticados e eficientes. Estes métodos dispõem de sensores capazes de registar valores fiáveis das várias condicionantes da qualidade da água.

Em função da sua origem há dois grandes tipos de água residuais: as domésticas e as industriais. As águas residuais domésticas são usualmente consequentes da atividade residencial, podendo ser integradas por águas fecais e saponáceas. As águas residuais industriais

proveem das descargas de diversas organizações industriais. Os principais componentes das águas residuais a serem tratados e analisados encontram-se descritos na Tabela 2.

Categoria	Parâmetro	Descrição
Químicos	Sólidos Suspensos Totais (SST)	Total de sólidos orgânicos e inorgânicos que não são filtráveis. Compostos minerais não oxidáveis pelo calor e inertes.
	Sólidos Suspensos Voláteis (SSV)	Parte de sólidos orgânicos e inorgânicos que não são filtráveis. Compostos minerais oxidáveis pelo calor.
	Carência Bioquímica de oxigénio (CBO)	Medidor da quantidade de oxigénio consumido pelos microrganismos na decomposição aeróbia da matéria orgânica, dentro de determinadas condições;
	Carência Química de Oxigénio (CQO)	Medidor da quantidade de oxigénio que é consumido na oxidação por via química da matéria orgânica e da matéria inorgânica presente, convertendo-as em dióxido de carbono e água.
	pH	Medidor de acidez ou alcalinidade das águas residuais.
Físicos	Temperatura	Varia com as estações do ano e influencia a atividade microbial, a solubilidade dos gases e a viscosidade dos líquidos.
	Odor	Águas residuais sépticas têm odores mais fortes e desagradáveis. Resíduos industriais têm odores característicos.
	Cor	Diferentes cores indicam diferentes tipos de águas, i.e. água mais escura indica fortes condições sépticas, água colorida indica que provém dos desperdícios industriais.
	Turbidez	Causada pela grande variedade de sólidos suspensos.
Biológicos	Bactérias	Organismos unicelulares, presentes em várias formas e tamanhos. Algumas bactérias são patogénicas, causam principalmente doenças intestinais.
	Protozoários	Essenciais no tratamento biológico para manter o equilíbrio dos vários grupos. Alguns são patogénicos.
	Vírus	Organismos parasitas, patogénicos e de difícil remoção no tratamento.

Tabela 2 - Descrição dos componentes normalmente encontrados em águas residuais. Extraído e adaptado de (Ribeiro,2012)

As ETAR são infraestruturas de extrema importância que têm a capacidade de proceder ao tratamento dos cursos de águas através de diversas técnicas de purificação da água. Nas ETAR são efetuadas várias medições dos parâmetros que caraterizam as águas residuais ao longo de todo o tratamento através de análises laboratoriais das águas ou por equipamentos de

monitorização, tais como sensores de medição. Nos últimos anos tem-se verificado uma grande evolução na abrangência dos sensores de medição dos parâmetros de qualidade das águas e dos efluentes das ETAR, os quais necessitam de ser aprovados em diferentes matrizes ambientais. Estas normas legais são bastante importantes pois impõe às ETAR a obrigação de respeitar valores recomendados para a composição das massas hídricas emitidas à saída da estação. Estes valores encontram-se regulamentados na lei, nomeadamente no Decreto-lei nº 152/97 de 19 de Junho do Diário da República. Neste decreto de lei são regulamentados os parâmetros químicos como o CBO, CQO e SST, com limites nas concentrações à saída da ETAR – veja-se a Tabela 3.

Parâmetros	Concentração
Total de partículas sólidas em suspensão (SS)	35 Mg/l
Carência bioquímica de oxigénio (CBO)	25 Mg/l
Carência química de oxigénio (CQO)	125 Mg/l

Tabela 3 - Valores máximos recomendados para os parâmetros CBO, CQO, SS (Decreto-lei nº 152/97).

O Processo de Tratamento

Numa ETAR, para além do tratamento do estado líquido (água), também se procede ao tratamento do estado sólido (lamas) e, por vezes, do estado gasoso (gases) de todos os componentes extraídos das águas residuais. O tratamento do estado líquido é composto por quatro tratamentos sequenciais. Em cada um destes tipos de tratamento podem ser utilizados um ou vários processos de tratamento. De referir:

- Tratamento preliminar.
- Tratamento primário.
- Tratamento secundário.
- Tratamento terciário.

Os processos de tratamento variam de ETAR para ETAR, uma vez que dependem de diversos fatores, tais como o tipo de substâncias presentes nas águas residuais ou a concentração e a capacidade do meio recetor em diluir e assimilar essas substâncias. Um determinado nível de qualidade do meio recetor é imposto de acordo com a sensibilidade dos ecossistemas e com o tipo de atividade humana que é realizada na área da descarga da ETAR. As ETAR podem ainda ser classificadas de acordo com o tratamento secundário aplicado, nomeadamente: ETAR por lagunagem, ETAR por leitos percoladores e ETAR por lamas ativadas.

Tratamento Preliminar

No tratamento preliminar (ou pré-tratamento) as massas hídricas à entrada da ETAR são sujeitas à decomposição dos sólidos de maiores dimensões através de vários processos

nomeadamente, a gradagem e desarenação por sedimentação dos sólidos. É necessário retirar da água residual parte da elevada carga de sólidos e detritos, antes de prosseguir para as etapas seguintes. Este processo permite evitar problemas de manutenção como o desgaste e avaria de equipamentos, assim como o desgaste das próprias tubagens do sistema. Na fase preliminar é ainda efetuada a remoção dos óleos e gorduras através de desoleadores e desengorduradores. Para uma remoção mais eficiente usam-se sistemas de flotação, que consistem na injeção de ar ascendente por forma a arrastar assim as impurezas para o topo, sendo estas posteriormente removidas.

O Tratamento Primário

O tratamento primário procede à remoção de algumas cargas de sólidos suspensos e de matéria orgânica. Este processo separa os agentes poluentes da água, por decantação nos decantadores primários. Este processo é normalmente apenas de ação física, porém pode em alguns casos ser auxiliado pela adição de agentes químicos, que através de processos de floculação permitem a obtenção de flocos de matéria poluente que são assim mais facilmente sedimentáveis e removíveis. Este processo reduz eficazmente cerca de 60% da carga de sólidos suspensos (SS) e 30% dos valores de Carência bioquímica e química de oxigénio (CBO/CQO).

O Tratamento Secundário

Após o tratamento primário, procede-se ao tratamento secundário. Este processo é normalmente um processo biológico que tem como objetivo remover a carga orgânica (CBO e CQO). Como foi referido anteriormente, existem diversos tipos de tratamentos secundários, porém, o mais comum é o tratamento por lamas ativadas por ser um processo intensivo de tratamento que permite obter elevadas eficiências em termos de remoção. Este processo é realizado por oxidação da matéria orgânica, em tanques de arejamento. A etapa de arejamento é essencial no tratamento secundário. No método de tratamento por lamas ativadas, a massa biológica aeróbica em suspensão utiliza o oxigénio lançado pelos difusores para decompor a matéria orgânica existente em dióxido de carbono, água e energia. O tratamento biológico é efetuado, geralmente, num reator em mistura completa, sendo a biomassa mantida em suspensão A biodegradação produzida nesta etapa depende de um adequado arejamento (níveis de oxigénio dissolvido) e de uma concentração ideal de microrganismos que é possível através de uma recirculação parcial das lamas recolhidas durante a etapa de decantação secundária (Barroso, 2012). A decantação secundária é parte essencial do tratamento secundário do efluente e é aplicada após o tratamento biológico, tendo como principais objetivos a clarificação do afluente através da separação da biomassa mineralizada e floculada no reator biológico devido à ação microbiana e o espessamento e extração das lamas criadas assim como, através da recirculação de lamas ao tanque de arejamento, manter a concentração de lamas ativadas necessárias para manter a eficiência do reator no tratamento das águas residuais (Luizi, 2012).

De salientar que, em paralelo a este procedimento realiza-se à etapa sólida do tratamento correspondentes aos tratamentos das lamas. O excesso de microrganismos aeróbios responsáveis pela oxidação da matéria orgânica é também prejudicial para o tratamento, e por

isso é necessário ocasionalmente proceder ao tratamento das lamas. A concentração ideal de microrganismos aeróbios é conservada através de uma recirculação parcial das lamas retiradas durante a etapa de decantação secundária. No final do tratamento secundário as águas encontram-se, segundo as normas mínimas aceitáveis legais anteriormente mencionadas, prontas para serem libertadas para o meio ambiente. No entanto, as águas residuais após o tratamento secundário podem ainda conter altos níveis de nutrientes, como azoto e fósforo. A emissão em demasia destes compostos para o meio hídrico pode levar à acumulação de nutrientes, dando-se a eutrofização do meio, que estimula o crescimento excessivo de algas (designado bloom) e cianobactérias (algas azuis). A maior parte destas algas acaba por morrer, contudo, a sua decomposição por bactérias remove oxigénio da água e a maioria dos peixes morre. Além disso, algumas espécies de algas produzem toxinas, que contaminam as fontes de água potável (cianotoxinas) (Meireles, 2011). Para além disso, as exigências normativas têm-se tornado cada vez mais rigorosas. Algumas ETAR apenas realizam o tratamento até ao tratamento secundário. Porém os níveis de qualidade exigidos pelas normas e pelo próprio meio ambiente levam à necessidade de realização do tratamento terciário.

O Tratamento Terciário

Antes do lançamento final do efluente tratado é necessário proceder à sua desinfeção, conforme a classificação do meio ambiente envolvente. Este processo permite proceder a remoção dos organismos patogénicos e à remoção de determinados nutrientes, como azoto e o fósforo, que podem potenciar, isoladamente ou em conjunto, a eutrofização das águas receptoras. Existem diferentes processos de remoção de azoto e fósforo, nomeadamente:

- A desnitrificação, que pode ser obtida por via biológica em condições anaeróbias (ausência de oxigénio), para que a comunidade biológica apropriada se forme. Neste processo os nitratos presentes na água são reduzidos a azoto gasoso, que se liberta para a atmosfera e simultaneamente ocorre a oxidação de matéria pelos microrganismos;
- A remoção do fósforo, que pode ser realizada por precipitação química, geralmente com sais de ferro (ex. cloreto férrico) ou alumínio (ex. sulfato de alumínio), mas a lama resultante deste tratamento químico é de difícil tratamento e o uso de produtos químicos torna-se dispendioso. Em alternativa pode recorrer-se à remoção de fósforo por via biológica, promovendo a recirculação do efluente tratado e criando uma etapa com condições anaeróbias. Na etapa aeróbia, os microrganismos removem o fósforo das águas acumulando-o nos seus tecidos.

A desinfeção das águas residuais tratadas tem como objetivo a remoção de organismos patogénicos. Os processos mais utilizados são (Myers, 1998):

- **Ozonização** - criação de ozono no local com a passagem de uma descarga elétrica através de ar seco ou oxigénio, de modo a possibilitar a remoção de microrganismos existentes;
- **Filtração por membranas** - retenção de microrganismos em membranas de malha bastante reduzidas;

- **Radiação ultravioleta** - eliminação dos microrganismos através da radiação ultravioleta;
- **Cloragem** - remoção de microrganismos através da adição de cloro, que é bastante tóxico para eles.

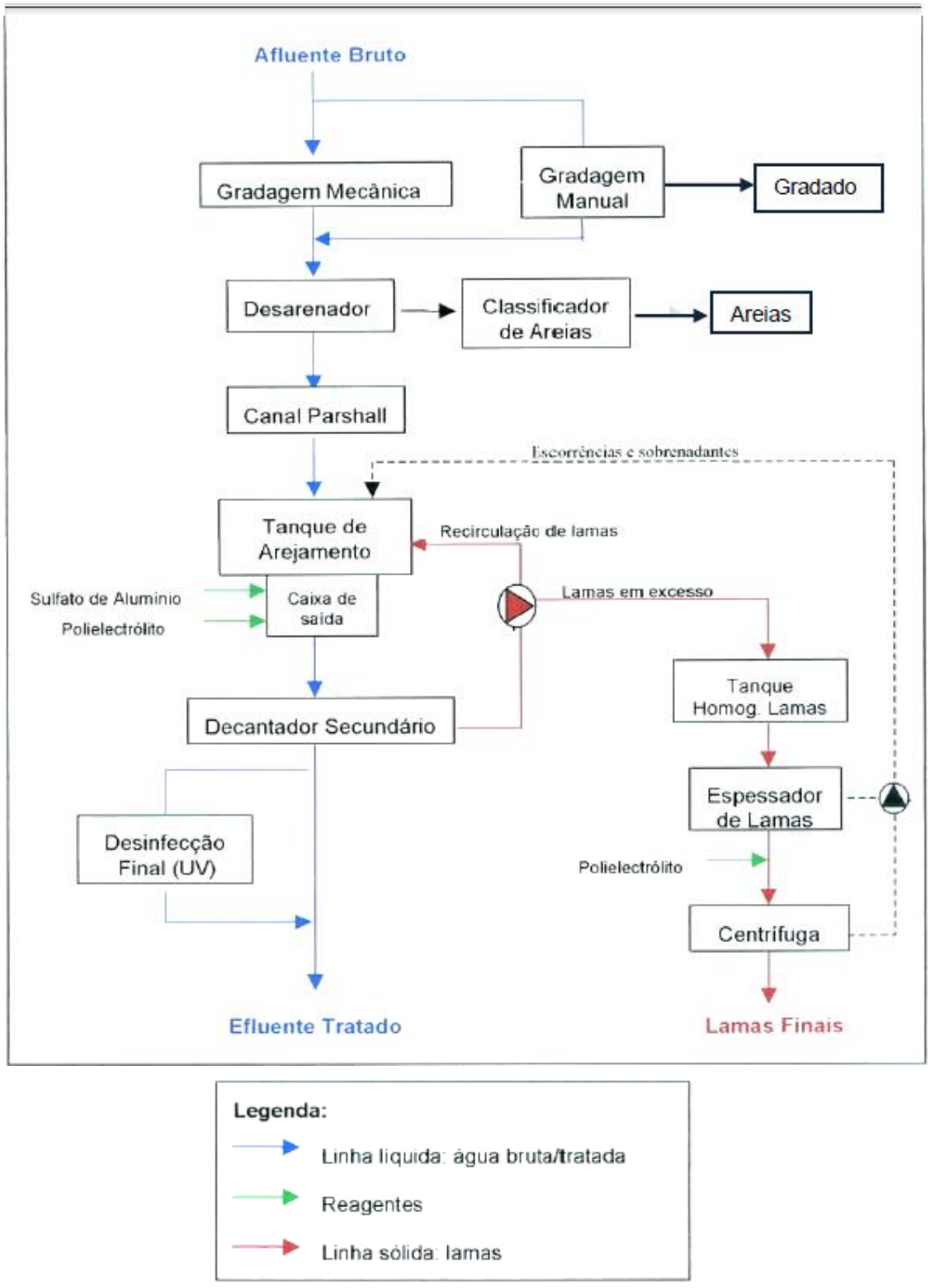

Figura 1 - Processo de tratamento realizado nas ETAR com lamas ativadas – figura extraída de (EFACEC, 2003).

Capitulo 3

Data Mining

3.1. Data Mining no Controlo do Tratamento de uma ETAR

Em grande parte das empresas, a manipulação de dados é muito importante para que não se perca informação, causando imprevistos, falhas ou custos desnecessários que podem ocorrer na hora de manipular e administrar a informação. Segundo alguns especialistas, o modelo tradicional de transformação de dados em informação útil consiste num processamento manual de todos esses dados por especialistas que, por sua vez, produzem informação posteriormente analisada. Devido ao grande volume de dados diariamente recolhidos, este processo manual torna-se completamente inviável. É neste contexto que entra a mineração de dados (*Data Mining*). Este processo é utilizado para trabalhar em grandes bases de dados de forma a procurar padrões e encontrar correlações e tendências entre as informações. Este conjunto de técnicas é capaz de inter-relacionar informações antigas com as do presente e predeterminar as do futuro (Cortês, et al., 2002).

A mineração de dados é um processo de análise de dados, sob diferentes perspetivas e de uma forma sumarizada, criando informação útil. Este processo permite a realização de várias operações sobre os dados que possibilitam, sob várias dimensões ou vistas, categorizar os dados e sumariá-los segundo as suas relações. A mineração de dados permite encontrar correlações, padrões, associações**,** mudanças e anomalias que estão subjacentes a esses dados e que não são imediatamente percetíveis por uma simples observação humana (Han, Pei e Kamber, 2006).

Os sistemas de *Data Mining* possuem dois objetivos, pois tanto apontam para a predição, capacidade para aprender critérios que possam suportar decisões futuras, como também permitem descobrir novo conhecimento, encontrando nos dados possíveis padrões desconhecidos que refletem comportamentos que ainda não tinham sido apercebidos pelo decisor. Assim, podemos esperar das ferramentas de *Data Mining* agilidade, confiança, prevenção e comparação dos dados, bem como a extração de informação útil para as organizações (Hand, et al., 2001).

Nos últimos anos, e devido as grandes exigências legais e ambientais, tem-se verificado uma grande melhoria na abrangência dos sensores para a medição dos parâmetros de qualidade dos efluentes das ETAR, os quais necessitam de ser validados em diferentes matrizes ambientais, tais como de salinidade variável e no conhecimento dos fatores que determinam o decaimento dos contaminantes. A monitorização rápida dos parâmetros microbiológicos de contaminação fecal é objeto de investigação de ponta, para o qual ainda não existem no mercado sensores com fiabilidade estabelecida. Diariamente é recolhido um grande número de dados relativos às leituras realizadas pelos diferentes sensores nas ETAR. Esses dados variam em número e em género, sendo que a composição das partículas existentes nos efluentes pode variar de dia para dia. Estas alterações nas quantidades de partículas microbiológicas e nos parâmetros físico-químicos levam a necessidade de classificar as águas segundo o seu grau de contaminação. Como pode ser inferido, enormes quantidades de dados são armazenadas na avaliação da qualidade das águas residuais. A vantagem da utilização de parâmetros físico-químicos em relação aos indicadores microbiológicos é a sua robustez, facilidade e rapidez de medição, podendo ser avaliados de modo automático em tempo real com a utilização dos sensores. A utilização de técnicas de *Data Mining* permite, por exemplo, uma maior facilidade de utilização dos indicadores microbiológicos o que torna o controlo do tratamento bastante mais completo e eficaz.

Nas ETAR a utilização deste tipo de técnicas para a previsão da qualidade dos seus efluentes representaria uma melhoria significativa, tanto no processo de tratamento, como no processo de avaliação dos riscos de contaminação das águas. Com o uso de técnicas de *Data Mining* em infraestruturas como as ETAR temos a possibilidade de prever o grau de contaminação da água e escolher o melhor tratamento a ser utilizado para aqueles valores. Assim como, possibilitar a substituição parcial de alguns sensores defeituosos permitindo assim manter o bom funcionamento de toda a infraestrutura. Estas são algumas das várias possibilidades das técnicas de *Data Mining* em ambientes como as ETAR.

A natureza do processo de tratamento biológico, efetuado nas ETAR, é bastante diversificada e requer um tratamento dinâmico, adaptado para cada caso em particular. Assim, a variabilidade e a complexidade da composição da água não tratada que entra na ETAR e a falta de sensores disponíveis são dificuldades que podem ser resolvidas com a aplicação das várias técnicas de *Data Mining*. Este tipo de sistemas permite apoiar os decisores nas tomadas de decisão relativas às medidas que devem ser tomadas em cada caso específico, conforme o nível de contaminação registado. Por exemplo, a previsão da qualidade das águas residuais, com base nos parâmetros medidos, permite a avaliação do desempenho do tratamento e ainda obter informações úteis para um melhor controlo de toda a infraestrutura da ETAR. A utilização dos dados históricos e resultados anteriormente obtidos em conjunto com estas técnicas permitem substituir parcialmente sensores avariados e possibilita ainda a prevenção dos valores que seriam recolhidos pelo sensor avariado consoante os valores lidos para os outros parâmetros.

3.2 Metodologias

O processo de *Data Mining* é parte de um processo maior conhecido como KDD (Knowledge Discovery in Databases). O KDD é um conjunto de passos bem definidos, executados de forma iterativa e interativa. São interativos porque envolvem a cooperação do analista de dados, cujo conhecimento sobre o domínio orientará a execução do processo. Por sua vez, a iteração deve-se ao fato de que, com frequência, esse processo não é executado de forma sequencial, mas envolve repetidas seleções de parâmetros e conjunto de dados, aplicações das técnicas de *Data Mining* e posterior análise dos resultados obtidos, a fim de refinar os conhecimentos extraídos. Para o caso em estudo, como as ETAR que podem possuir uma grande variabilidade nos seus dados, é essencial que este processo seja iterativo a fim de obter sempre que possível uma adaptação dos modelos criados no processo de *Data Mining* e extrair conhecimento do mais fiável possível.

O KDD consiste, então, na descoberta de conhecimento a partir de um conjunto de dados. Esta descoberta é feita extraindo informação útil para o negócio. O processo de descoberta de conhecimento é atualmente uma referência e tida como "um processo não trivial de identificação de padrões presentes nos dados, novos, válidos, potencialmente úteis e compreensíveis"(Fayyad et al., 1996). Um processo típico de KDD inclui 5 fases (Figura 2):

- **Seleção -** Nesta fase inicial são escolhidos apenas os atributos relevantes do conjunto de atributos da base de dados. Ou seja, faz-se a seleção de um subconjunto de atributos relevantes para o objetivo da tarefa. O subconjunto selecionado é então fornecido para as tarefas de *Data Mining*.
- **Pré-processamento -** O objetivo desta fase é assegurar a qualidade dos dados selecionados. A limpeza dos dados envolve a verificação da consistência das informações, a correção de possíveis erros e o preenchimento ou a eliminação de valores nulos e redundantes. São identificados e removidos os dados duplicados e os ruídos. A execução dessa fase tem efeitos de melhoria do processo de *Data Mining* pois elimina consultas desnecessárias que seriam executadas pelo algoritmo e que afetariam o seu treino.
- **Transformação -** O objetivo da transformação de dados é converter o conjunto bruto de dados em uma forma padrão de utilização. Os dados do atributo podem por exemplo ser padronizados dentro de uma faixa de valores, como por exemplo: 0 a 1 (processo de normalização). Normalmente essa etapa exige a experiência do analista de dados e seu conhecimento sobre os próprios dados em questão.
- **Data Mining -** Segundo Adriaans e Zantinge (1996), existe uma grande interligação entre os termos *Data* Mining e KDD. O termo KDD é usado para nomear o processo de extração de conhecimento de um conjunto de dados. Pode-se afirmar que conhecimento significa relações e padrões entre os elementos dos conjuntos de dados. O termo *Data Mining* deve ser usado somente para a fase de descoberta de conhecimento do processo de KDD.
- **Interpretação e avaliação -** A interpretação tem como objetivo melhorar a compreensão do conhecimento descoberto pelo algoritmo de modelação,

validando-o através de medidas de qualidade (precisão, desvio padrão, entre outras).

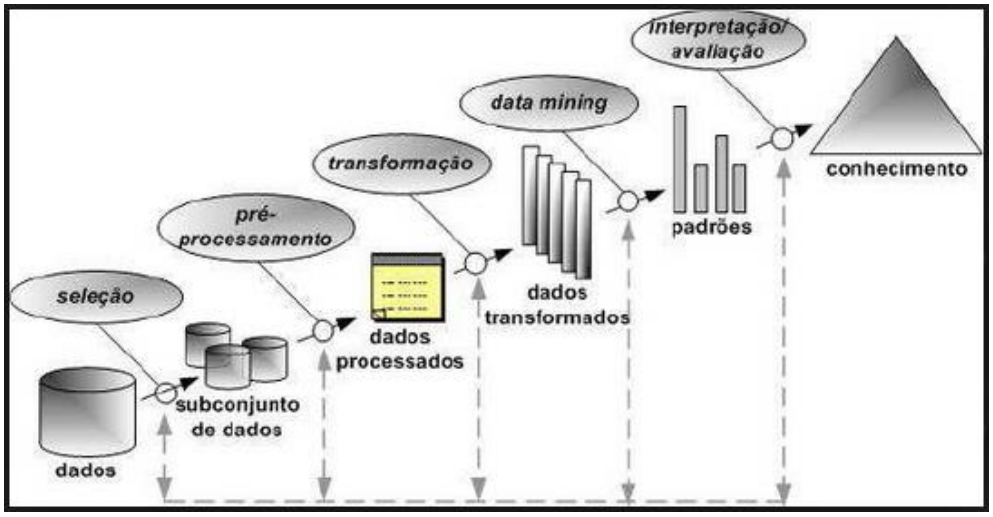

Figura 2 - Fases do processo de KDD - fonte: Fayyad et al. (1996).

Atualmente as duas metodologias de *Data Mining* mais utilizadas são a metodologia CRISP-DM (CRoss-Industry Standard Process for *Data Mining*) e a metodologia SEMMA (Sample, Explore, Modify Model, Assessment). Estas metodologias foram desenvolvidas em ambientes diferentes. A primeira foi desenvolvida por um grupo de organizações de diferentes sectores de atividade e a segunda por uma organização fornecedora de soluções de suporte à decisão e *Business Intelligence* chamada SAS. Porém, a metodologia mais adotada em todo o mundo é a metodologia Crisp-DM.

A Metodologia SEMMA

A metodologia SEMMA (Azevedo e Santos, 2008) desenvolvida pela SAS define o processo de *Data Mining* como um processo de extração de informação útil e de relações complexas entre os atributos de um grande volume de dados. Nesta metodologia o processo de *Data Mining* encontra-se dividido em cinco fases distintas, que compõe o acrónimo SEMMA e que são:

- ***Sample*** – Recolha de uma amostra significativamente grande para ser representativa do conjunto total e pequena o suficiente para ser manipulada rapidamente.
- ***Explore*** – Exploração estatística e gráfica dos dados para se ter ideia à partida de algum padrão, tendências ou anomalias nos dados.
- ***Modify*** – Modificação dos dados criando, selecionando e transformando variáveis para obter novas informações. Identificação de *outliers*, tratamento de valores nulos e partição do conjunto de dados.

- ***Model*** – Treino de modelos preditivos, modelação das variáveis de previsão usando algoritmos baseados em árvores de decisão, regressões, redes neuronais ou modelos definidos pelos analistas.
- ***Assess*** – Comparação analítica e gráfica dos vários modelos preditivos. Avalia-se o melhor e "classifica-se" a nova informação.

A Metodologia CRISP-DM

A metodologia CRISP-DM (CRoss-Industry Standard Process for *Data Mining*) (Azevedo e Santos, 2008) foi desenvolvida em finais de 1996. O crescente interesse por parte de diversas organizações de vários sectores de atividade numa metodologia padronizada e não especifica para cada área levou ao seu desenvolvimento. Este conjunto de procedimentos baseia-se não só em princípios académicos e teóricos como também na prática e experiência das organizações que desenvolvem projetos de *Data Mining*. Esta metodologia foi desenvolvida baseando-se não só na tecnologia utilizada mas essencialmente na resolução de problemas do negócio (Han et al., 2001). As fases não possuem uma sequência fixa, dependendo do resultado e do desempenho das outras fases ou das tarefas particulares de determinada fase. O CRISP-DM tem-se destacado como a metodologia padrão em todo o mundo. É composta por 6 fases distintas:

- **Análise do Negócio -** A análise do negócio visa compreender problemas relativos a um contexto. Esta fase inicial concentra-se em entender os objetivos do projeto e os requisitos de uma perspetiva de negócios e posteriormente converter o conhecimento dos dados na definição de um problema de *Data Mining*.
- **Análise dos dados -** A análise dos dados visa a recolha dos dados iniciais, um estudo preliminar destinado a uma familiarização maior com os mesmos e a avaliação da qualidade dos dados. Em consequência dessas atividades, é comum a descoberta de padrões interessantes já nesta fase. Possui como tarefas: recolha de dados, descrição dos dados, exploração dos dados e verificação da qualidade dos dados.
- **Preparação dos dados** - A preparação dos dados visa a construção final do conjunto de dados que será submetido à ferramenta de modelação. Os dados são sujeitos a um tratamento de limpeza e de transformação. As tarefas associadas a esta fase são a seleção dos dados, limpeza dos dados, formatação e transformação de dados.
- **Modelação** - Nesta fase é escolhida a técnica de modelação dos dados que será utilizada. Dependendo da técnica escolhida, pode ser necessário voltar a fase de preparação de dados para ajustar os dados às exigências de algumas técnicas. As tarefas desta fase são a seleção da técnica de modelação, criação do modelo, e validação do mesmo.
- **Avaliação** - A fase de avaliação dos modelos visa avaliar se os modelos já criados cumprem os objetivos de negócio propostos inicialmente. Caso não seja o caso, é necessário rever todos os passos realizados e proceder as alterações necessárias. Caso os modelos cumprem com os requisitos iniciais, seguimos para a última fase.
- **Implementação** - A fase da implementação é a última fase da metodologia de CRISP-DM. Nesta fase é necessário organizar o conhecimento com a finalidade de apresentá-lo ao utilizador final. Esta fase, dependendo dos requisitos do projeto,

pode passar simplesmente pela elaboração de um relatório ou pelo desenvolvimento de uma interface integrada num sistema de suporte à decisão.

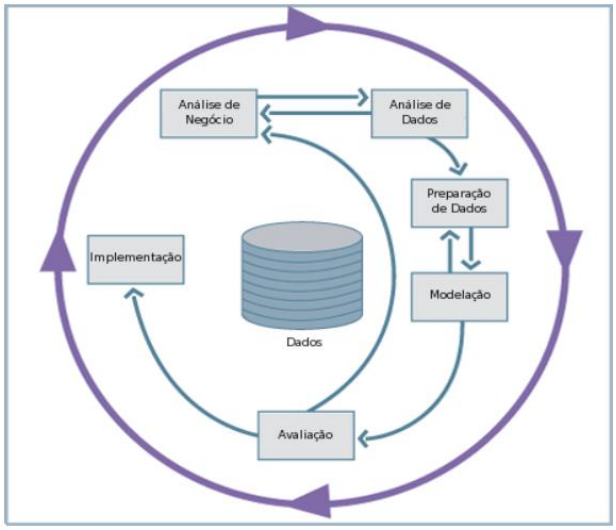

Figura 3 – As diversas fases da metodologia CRISP-DM – figura extraída de (Ribeiro, 2012).

Como se pode verificar na figura anterior, as fases possuem uma sequência lógica para o desenvolvimento ideal de um projeto de *Data Mining*. Porém, e como já foi referido anteriormente, o CRISP-DM é um processo iterativo. Por vezes é necessário fazer várias tentativas e modificações na definição do problema, nos dados, ou nos parâmetros da construção do modelo, o que leva frequentemente a voltar a fases anteriores. Estes cenários ocorrem normalmente na análise dos dados em que a dificuldade na interpretação dos mesmos obriga a reiniciar a fase de análise do negócio para uma melhor compreensão dos dados. Outra situação relativamente comum é na fase de modelação. Por vezes é necessário proceder a algumas alterações nos dados para satisfazer algumas exigências dos modelos com a finalidade de melhorar o desempenho de um modelo e consequentemente os resultados obtidos. Também é ainda comum uma terceira situação, mais dispendiosa e trabalhosa, na fase de avaliação em que os resultados obtidos não são os esperados. Com a finalidade de melhorar o desempenho de um modelo e consequentemente os resultados obtidos, é necessário voltar a analisar o problema do negócio e construir um novo modelo de *Data Mining*.

Com o objetivo de comparar as duas metodologias referidas, pode-se afirmar que ambas se baseiam na mesma base de desenvolvimento. Ambas dividem o processo de KDD em fases que se encontram interligadas entre si. Pode-se então afirmar que ambas tornam o processo de descobrimento de conhecimento um processo interativo. A metodologia SEMMA foca-se sobretudo nas caraterísticas da implementação das técnicas e do processo, enquanto a metodologia CRISP-DM é mais padronizada e ao mesmo tempo generalizada, não sendo específica para cada tipo de sectores de atividade. O CRISP-DM mantém assim uma visão mais generalizada em relação aos objetivos organizacionais do projeto em desenvolvimento.

As principais diferenças entre estas duas metodologias de *Data Mining* encontram-se desde logo na primeira fase. Enquanto a metodologia SEMMA inicia com uma recolha inicial de dados, o CRISP-DM preocupa-se na análise e entendimento do problema de negócio. É assim possível afirmar que o CRISP-DM centra-se mais na conceção real do projeto. A relação com as ferramentas existentes que as duas metodologias de *Data Mining* possuem também é bastante diferente. Se a SEMMA se encontra mais relacionado com ferramentas da SAS, a distribuição do CRISP-DM é livre e gratuita, pois foi desenvolvida como uma metodologia neutra compatível com qualquer ferramenta utilizada para o desenvolvimento do projeto de *Data Mining*. Por estes motivos e por ser uma metodologia mais estudada durante o percurso académico, optou-se pela metodologia CRISP-DM**.**

3.3 Técnicas de data Mining

A fase de modelação da metodologia adotada pressupõe a escolha de uma ou várias técnicas de *Data Mining*. Esta escolha baseia-se sobretudo no objetivo de um projeto de *Data Mining* e na melhor abordagem para a resolução do problema de negócio. Os dois principais objetivos de um projeto de *Data Mining* são a previsão e a descrição. A modelação pode assim ser descritiva ou preditiva. A modelação preditiva visa a automatização do processo de tomada de decisão. A previsão envolve o uso de todas ou algumas variáveis do conjunto de dados com a finalidade de prever valores desconhecidos para uma variável de interesse. A previsão é assim capaz de criar um modelo que permite fornecer valores de previsão ou de estimação. A modelação descritiva centra-se essencialmente em descobrir padrões descritivos dos dados. O principal objetivo é então a maximização do conhecimento e da compreensão dos dados (Han, Pei e Kamber, 2006) Alguns modelos descritivos podem ser preditivos, na medida em que é possível obter previsões de valores. Assim como um modelo preditivo pode ser descritivo pelo simples fato de que é possível interpreta-lo. A distinção entre os tipos de modelação é importante pois é necessário definir qual o objetivo principal do projeto de *Data Mining* (Fayyad et al., 1996b).

3.3.1 Modelos Descritivos

Neste tipo de modelos, os novos conhecimentos adquiridos sobre os dados durante a construção dos mesmos é o aspeto mais importante do processo. Porém os resultados obtidos nem sempre se traduzem em ações diretas, podendo acontecer que os resultados obtidos nunca venham a ser relevantes para o negócio. Os modelos descritivos mais comuns assentam em técnicas de segmentação, associação, sumarização e visualização.

Segmentação

A técnica de Segmentação, também conhecida por *Clustering*, é um dos modelos descritivos mais utilizados. Nesta técnica, o conjunto dos dados é dividido num número não limitado de segmentos/*clusters*/categorias de acordo com algumas métricas que avaliam as semelhanças naturais entre os dados. Esta técnica categoriza os exemplos segundo as suas caraterísticas o que permite descobrir conhecimento nas relações e semelhanças entre os dados. Um bom modelo de segmentação deve apresentar uma alta similaridade dentro do mesmo *cluster* e uma

baixa similaridade entre exemplos de *clusters* diferentes. Alguns algoritmos de segmentação produzem *clusters* segundo uma hierarquia, a qual se define o grau e a força de relacionamento. Alguns algoritmos de segmentação possibilitam ainda que um exemplo pertença a mais do que um *cluster*, esta caraterística produz diagramas com sobreposição de clusters (Berkhin, 2002).

Associação

Esta técnica possibilita a definição de um modelo que descreva possíveis correlações significativas entre variáveis, através da identificação de grupos de dados fortemente correlacionados. As associações detetam-se quando se verificam várias ocorrências num único exemplo. Esta técnica consiste assim em identificar e descrever associações entre variáveis no mesmo objeto ou associações entre objetos diferentes que ocorram juntamente. (Han, Pei e Kamber, 2006).

Sumarização

Esta técnica cria uma descrição compacta para um dado subconjunto de dados. Alguns algoritmos mais avançados de sumarização envolvem técnicas de visualização e a necessidade de determinar relações funcionais entre as variáveis. Os algoritmos de sumarização são frequentemente usados na análise exploratória de dados com geração automatizada de relatórios, sendo responsáveis pela descrição compacta de um conjunto de dados. A sumarização é utilizada, principalmente, no pré-processamento dos dados, quando valores inválidos são determinados por meio do cálculo de medidas estatísticas, como mínimo, máximo, média, moda, mediana e desvio padrão, no caso de variáveis quantitativas. No caso de variáveis qualitativas, por meio da distribuição de frequência dos valores (Han, Pei e Kamber, 2006).

Visualização

A visualização é a representação gráfica dos resultados finais ou intermédios do processo de *Data Mining,* utilizando-se formas visuais de fácil compreensão. O objetivo da visualização consiste em apresentar as informações através de diagramas, permitindo uma melhor representação dos padrões e tendências. Quanto melhor for a descrição de um conjunto de dados, maior é a possibilidade de o entender e de compreender o domínio em que está inserido. Uma das grandes vantagens deste tipo de técnicas é que permite evidenciar aspetos nos dados que habitualmente não seriam diretamente percetíveis (Han, Pei e Kamber, 2006).

3.3.2 Modelos Preditivos

Normalmente, os resultados do modelo são utilizados diretamente nos dados, tornando a acuidade do modelo como a medida desempenho mais importante para a sua avaliação, tanto nos de classificação como nos casos de regressão.

Classificação

A Classificação (Han, Pei e Kamber, 2006) é a técnica utilizada neste trabalho e é uma das técnicas mais utilizadas de *Data Mining*, pois é uma das mais efetuadas tarefas cognitivas realizadas pelo homem como apoio à compreensão do meio em que se insere. O ser humano classifica tudo o que entende a sua volta. Segundo o autor desta técnica, o homem, procura classificar diversas situações e acontecimentos do seu cotidiano em categorias já entendidas no passado e para os quais possui uma resposta pronta fundada (Camilo e Silva, 2009). O objetivo dos métodos de classificação é construir um modelo (classificador) capaz de prever a classe de uma nova amostra, com uma resolução satisfatória. Seja uma base de dados constituída de diversas amostras (registos), em que as amostras são descritas por atributos e cada uma delas pertence a uma classe predefinida, identificada por um dos atributos, chamado atributo rótulo da classe ou, simplesmente, classe. Em termos gerais, um processo de classificação desenvolve-se da seguinte maneira:

- 1º - Divide-se os dados em dois conjuntos: conjunto de treino e conjunto de teste. O conjunto de treino é normalmente maior do que o conjunto de teste. Quanto maior o conjunto de treino melhor será a aprendizagem do modelo.
- 2º - Executa-se o treino ou a aprendizagem, que consiste na construção de um modelo (classificador) que analisa e compara as amostras do conjunto de treino.
- 3° - Realiza-se o teste do modelo, que é a aplicação do modelo sobre o conjunto de teste. O modelo realiza a previsão da classe de cada amostra do conjunto de teste consoante os padrões e tendências descobertas na sua fase de aprendizagem. Calcula-se a percentagem de assertividade das classes preditas pelo modelo em relação as classes conhecidas do conjunto de teste. Essa percentagem é chamada de precisão do modelo para o conjunto de teste em questão.
- 4°. Se a precisão for considerada aceitável, o modelo pode ser usado para classificar amostras desconhecidas futuras, ou seja, amostras cuja classe não é conhecida.

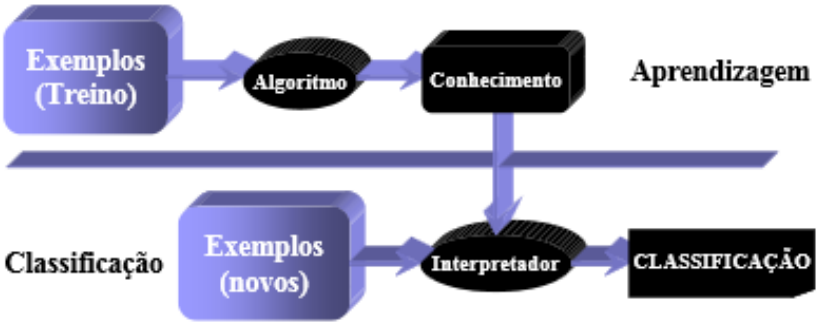

Figura 4 - Processo de aprendizagem e classificação – figura extraída de (Lobo V, 2008).

Avaliação dos Métodos de Classificação

Identificada a necessidade de resolver um problema de classificação, deve-se escolher um dos diversos modelos existentes. Para isso, pode-se comparar esses métodos conforme os seguintes critérios (Cortez, P. 2013):

- **Precisão de previsão** - é a capacidade do modelo prever corretamente a classe de amostras desconhecidas.
- **Desempenho** – é a medida dos custos computacionais envolvidos na geração e na utilização do modelo.
- **Robustez** – é a capacidade do modelo fazer predições corretas em amostras com atributos em falta ou com ruídos.
- **Escalabilidade** – é a capacidade de construir um modelo eficiente a partir de grandes quantidades de dados.
- **Fácil interpretação** – é a capacidade de tornar compreensível o conhecimento gerado pelo modelo.

Tipo de classificação

Relativamente aos métodos de classificação, o tipo de aprendizagem abordado neste trabalho é o supervisionado. Nesse sentido, dado um conjunto de exemplos rotulados por uma classe na forma $(x_i; y_i)$, em que x_i representa um exemplo e y_i indica a sua classe, deve-se produzir um classificador, também denominado por modelo, capaz de prever com precisão a classe de um novo conjunto de dados. Esse processo de execução de uma técnica de classificação a partir de um conjunto de dados é denominado treino. O classificador obtido também pode ser visto como uma função *f*, a qual recebe um dado *x* e fornece uma previsão *y*. Os rótulos ou classes representam a variável de interesse sobre a qual se deseja fazer previsões. Considera-se que os rótulos assumem valores discretos 1;2;…;k, e assim temos então um problema de classificação. Um problema de classificação no qual k = 2 é denominado binário pois é a separação de apenas duas classes. Para k > 2, configura-se um problema com multi-classes, isto é mais do que duas classes (Lorena e Carvalho, 2007).

Árvores de Decisão

Os algoritmos baseados em árvores de decisão (Hand, Mannila e Smyth, 2001) dividem o conjunto inicial de dados em vários subconjuntos separando as amostras mediante certas condições para posteriormente as classificar. Cada divisão é relativa a um atributo e segundo algumas condições de separação as amostras são divididas em subconjuntos sucessivamente. Os nós das árvores de decisão comportam testes lógicos de um determinado atributo, realizando normalmente comparações com uma constante. As divisões em subconjuntos são recursivamente aplicadas a cada um dos subconjuntos anteriores criando o formato de ramificações de uma árvore. Para cada amostra a classificar é percorrido um trajeto descendente na árvore, respeitando as condições lógicas dos nós, até que seja alcançada uma folha da árvore com a respetiva classe a ser atribuída. Este modelo de classificação é

normalmente de fácil compreensão e interpretação. As árvores de decisão possuem uma representação bastante intuitiva.

Classificadores Bayesianos

Todos os outros modelos de classificação já referidos atribuem uma classe de uma forma rigorosa e precisa para cada amostra. Os Classificadores Bayesianos (Zembrzuski, 2010), ao contrário dos outros modelos, são modelos estatísticos que calculam as probabilidades de determinada amostra pertencer a uma das classes possíveis, prevendo para a amostra a classe que obteve a maior dessas probabilidades. O cálculo das probabilidades para cada classe é feito com base no teorema de Bayes. Uma das singularidades deste modelo é a suposição de que existe independência condicional de classe, isto é, o valor de um atributo sobre uma determinada classe não é afetado pelos valores dos outros atributos. Apesar de esta suposição ser bastante contestada, a verdade é que o Classificador Bayesiano traz bons resultados na prática podendo ser comparado em desempenho, precisão e escalabilidade com classificadores como as Árvores de Decisão ou as Redes Neuronais (Han, Pei e Kamber, 2006).

Redes Neuronais

As técnicas baseadas em redes neuronais (Neumann, 1940) têm sido cada vez mais aplicadas em problemas de difícil modelação computacional ou em áreas em que um modelo matemático seria complexo demais para ser útil. As Redes Neuronais são constituídas por nodos ou neurónios. Estes nodos são unidades básicas de processamento fortemente interligadas entre si. A informação é propagada de um nodo a outro através de conexões. O conhecimento é armazenado nas conexões entre os nodos sob a forma de pesos, que não são mais do que uma condição inicial que afetará os valores produzidos pelos nodos. Assim, a determinado conhecimento corresponde uma ou mais distribuições diferentes de pesos, tornando cada rede única. Fornecendo sinais de entrada na rede, ela dará uma resposta que corresponderá à aplicação do conhecimento que ela possui. Os pesos e sua distribuição são obtidos através de um processo de aprendizagem da rede. Cada nodo recebe uma série de valores e, em função deles, determina um valor a apresentar como saída. Os valores de saída de alguns nodos são em alguns casos valores de entrada de outros nodos (Lazzarotto, Oliveira, e Lazzarotto, 2006)

Support Vector Machines (SVM)

As Support Vector Machines (Vapnik, 1979) são um modelo de classificação cada vez mais utilizado em *Data Mining*. Os algoritmos SVM são utilizados atualmente tanto para tarefas de classificação como para tarefas de regressão. Resumidamente, as SVM para classificação tem como objetivo encontrar a maior margem do híperplano que separa duas classes, sendo que os pontos que se encontram sobre as margens de separação são chamados de vetores de suporte. Sob essa teoria, as SVM foram desenvolvidas. Esse algoritmo baseia-se na ideia de que é possível melhorar a generalização de classificadores lineares encontrando um hiperplano que maximiza a distância entre os dados de classes opostas. Na sua formulação usual, uma SVM é um problema de programação quadrática. O presente trabalho baseia-se nesta técnica e em

possíveis variantes da técnica. Como tal, uma descrição mais pormenorizada será apresentada no decorrer do presente relatório (Boswell, 2002).

Regressão

O conceito de regressão foi utilizado inicialmente por Francis Galton num estudo da relação entre as alturas dos pais e filhos. Os modelos de regressão (Galton, 1911) usam variáveis de interesse ou objetivo (*label*) contínuas ou discretas. A grande diferença entre Classificação e Regressão encontra-se nesse mesmo facto. Enquanto que nos modelos de classificação o valor de previsão ou *label* é do tipo nominal, sendo que é feita uma previsão da classe para cada exemplo do conjunto de teste. Nos modelos de regressão a *label,* ou variável, de previsão é um valor real. É feita uma previsão não para determinar a classe de um determinado exemplo mas sim o próprio valor numérico da variável de interesse. A semelhança dos modelos de classificação, os modelos de regressão também seguem o paradigma de aprendizagem supervisionado. O modelo é treinado a partir de um conjunto de treino. Posteriormente é testado realizando a previsão de valores de um novo conjunto, o conjunto de teste.

3.4 Aprendizagem Máquina

O termo aprendizagem de máquina refere-se a tomada de conhecimento por parte de uma determinada máquina. Esta aprendizagem é possível graças a um processo de treino. As técnicas de *Aprendizagem de Máquina* (AM) baseiam-se num princípio denominado indução. Este princípio permite, a partir de um determinado conjunto de exemplos ou amostras, obter-se conclusões genéricas. A aprendizagem indutiva divide-se em dois tipos:

- **Aprendizagem supervisionada -** Na aprendizagem supervisionada, nomeadamente nos modelos de classificação, o conjunto de treino é formado pelas variáveis regulares e pela variável de previsão (*label*). Esta última possui valores conhecidos e a aprendizagem é realizada com essa informação. O algoritmo de classificação prevê os valores da variável de previsão do conjunto de teste a partir dessa informação obtida no conjunto de treino e aprendida pelo modelo de classificação.
- **Aprendizagem não supervisionada** - Na aprendizagem não supervisionada, o conjunto de treino não possui valores para a variável de previsão ou seja, não existem exemplos já classificados. O algoritmo de AM aprende a representar as entradas submetidas segundo um padrão de qualidade. Essas técnicas são utilizadas principalmente quando o objetivo for encontrar padrões ou tendências que auxiliem no entendimento dos dados.

Uma condição essencial para as técnicas de AM é que elas se encontrem aptas para lidar com dados incompletos ou incoerentes chamados ruídos. A técnica de AM deveria teoricamente ser resistente a ruídos presentes nos dados, descartando este tipo de exemplos na obtenção dos classificadores. É também necessário minimizar a ocorrência de *outliers* no processo de indução. Os *outliers* são amostras muito distintas e com valores muito distantes dos demais presentes no conjunto de dados. Esses dados podem ser ruídos ou exemplos incoerentes presentes no

domínio, normalmente resultante de falhas humanas ou no próprio sistema. (Baranauskas e Monard, 2000).

3.5 As Dificuldades de um Processo de Data Mining

A experiência sugere que com a aplicação de técnicas de *Data Mining* podem surgir algumas dificuldades que são típicas do processo de desenvolvimento em questão. Apresentam-se abaixo alguns dos problemas mais comuns em *Data Mining* de acordo com Fayyad et al. (1996b) e Ribeiro (2012):

- **Grande volume de dados -** Quando um conjunto de dados apresenta algumas centenas de variáveis e alguns milhões de registos torna-se difícil analisar e tratar toda essa informação. Alguns métodos comuns para resolver este problema são a seleção de atributos, p processamento paralelo ou a utilização de algoritmos computacionalmente mais eficientes.
- **Dimensionalidade -** Outro grande problema revela-se quando deparados com um demasiado grande ou pequeno número de atributos. Isso implica um problema de alta ou baixa dimensionalidade. Conjuntos de dados com muitos atributos criam dificuldades na procura de semelhanças entre os registos pelos algoritmos de *Data Mining* devido ao aumento da complexidade. Este problema aumenta também a possibilidade de se encontrarem padrões que não são relevantes para a previsão da variável de interesse. A melhor forma de tratar este problema é através da utilização de técnicas de seleção de atributos e um bom conhecimento do negócio, que permite escolher os atributos mais significativos para uma determinada tarefa. Os atributos irrelevantes são descartados.
- **Sobre ajustamento (*Overfitting*) -** É possível que o conjunto de dados apresente desvios causados por erros de medição ou fatores aleatórios. Quando tais valores se encontram presentes, ocorre um sobre ajuste para que o modelo se ajuste a estes valores erróneos. Um modelo sobre ajustado apresenta uma alta precisão quando testado com estes valores mas pode mostrar-se pouco eficiente quando testado com novos dados. Porém um modelo sobre ajustado não é uma representação ótima da realidade e deve ser assim evitado. A melhor forma de evitar este problema é a utilização de métodos de validação, tais como a validação cruzada.
- **Valores nulos e irregulares -** Algumas instâncias presentes nos conjuntos de dados podem possuir ruído, ou valores irregulares, caudados por diversos motivos, tais como erros na medição, erros humanos na inserção de valores, avaria nas ferramentas de medição, entre outros. Outro problema é o dos valores nulos, visto que muitos conjuntos de dados apresentam campos vazios. Isto acontece devido a falhas em equipamentos de medição, campos vazios em formulários preenchidos por pessoas, entre outros. Este é um grande problema, pois alguns algoritmos de *Data Mining* não têm a capacidade de lidar com valores nulos. As soluções para estes problemas passam por aplicar técnicas, tais como a remoção dos registos com valores nulos, ou a troca dos valores nulos por valores determinados pelo analista.

- **Interação do utilizador e conhecimento prévio -** Muitas das técnicas de *Data Mining* não possibilitam interatividade e não permitem a introdução de conhecimento prévio. Uma vez que a aplicação do conhecimento sobre o domínio e sobre o problema de negócio tem uma importância muito significativa em todas as fases da metodologia CRISP-DM, esta falta de interatividade e troca de conhecimento é um problema.
- **Integração com outros Sistemas -** A integração de um projeto de *Data Mining* com outros sistemas de informação é uma mais-valia e oferece muitas outras possibilidades de utilização. Algumas dificuldades passam pela integração com os SGBD, com ferramentas de visualização, folhas de cálculo e também a integração com sensores em tempo real.

Capítulo 4

Support Vector Machines

4.1 Teoria da Aprendizagem Estatística

Uma das bases das técnicas de SVM é a teoria da aprendizagem estatística, também conhecida pela teoria VC implementada por Vapnik (Vapnik, 1995). O desenvolvimento desta teoria resolve um dos problemas fundamentais nas teorias da aprendizagem, que é a criação de classificadores com grandes capacidades de generalização. Essa teoria foi um importante passo na construção de melhores classificadores. Seja *f* um classificador, isto é um mapeamento de um conjunto de padrões x_i, e *F* o conjunto de todos os classificadores que um determinado algoritmo de aprendizagem de máquina pode criar. Durante o processo de aprendizagem o algoritmo utiliza um conjunto de treino T, composto de n pares $(x_i; y_i)$, em que y_i é a classe do padrão x_i, para gerar um classificador particular $f' \in F$.

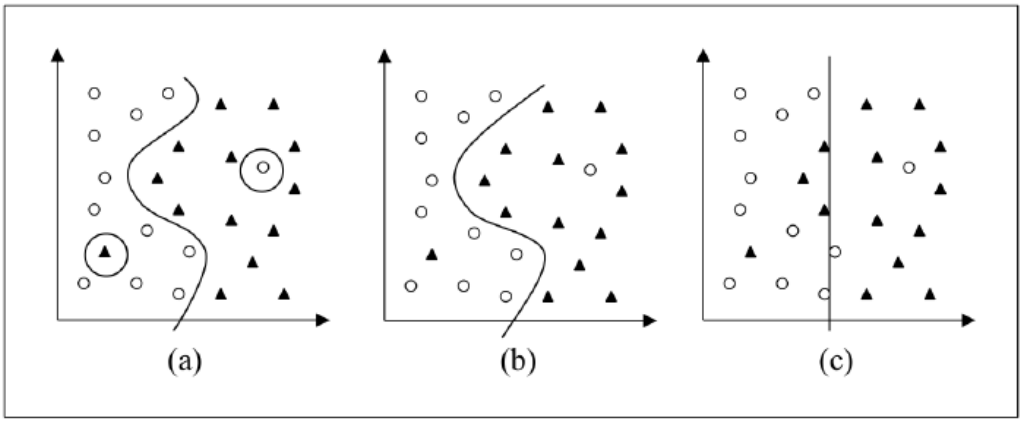

Figura 5- representação de três situações possíveis de aprendizagem de máquina – figura extraída de (Smola, 2000).

Na Figura 5 estão ilustrados três possíveis exemplos de um conjunto de treino classificado em duas classes distintas (triângulos e círculos). Como a figura demonstra foi realizada a separação das instâncias em duas classes através de uma função ou classificador. A linha que separa as duas classes é a função do classificador. Esta representa o hiperplano que separa as duas classes. Na Figura 5(a) tem-se um exemplo de uma classificação correta de todos os exemplos

do conjunto de treino, incluindo dois possíveis erros de classificação. Por ser muito específica para o conjunto de treino utilizado, esse classificador apresenta uma grande probabilidade de cometer erros quando testada com novos dados. Essa situação representa um caso de sobre-ajustamento do modelo aos dados de treino.

Na Figura 5 (c) tem-se um outro exemplo completamente oposto. O classificador não considera os pontos pertencentes a classes opostas que estejam muito próximos entre si. Este tipo de classificação comete muitos erros, mesmo para casos que podem ser considerados simples. Existe, assim, um sub-ajustamento, pois o classificador não é capaz de se ajustar mesmo aos exemplos de treino. A situação ideal seria então um meio-termo entre as duas funções descritas anteriormente (Figura 5 (b)). Esse preditor tem complexidade intermediária e classifica corretamente grande parte dos dados, sem se fixar demasiadamente em qualquer ponto individual. A Teoria de Aprendizagem Estatística visa estabelecer condições matemáticas que permitam a escolha de um classificador *f' c*om bom desempenho para os conjuntos de treino e teste. Ou seja, busca-se uma função *f'* capaz de classificar os dados de treino o mais eficazmente possível, sem dar atenção excessiva a qualquer ponto para evitar casos de *overfitting* (sobre-ajustamento). (Lorena e Carvalho, 2006)

4.1. Escolha de um Classificador

Na seleção de um classificador particular *f*, é normal selecionar a função que produz a maior assertividade durante o treino, ou seja, aquela que possui maior capacidade de classificar corretamente os padrões do conjunto de treino *T*. O erro ou risco esperado de um classificador *f* para dados de teste pode então ser quantificado pela Equação 1, a seguir apresentada, para o caso em concreto de um conjunto de treino com uma distribuição de probabilidade desconhecida $P(x, y)$ e sendo esses dados independentes e identicamente distribuídos. A aprendizagem da máquina realiza o treino sobre o conjunto e define x_i -> y. Isto considerando x como o vector de atributos dos dados de entrada e y como o valor de previsão da classe. O erro esperado da máquina é portanto definido pela seguinte equação (Burges, 1997):

$$R(f) = \int c(f(x), y) dP(x, y)$$

O risco esperado ou risco atual $R(f)$ é o valor de erro utilizado para avaliar se um classificador tem uma boa capacidade preditiva. O objetivo é então encontrar a função que minimize o risco atual. Contudo não é possível fazê-lo diretamente, uma vez que em geral a distribuição de probabilidade $P(x, y)$ é desconhecida. Seja o risco empírico de f ($Remp\ (f)$) a percentagem de classificações incorretas obtidas em *T*, em que $c(f(x_i), y_i)$ é uma função de custo que relaciona a previsão $f(x_i)$ com o valor de previsão pretendido y_i (Müller et al., 2001). Um tipo de função de custo é a "perda 0/1", definida por $c(f(xi), yi) = \frac{1}{2} |yi - f(xi)|$.O resultado desta função é 0, caso o ponto é classificado corretamente, e 1 caso contrário (Müller et al., 2001). O processo de procura de uma função *f'* que apresente o menor *Remp* é denominado Minimização do Risco Empírico e a sua equação tem a seguinte forma:

$$Remp\ (f) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2}\ c(f(xi), yi)$$

Limites no Risco Esperado

Um limite importante fornecido pela Teoria da Aprendizagem Máquina relaciona o risco esperado de uma função com seu risco empírico e ainda com um termo de capacidade. Esse limite, apresentado na seguinte inequação, é garantido com probabilidade $1-\theta$, em que $\theta \in [0; 1]$ (Burges, 1997):

$$R(f) \leq Remp(f) + \sqrt{\frac{\boldsymbol{h\left(ln\left(\frac{2n}{h}\right)+1\right)-ln\left(\frac{\theta}{4}\right)}}{\boldsymbol{n}}}$$

O valor *n* é o número de exemplos do conjunto de dados. A dimensão Vapnik e Chervonenkis (VC) apresentada pelo valor inteiro *h* é uma caraterística do conjunto de funções $f \in F$. O valor de *h* é o número máximo de exemplos do conjunto de treino que podem ser classificados. Quanto maior for o valor de *h*, mais complexas são as funções de classificação que podem ser produzidas a partir do conjunto de funções *F*. Seguindo o exemplo de separação de duas classes (classificação binária), uma dimensão VC de valor *h*, permite saber que pelo menos *h* pontos podem ser classificados. A dimensão VC trata-se, portanto, do termo que mede a noção de capacidade de classificação descrita na equação anterior. Todo o lado direito da inequação é chamado de limite do risco, já a segunda parcela do lado direito da inequação é conhecida pela confidência VC. Segundo Burges (Burges, 1997), o risco esperado $R(f)$ não pode ser calculado diretamente mas, sabendo *h*, pode ser calculado o lado direito da inequação, isto é a confidência VC (Vapnik, 1999).

Quanto maior for o conjunto de treino menor será o erro na aprendizagem e maior será a assertividade nas variáveis de previsão. Os conjuntos de dados com elevado volume de dados, como por exemplo os conjuntos tratados numa ETAR, encaixam-se perfeitamente com este tipo de atividade. Porém, é necessário lidar com tal volume de dados. Para isso projetou-se o princípio da minimização de risco empírico. A situação ideal seria obter um risco esperado muito próximo de um risco empírico baixo. O limite $n -> \infty$ prova que o erro esperado vai convergindo com o erro empírico. Analisando a inequação anterior podem-se tirar duas conclusões importantes:

- Se o valor de *n/h* for grande, o valor de *n* também é grande. Logo a confidência VC será pequena e o erro esperado fica assim mais próximo do erro empírico.
- No caso contrário com um valor de *n/h* pequeno, mesmo tendo um erro empírico baixo, não implica que o erro esperado também seja baixo.

Princípio da Minimização do Risco Estrutural

O princípio de minimização do risco estrutural, surge com o objetivo de minimizar o risco empírico e controlar a confidência VC de um conjunto de funções (Vapnik, 1999). Embora o limite representado na inequação 3 tenha sido útil na definição do princípio de minimização do risco estrutural, na prática surgem alguns problemas. Primeiramente calcular a dimensão VC de uma classe de funções é geralmente uma tarefa complexa. Para além disso existe o problema

que o valor de *h* pode ser desconhecido. O erro empírico calcula o erro relativo a aprendizagem de uma função de previsão f em particular. Para minimizar o erro empírico é necessário particionar *F* em vários subconjuntos. Considerando esses subconjuntos, $F_0 \subset F_1 \subset \ldots \subset F_q \subset F$, em que com o aumento do índice também aumenta a dimensão VC $h_0 < h_1 < \ldots < h_q < h$, e consequentemente a confidência VC. Então, sendo f_k a função com menor erro empírico, à medida que *k* aumenta o erro empírico diminui. Isso deve-se ao facto do número de exemplos e consequentemente a complexidade da máquina também aumentar. Porém, o termo de capacidade h também aumenta com *k*. Pode-se então concluir que se deve encontrar um ponto ideal no qual se obtém o valor mínimo possível da soma do risco empírico com a confidência VC, ou seja, o mínimo erro esperado ($R(f)$).

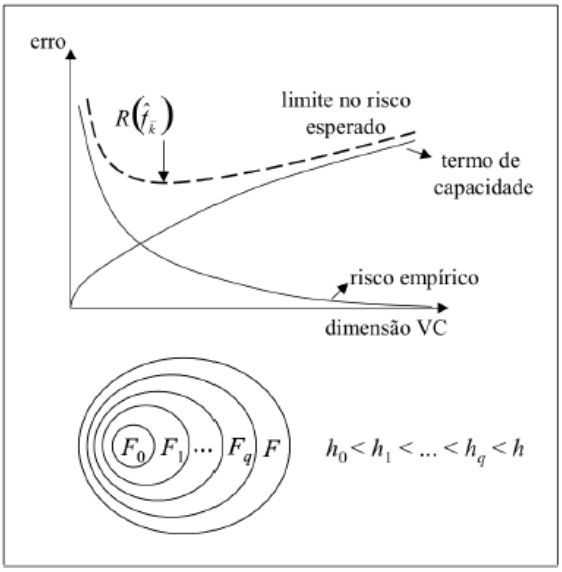

Figura 6 - Princípio de minimização do risco estrutural extraído de (Lorena e Carvalho, 2006).

As técnicas SVM que serão apresentadas de seguida implementam esse princípio estatístico para controlar a capacidade e prevenir sobre ajustamentos. No entanto, e como seria de imaginar, esta é apenas uma das teorias que fundamentam a técnica.

4.2. Otimização Operacional

As técnicas SVM estão envolvidas com a otimização operacional. A otimização operacional é uma ciência que fornece instrumentos quantitativos no processo de decisão. É uma técnica importante para determinar a melhor utilização de recursos e otimizar processos, sendo bastante empregada nas indústrias. Entre o conjunto de métodos de otimização da pesquisa operacional encontram-se a programação linear e não-linear, que envolve a programação quadrática convexa, Teoria de Lagrange e Dualidade. Existem dois tipos de otimização:

- **Otimização Não Restrita -** A otimização não restrita engloba os problemas em que as variáveis podem assumir qualquer valor, sem qualquer tipo de restrição.

- **Otimização Restrita** - A otimização restrita abrange todo o tipo de problemas em que as variáveis podem assumir apenas alguns valores condicionados pelas restrições. Existem dois tipos de restrições que podem ser de *igualdade* e *desigualdade*. As restrições de igualdade detalham normalmente a operação realizada pelo sistema. As restrições de desigualdade restringem as variáveis segundo limites inferior e/ou superior (Santos, 2002).

Programação Linear

Consoante algumas restrições lineares de igualdade ou de desigualdade, as situações reais podem ser descritas por uma função objetivo. Essa função objetivo deve ser maximizada ou minimizada para satisfazer essas restrições. Constitui-se assim um problema da Programação Linear (Ales, 2008). A forma padrão é representada por:

$$\text{Minimizar } f(x)$$
$$x \in D$$

Na qual $f(x)$ é uma função linear e *D* é o conjunto das restrições também lineares. As condições de Karush-Kuhn-Tucker (KKT) são as condições de otimização de um Problema Linear. Tratando-se de um problema linear, as condições de primeira ordem que permitem a convexidade bastam para garantir a existência de um ótimo global.

Programação não Linear

Quando a função objetivo e/ou as funções de restrição são não-lineares. Uma generalização do problema tem a seguinte forma:

$$\text{Minimizar } f(x)$$
$$x \in D$$

Em que $f(x)$ pode ser uma função não linear e *D* o conjunto das restrições. Esse problema possui uma solução global garantida pelo teorema a seguir:

> ***Teorema de Weierstrass*** **-** Sejam $D \in \mathbb{R}^n$ um conjunto compacto não vazio e $\boldsymbol{f}: \boldsymbol{D} \rightarrow \mathbb{R}$ uma função contínua. Então o problema de minimizar a função $\boldsymbol{f}(\boldsymbol{x})$com $x \in D$ possui solução global. Um problema não linear pode ser resolvido através de vários métodos, como por exemplo, Multiplicadores de Lagrange, Método de Newton, Método do Gradiente, entre outros. As caraterísticas de cada método diferem entre si e possuem diferentes tipos de convergência.

Programação Quadrática

Situações gerais de programação quadrática possuem uma função objetivo quadrática e estão sujeitos as restrições lineares ou quadráticas. Neste caso, a forma da função objetivo é:

$$f(x) = k^T w + \frac{1}{2} x^T Q w$$

Para algum vetor *k*, *w* é a variável desconhecida, *T* é o expoente e *Q é* uma matriz simétrica.

Programação Quadrática Convexa

Um dos tipos mais específicos de problemas da programação quadrática trata situações quadráticas convexas, como é o exemplo das técnicas de SVM. O principal teorema da programação convexa é: *"Um qualquer mínimo local de um problema de programação convexa é um mínimo global*."

A programação quadrática convexa divide-se, ainda, em dois tipos dependendo se o tipo de problema é primal ou dual. Para efetuar a pesquisa de máximos e mínimos condicionados, como nos problemas quadráticos convexos e demais problemas de otimização, é necessário utilizar o método de Lagrange. Esse método é usado para solucionar o treino de SVM.

Condições de Kuhn-Tucker (KKT)

As condições de kuhn-Tucker têm como objetivo reconhecer quando foi atingida a solução ótima para o problema. A função objetivo na maioria dos casos é não linear e sujeita as restrições. Como tal, não é suficiente analisar a derivada. As condições para a otimização são:

- A função objetivo deve ser convexa para a minimização ou côncava para a maximização.
- As restrições do problema devem delimitar um conjunto convexo.

Teoria de Lagrange

A teoria de Lagrange tem como objetivo caraterizar a solução de um problema de otimização quando normalmente não existem restrições de desigualdade. Este método foi desenvolvido por Lagrange em 1797 e é uma generalização do teorema de Fermat de 1692. Em 1951, Kuhn e Tucker adaptaram o método para permitir restrições de desigualdade.

Exemplificação do Teorema de Fermat

Uma condição necessária para x_i ser um mínimo de $f(x)$, $f \in C_1$, é que a derivada da função seja nula nesse ponto. Como tal,

$$\frac{\partial f(x_i)}{\partial x} = 0$$

Caso existam e sejam contínuas $\partial_1 f$ e $\partial_2 f$, considera-se uma função f pertencente a uma classe C_1 . Ou seja, se uma função é convexa e de classe C_1 basta encontrar o ponto x_i que satisfaça a condição acima, isto é, com derivada igual à zero, e este ponto será o resultado da minimização da função. Em problemas com restrições, é preciso definir a função de Lagrange, a qual engloba informações sobre as restrições e a própria função objetivo. O conceito da função de Lagrange define-se como a função objetivo somada

com a combinação linear das restrições, onde os coeficientes da combinação linear são chamados de Multiplicadores de Lagrange (Santos, 2002). Dado um problema de otimização com função objetivo $f(x)$ e restrições de igualdade $h_i(x)$ = 0, i = 1, …, m, a função de Lagrange é definida como:

$$L(x, \alpha) = f(x) + \sum_{i=0}^{m} \alpha_i h_i(x)$$

Em que os coeficientes α_i são os Multiplicadores de Lagrange. A partir dessa definição é possível apresentar o teorema de Lagrange.

Teorema de Lagrange:

Uma condição necessária para um ponto x_i ser um mínimo de $f(x)$ sujeito à restrições $h_i(x) = 0, i = 1, \dots, m$, com $f \in C_1$, é:

$$\frac{\partial L(x_i, \alpha_i)}{\partial x} = 0$$
$$\frac{\partial L(x_i, \alpha_i)}{\partial \alpha} = 0$$

para qualquer valor α_i. Essas condições são suficientes para afirmar que um determinado ponto é mínimo de $f(x)$ desde que a função $L(x, \alpha_i)$ seja uma função convexa de x. Verificando-se essas duas condições nos sistemas solucionados a partir delas, é alcançada a solução global. Os dois teoremas descritos até o momento, abrem caminho para o caso mais geral, o Teorema Kuhn-Tucker, no qual o problema de otimização contém tanto restrições de igualdade quanto de desigualdade (Santos, 2002).

Dualidade

A teoria de dualidade baseia-se na associação de um problema original (primal) com um outro problema, denominado dual, que sob certas condições é equivalente ao primal e que, por vezes é mais fácil de ser resolvido. As relações de dualidade são muito úteis na teoria e nas técnicas computacionais.

A dualidade mais forte é obtida em problemas primais de minimização em que a função é convexa, o que se torna mais conveniente obter a função de Lagrange. Dado um problema de otimização Primal (P) é possível encontrar um problema Dual (D) relacionado e do mesmo tipo que P, e em que os Multiplicadores de Lagrange de P são parte da solução de D, e os Multiplicadores de Lagrange de D estão contidos na solução de P. Então concluindo, se y_i é a solução do problema D, a solução do problema P pode ser determinada a partir de y_i (Santos, 2002)

4.3. A Técnica de *Support Vector Machines*

4.3.1. Análise do Modelo de *Support Vector Machines*

***Support Vector Machines* para Classificação**

Os resultados obtidos com esta técnica, relativamente recente, têm despertado a atenção de muitos analistas. Esta técnica obtém altos índices de assertividade, permitindo criar modelos de situações não lineares e complexas gerando modelos simples e de fácil interpretação. Pode ser usada para relações lineares e não lineares, entre outras. É utilizado tanto para tarefas de classificação como de predição. Atualmente um dos problemas da técnica SVM, e que tem sido um dos focos de várias pesquisas, é o tempo utilizado para a aprendizagem dos conjuntos de dados.

O objetivo das SVM é receber dados de fontes que contem os valores dos atributos *(x,y)* de cada ponto, descobrir a sua classificação (0 e 1), e encontrar uma reta (hiperplano) que traga a maior margem de separação possível entre as duas classes (Mangasarian, 2003).

Linearmente separável

As técnicas SVM lineares definem fronteiras lineares a partir de dados linearmente separáveis. Seja *T* um conjunto de treino com n dados x_i e as suas classes $y_i \in Y$, em que *X* constitui o conjunto dos dados e Y = {-1,+1}. O conjunto *T* é linearmente separável se for possível separar os dados das classes +1 e -1 por um hiperplano.

Dado um conjunto de dados:

$$(y_1; x_1) \dots (y_i; x_i)\ \ y\ \in Y = \{-1; +1\}$$

Os dados linearmente separáveis podem ser classificados linearmente pela função real $f : X \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ da seguinte forma: o conjunto de dados de treino $X = (x_1, \dots, x_i)$ é considerado da classe +1 se $f(x) \geq 0$, e da classe -1 se $f(x) \leq 0$. Assim, a função do hiperplano de decisão é representada da seguinte forma:

$$f(x) = w^t.x + b$$

na qual $w\ \in\ \mathbb{R}^n$ é o vetor de pesos, e $b\ \in\ \mathbb{R}$ é a constante chamada bias. A classificação de cada padrão x do conjunto de treino é atribuída consoante a distância em relação às margens do hiperplano separador. Ou seja será classificado como -1 se estiver mais próximo da margem negativa:

$$w^t.x + b = -1$$

e será classificado como +1 se estiver mais próximo da margem positiva:

$$w^t.x + b = +1$$

Os dados consideram-se corretamente classificados se estiverem fora da margem de separação da sua classe. Então considera-se corretamente classificado se respeitar a seguinte condição:

$$w^t x_i + b \geq 1, \text{se } y_i = +1$$

$$w^t x_i + b \leq 1, \text{se } y_i = -1$$

A bias, representada pela letra b representa o parâmetro permite que o hiperplano separador se posicione no local correto para a separação dos pontos.

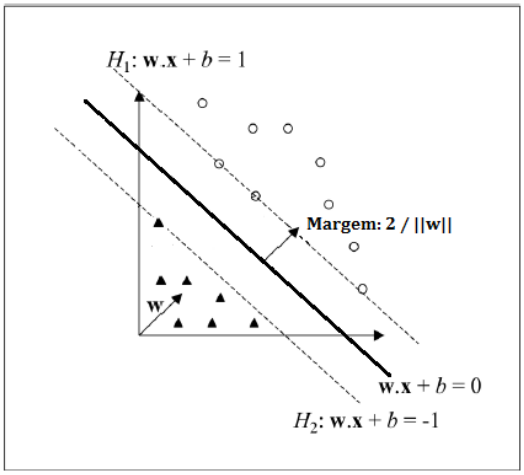

Figura 7 - Representação da separação de duas classes e hiperplano ótimo – figura extraída e adaptada de (Lorena e Carvalho, 2006).

A margem representa a distância entre os pontos de cada classe mais próximo do hiperplano. Quanto maior a margem que separa as duas classes maior é a assertividade da classificação, pois mais diferentes são os pontos de cada classe. Uma das principais ideias do SVM é, então, construir um hiperplano de separação ótimo no espaço de caraterísticas, de modo que a distância entre as diferentes classes pode atingir o valor máximo. O SVM visa assim encontrar a maior margem de separação entre os pontos de cada classe. Os vetores de suporte são os pontos que ficam sobre as linhas representativas das fronteiras e compõem a margem de separação. O objetivo é encontrar o híperplano ótimo que tenha a margem com máxima distância de cada classe, para isso torna-se necessário encontrar os valores mínimos da função sob algumas restrições. Esta afirmação leva então a ideia de que se está perante um problema da programação quadrática, com a seguinte formulação para a minimização (Burges, 1998):

$$\underset{w,b}{\text{Minimizar}} \frac{1}{2}||w||^2$$

Com as seguintes restrições:

$$y_i(w.x_i + b) - 1 \geq 0, \forall\, i = 1, \ldots, k$$

As restrições permitem assegurar que não existem pontos entre as margens de separação das classes. Este tipo de SVM possui, assim, margens rígidas. Caso todos exemplos do conjunto de treino sejam linearmente separáveis, a utilização da programação quadrática convexa permite encontrar o hiperplano com maior margem possível. A separação das classes seria assim idealmente linear, porém em muitos casos os pontos não são separáveis linearmente. Para esses casos é necessário utilizar funções de Kernel. Quando os dados são mais complexos nem sempre são linearmente separáveis, para poder classificar este tipo de dados surgiram as SVM com margens suaves (Burges, 1997).

Para os casos em que os dados não são linearmente separáveis, é necessário introduzir variáveis que permitem relaxar as restrições dando alguma folga ou tolerância ao hiperplano. Isto permite que alguns pontos permaneçam dentro das margens e que aceite alguns casos de exemplos classificados incorretamente. Sendo assim, as restrições suavizadas são representadas com uma nova variável, ξ_i , $= 1, \dots, k$:

$$w.x_i + b \geq +1 - \xi_i \text{ para } y_i = +1$$
$$w.x_i + b \geq -1 + \xi_i \text{ para } y_i = -1$$
$$\xi_i \geq 0 \quad \forall\, i$$

Adicionalmente, com a finalidade de permitir representar o custo extra para os erros, resultante do acréscimo das variáveis de folga, existe a necessidade de mudar a função objetivo, ou de decisão, a ser minimizada para:

$$\frac{1}{2}||w||^2 + C(\sum_{i=1}^{n} \xi_i)^k$$

A componente *C* é uma constante que opera como uma função de correção e previne que outliers influenciem o hiperplano ótimo. O parâmetro *C* é definido pelo utilizador.

$C > 0$ relaciona o termo de confiança com o risco empírico. Quanto maior o valor de *C* assume-se que maior será a correção para os erros. Por ser um problema da programação convexa, o valor de *k* pode ser qualquer inteiro positivo. Nomeadamente, se *k* = 1 ou *k* = 2 então o problema também é de programação quadrática, para além de convexa (Santos, 2002).

Funções de Kernel

As funções de kernel proporcionam um meio de mapear dados de um conjunto *T*, de um espaço *X* para outro espaço *Z* com um número maior de dimensões. A finalidade deste mapeamento é encontrar um espaço *Z* em que exista separação linear dos dados, fazendo com que as classificações lineares em *Z* correspondam a classificações não lineares em *X*. O mapeamento é feito através das funções de Kernel (Gordon, 2004). É inicialmente uma caraterística de pré-processamento que permite alterar a representação dos dados da seguinte forma:

$$x = (x_1, \dots, x_n) \mapsto K(x) = (K_1(x), \dots, K_n(x))$$

Este método representa o mapeamento do espaço de entrada X para um novo espaço $Z = \{K(x)|x \in X\}$ em que K_i são as funções de kernel. Dada essa representação mapeada, uma classificação ou regressão linear em *Z* pode facilmente ser realizada. Porém, e como já foi visto, a Teoria da Aprendizagem Estatística considera a alta dimensionalidade um problema que torna a aprendizagem mais difícil e complexa. Ora à medida que a dimensão aumenta, o problema de estimação torna-se mais complexo. No entanto, a própria Teoria da Aprendizagem Estatística refere que a aprendizagem no espaço *Z* pode ser mais simples, isto é, regras de decisão mais simples são geradas pois as classificações tornam-se lineares. Resumindo, embora a dimensionalidade do espaço aumente em *Z*, a complexidade diminui. Isso porque a classificação, que no espaço de entrada *X* era apenas possível não linearmente passa no espaço *Z* a ser uma classificação linear com apenas um hiperplano (Santos, 2002).

A utilização das funções de kernel permite a implementação de máquinas de aprendizagem com diferentes tipos de superfícies de decisão não linear no espaço de entrada. A tabela seguinte apresenta as funções de kernel mais usadas (Gordon, 2004)

Tipo de Kernel	**Função** $K(\mathbf{x}_i, \mathbf{x}_j)$
Polinomial	$(\delta(\mathbf{x}_i \cdot \mathbf{x}_j) + \kappa)^d$
Gaussiano	$\exp(-\sigma \|\mathbf{x}_i - \mathbf{x}_j\|^2)$
Sigmoidal	$\tanh(\delta(\mathbf{x}_i \cdot \mathbf{x}_j) + \kappa)$

Tabela 4 - Tipos de funções kernel.- figura extraída de (Lorena e Carvalho, 2006).

***Support Vector Machines* para Regressão**

As técnicas de SVM para problemas de regressão, também chamados Support Vector Regression (SVR), possuem a mesma base de funcionamento que as SVM para problemas de classificação. Enquanto as SVM tem como objetivo a previsão da classe das instâncias de um determinado conjunto de teste, as SVR tem como objetivo prever os valores reais das variáveis de interesse com um desvio máximo definido pelo utilizador. O método SVR tem o mesmo objetivo de maximização da margem que as técnicas de SVM. Também utilizam variáveis de folga e as funções de kernel para os casos onde os dados não são separáveis linearmente. Qualquer técnica de regressão, incluindo as SVR, tem uma função de perda. A função de perda (Loss Function) tem como objetivo estimar o desvio entre a função real e a função encontrada. Existem diversos tipos de funções de perda, como por exemplo, a função de perda ε-insensitive que é a mais utilizada. Esta função de perda permite limitar o desvio máximo de uma função. O valor atribuído a ε é o valor limite da diferença entre o valor real e o valor previsto de todo o conjunto de dados. A seguir é apresentado um exemplo da representação de um problema de SVR com dados linearmente separáveis e outro exemplo com dados não linearmente separáveis. (Smola e Scholkopf, 2003)

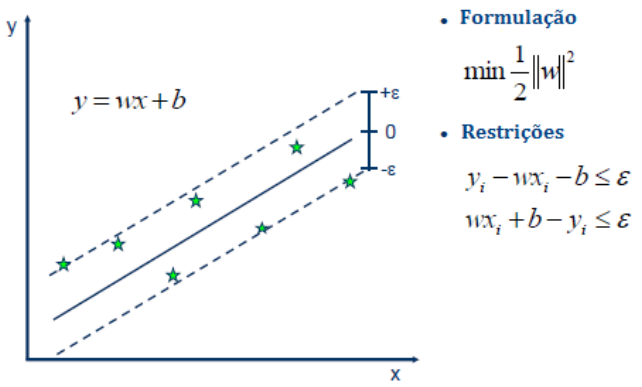

Figura 8 - Representação do hiperplano para um caso de separação linear e funções de minimização e restrições – figura extraída e adaptada de (Sayad, 2010-2014).

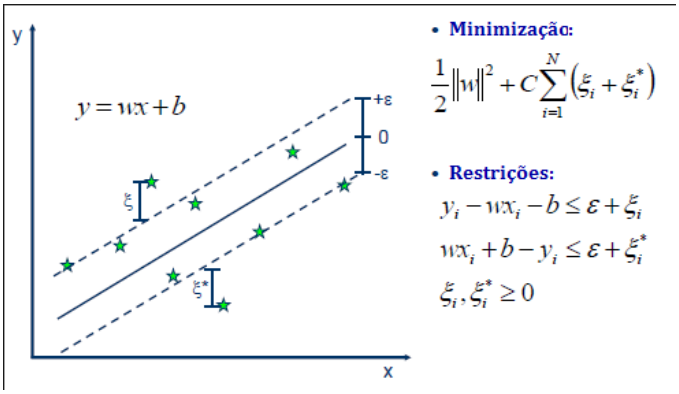

Figura 9 - Representação do hiperplano separador com variáveis de folga, função de minimização e restrições – figura extraída e adaptada de (Sayad, 2010-2014).

4.3.2. Vantagens das *Support Vector Machines*

As principais vantagens das Support Vector machines (SVM), segundo Smola, Scholkopf e Muller (1999), são:

- **Boa capacidade de generalização -** A capacidade de generalização de um classificador é obtida pela sua eficiência na classificação de dados que não pertençam ao conjunto utilizado na sua execução de treino. Quando o preditor se torna muito especializado no conjunto de treino, é chamado de *overfitting*.
- **Robustez em grandes dimensões:** Diferentemente das técnicas para gerar classificadores mais comuns, as SVM não causam *overfitting* quando usadas em objetos de grandes dimensões;
- **Convexidade da função** - Para encontrar a solução ótima do classificador é usada uma função quadrática, em que não existe presença de vários mínimos locais, e sim apenas um mínimo global, o que permite encontrar com mais facilidade o valor ótimo;

- **Teoria bem definida -** A técnica das SVM está bem fundamentada em teorias da matemática e estatística.

Capítulo 5
Data Mining Incremental

As grandes organizações cada vez se preocupam mais com a exploração e a análise dos milhões de registos que possuem as suas bases de dados. O objetivo de extrair conhecimento útil para a evolução e desenvolvimento da organização é diretamente relacionado com técnicas de *Data Mining*. As melhorias tanto a nível organizacionais como operacionais que estas técnicas podem trazer para as organizações são imensas.

Organizações como por exemplo as ETAR que diariamente armazenam grandes volumes de dados necessitam de métodos eficazes, assertivos e com um desempenho computacional excelente. Por isso, as próprias técnicas de *Data Mining* são alvas diárias de estudos que as possam aperfeiçoar. Como já foi referido, as ETAR, que são infraestruturas cada vez mais importantes, realizam leituras diárias da qualidade das águas residuais. A composição das águas residuais pode sofrer variações frequentes. Isto requer um algoritmo de *Data Mining* com um nível de desempenho suficientemente bom para que se possa atualizar os padrões frequentemente. As constantes alterações que ocorrem ao longo do tempo nos dados tornam os modelos obsoletos, desatualizados e por vezes erróneos o que pode afetar o funcionamento das organizações. As técnicas de *Data Mining* convencionais provaram ser ineficientes, uma vez que necessitam de ser reexecutadas descobrir os novos padrões envolvidos nos novos dados. Porém, este processo faz o redescobrimento desnecessário e demoroso de padrões já extraídos em iterações anteriores dos modelos. (Cavalcanti, 2005)

As técnicas de *Data Mining* incremental surgiram com o objetivo de evitar essa reexecução do algoritmo para atualizar os padrões quando novos dados (incrementais) são adicionados ou dados antigos são removidos. Assim, é possível assegurar uma maior eficiência e desempenho aos processos de *Data Mining*. (Laskov, Geh e Kruger, 2006)

Resumindo, a vertente incremental dos processos de *Data Mining*, onde o sistema considera os exemplos um a um, labora com respostas a cada novo exemplo de treino armazenado. Ou seja, após o primeiro exemplo ser classificado, o mesmo é tomado como exemplo para construir uma determinada hipótese. Depois considera-se um segundo exemplo, podendo fazer mudanças na primeira hipótese, apenas baseando-se em como esta classifica o segundo exemplo, de forma a fazer as modificações nos padrões descobertos, à medida que mais exemplos são apresentados. Algumas vantagens deste algoritmo é a possibilidade de ser

atualizado a cada nova observação, isto é, em tempo real bem como a sua utilização em conjuntos de dados relativamente volumosos. (Laskov, Geh, Kruger, 2006)

5.1. Support Vector Machines Incremental

As técnicas tradicionais de SVM requerem que seja feita uma nova execução com o conjunto total de dados, tendo-se que reprocessar todo o conjunto sempre que haja uma alteração nos dados. As SVM no fim da sua execução identificam um conjunto de pontos, chamados vetores de suporte, que representam uma pequena parte do total de pontos que compõem o conjunto de execução. Esses vetores de suporte servem de margem para a separação de duas classes.

A seleção dos pontos que constituam os vetores de suporte é uma caraterística das SVM que permite estudar a abordagem incremental da execução do treino do SVM. Ou seja, é possível obter o mesmo nível de desempenho com um número reduzido de dados. A reutilização de resultados anteriores, proposta pela técnica de SVM incremental, torna os processos de aprendizagem sucessivos dos padrões mais rápidos e também pode reduzir o custo de armazenamento descartando conjuntos de pontos antigos.

Um algoritmo incremental de SVM é a solução ideal para grandes conjuntos de dados com constante alteração e acréscimo de registos provenientes das diversas fontes de dados. Com este algoritmo é possível melhorar a eficiência do processamento dos dados em larga escala. O modelo do algoritmo incremental das SVM é parecido com o modelo das SVM padrão. O resultado obtido em ambas as variantes das SVM são relativamente semelhantes. Este método incremental permite obter os resultados precisos obtidos pelas SVM com um tempo de processamento eficazmente reduzido. Não necessita interromper o processamento das consultas feitas pelos utilizadores enquanto a atualização é realizada. O tempo de processamento é independente do tamanho total do conjunto de dados (Diehl e Cauwenberghs, 2003).

5.1.1. Técnicas de SVM incremental

Segundo o método proposto por Xiao, Wang, Zhang (2000) este processo pode ser realizado da seguinte forma: primeiro o classificador antigo é utilizado no novo conjunto de exemplos incrementais e aqueles que forem classificados incorretamente são combinados ao novo conjunto de vetores de suporte para construir um novo conjunto de treino. Os exemplos corretamente classificados não precisam de voltar a ser processados e formam um conjunto de testes. A seguir, um novo classificador é executado no novo conjunto de treino, e o novo conjunto de testes é utilizado para repetir a operação anterior. Este processo é repetido sucessivamente até que todos os pontos sejam classificados corretamente. (Fung e Mangasarian, 2002). Algumas medidas podem ser tomadas para reduzir o custo de armazenamento e acelerar a convergência: os pontos que nunca são selecionados como vetores de suporte são descartados gradualmente, e os pontos que aparecem frequentemente no conjunto de vetores de suporte são introduzidos de forma otimizada no conjunto de treino.

Outros autores laboram também nesse sentido, mas removendo alguns pontos. A aprendizagem incremental proposta por Syed et al (1999) funciona da seguinte forma: dado um subconjunto inicial *A*, executa-se o SVM o que permite encontrar os vetores de suporte. Esses vetores de suporte são adicionados a outro subconjunto *B* para ser executado novamente pelo SVM para encontrar novos vetores de suporte. E assim sucessivamente. O problema desta abordagem é que considera que os subconjuntos dos pontos selecionados são representativos do total, em que a distribuição acompanha também o universo completo. Esta condição nem sempre é verdadeira. Neste processo iterativo obtém-se vetores de suporte a cada nova iteração. Esses pontos são muito poucos quando comparados ao total de pontos do subconjunto, assim, a sua importância para o surgimento de novas margens de separação também será reduzida, caso a distribuição seja diferente.

Para diminuir a utilização de memória, Domeniconi e Gunopulos (2001) propuseram um método de aprendizagem incremental. Este método remove os pontos que não são vetores de suporte a cada passo da execução do SVM. O facto de remover esses pontos pode tornar-se um problema visto que ao acrescentarmos novos dados nas sucessivas iterações – os pontos removidos poderiam tornar-se vetores de suporte. A variante incremental também tem que se preocupar com os futuros vetores de suporte.

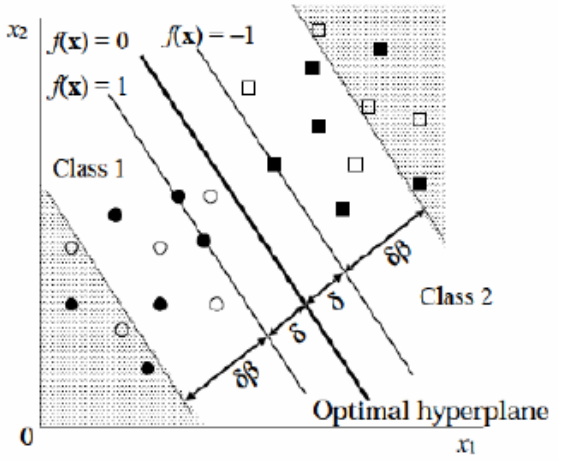

Figura 10- Gráfico que representa a execução do SVM. Separação de duas classes - figura extraída de (Kinto, 2010).

Na Figura 10 podemos ver que os pontos encontram-se separados linearmente. Os pontos que serão removidos encontram-se na área sombreada. O processo incremental utiliza um subconjunto de pontos (pontos preenchidos) para obter os vetores de suporte. Depois de executado, novos pontos são adicionados (pontos com fundo vazio). O hiperplano que separa as duas classes é representado pela função $f(x)$ e a margem entre as classes é *2*δ*. É no parâmetro *β* de ajuste (manual) que se definem os pontos, antigos e novos, a serem excluídos da execução do novo subconjunto composto pelos vetores de suporte. As retas que definem os limites das classes que correspondem as margens são $f(x) = -1$ e $f(x) = 1$, sendo a função do hiperplano $f(x) = 0$. Dependendo do conjunto de pontos selecionado são obtidos diferentes configurações para a separação das classes e diferentes hiperplanos.

A técnica SVM incremental, aqui proposta, trabalha com subconjuntos de pontos. O gráfico da Figura 11 mostra o hiperplano ótimo dos pontos com fundo preenchido. Os pontos com fundo branco são provenientes de um outro subconjunto e são os novos elementos que serão introduzidos na iteração seguinte.

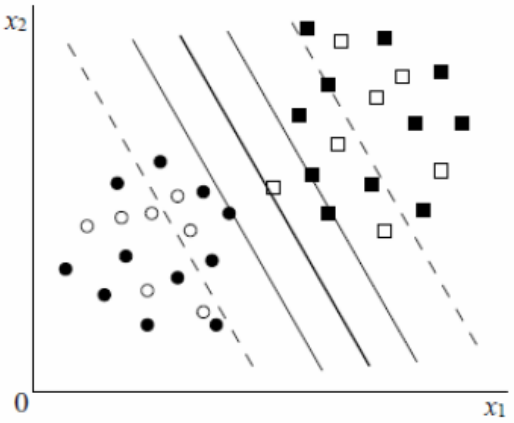

Figura 11 - Gráfico que representa a separação das classes realizada em várias iterações do SVM - figura extraída de (Kinto, 2010).

Por sua vez, Katagari e Abe (2006) tentaram resolver o problema da rotação do hiperplano que separa as duas classes. Estima-se que os vetores de suporte candidatos estão sempre próximos do hiperplano e também da região que inclui os vetores de suporte de uma iteração.

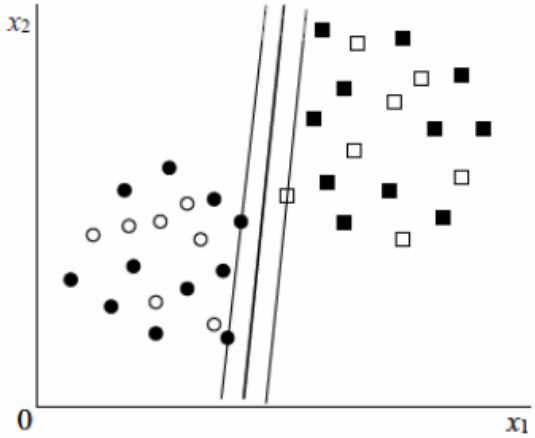

Figura 12 - Rotação do hiperplano para uma separação ótima das classes - figura extraída de (Kinto, 2010).

A Figura 12 ilustra a segunda iteração do SVM com introdução dos pontos de fundo branco. De reparar a análise que foi referida anteriormente. Mesmo que haja rotação do hiperplano, os pontos candidatos a vetores de suporte continuam a estar próximos do hiperplano.

Além dos autores anteriores, Katagiri e Abe (2006) propuseram uma forma para resolver o problema da remoção dos vetores de suporte de um conjunto prévio de execução do SVM. Quando se adicionam novos pontos, a distribuição cria uma rotação dos hiperplanos que separaram as classes. Para tal, é criada uma hiperesfera para cada um dos lados da classificação. Duas hiperesferas de maior volume são criadas inicialmente contendo todos os pontos de cada classe. De seguida é criada uma hiperesfera concêntrica de raio menor. Depois o híper-cone é criado cujo ângulo está localizado no centro da hiperesfera e que abre em direção oposta ao hiperplano. Todos os pontos que se encontram dentro da hiperesfera de menor raio, assim como do hipercone são removidos do conjunto de execução do SVM incremental. Este método de SVM incremental com hiperesferas de raios R1 e ρR1, e hipercone criado usando-se um ângulo θ ajustável.

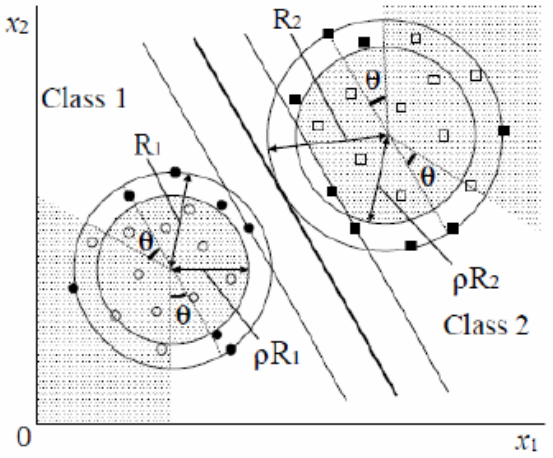

Figura 13 - Criação das Hiperesferas e do hipercone que definem os pontos a serem descartados (zona sombreada).- figura extraída de (Kinto, 2010).

A Figura 14 ilustra o processo de aprendizagem incremental proposto por Syed et al., (1999). A divisão do conjunto completo em subconjuntos de pontos. Para cada subconjunto executa-se o SVM obtendo-se um conjunto de vetores de suporte. Os resultados obtidos por este método mostraram que o modelo incremental consegue uma elevada precisão pelo menos tão boa quanto o processo batch tradicional que considera todos os pontos de uma só vez.

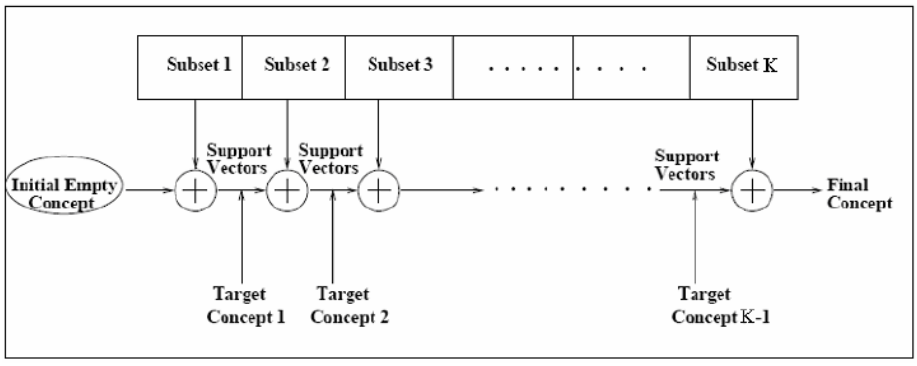

Figura 14 - Representação das várias iterações e execuções do algoritmo SVM para cada subconjunto de dados - figura extraída de (Kinto, 2010).

Gert Cauwenberghs e Tomaso Poggio (Cauwenberghs e Poggio, 2003) consideram que os métodos que propõem a realização do treino de novos dados descartando os pontos antigos que não são vetores de suporte apenas retornam resultados aproximados. Estes autores propuseram um método interativo para a construção de modelos recursivamente, um ponto de cada vez. O objetivo é manter as condições de Kuhn-Tucker (KKT) em todos os dados anteriormente vistos, enquanto "adiabaticamente" um novo ponto é adicionado à solução.

Condições de Kuhn-Tucker

Nos métodos de classificação utilizando a técnica de SVM a função de separação ótima reduz-se a uma combinação linear de funções de kernel sobre o conjunto de treino, $f(x) = \sum_j \alpha_j y_j K(x_j, x) + b$, com os dados de treino x_i e as suas correspondentes classes $y_i = \pm 1$. As condições de Kuhn-Tucker definem-se da seguinte forma:

$$g_i = \frac{\partial W}{\partial \alpha_i} = \sum_{j=1}^{N} Q_{ij}\alpha_j + y_i b - 1 \begin{cases} > 0 & \alpha_i = 0 \\ = 0 & 0 \le \alpha_i \le C \\ < 0 & \alpha_i = C \end{cases}$$

$$h = \frac{\partial W}{\partial b} = \sum_{j=1}^{N} \alpha_j y_j \equiv 0$$

O conjunto de treino fica assim dividido em 3 categorias de pontos, baseadas nas regras derivativas em g_i:

- Vetores de reserva (*R*), valores que excedem a margem ($g_i > 0$);
- Vetores de margem (*S*), valores que se encontram sobre a margem ($g_i = 0$);
- Vetores de erro (*E*), pontos que se encontram dentro das margens ($g_i < 0$).

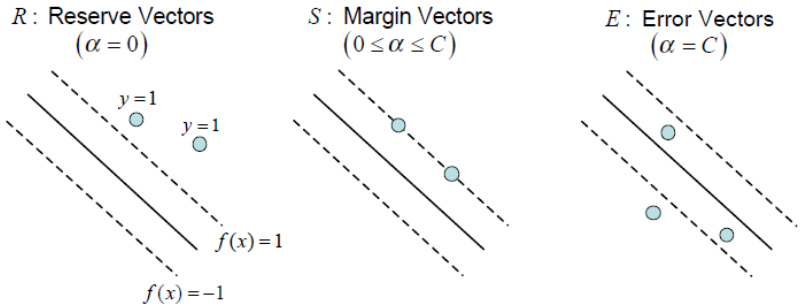

Figura 15 - representação das três situações possíveis de categorias de vetores – figura extraída de (Diehl e Cauwenberghs, 2003).

Durante a aprendizagem incremental, os novos dados de treino com $g_i > 0$ são categorizados diretamente para o conjunto *R* e não fazem parte do processo de treino do modelo. Todos os outros novos exemplos são inicialmente elementos do conjunto U, que corresponde aos exemplos ainda por categorizar (*Unlearned vectors*).

Incrementação Adiabática

Ao acrescentar exemplos novos, isto é, não categorizados, para o processo de treino do modelo, o objetivo deverá ser preservar as condições KKT de todos os dados de treinos anteriores. As condições KKT são preservadas através da variação dos coeficientes α dos vetores de margem dando resposta à alguma perturbação derivada da adição de novos coeficientes. Durante o processo, os pontos podem mudar de categoria, de modo que a aprendizagem incremental prossegue através de uma sequência de etapas adiabáticas, de amplitude determinada pela contabilização dos membros da categoria como dado pelas condições KKT. Antes de uma perturbação do processo de SVM, as derivadas parciais de $\{\alpha_i, b : \forall\, i \in S\}$ são:

$$g_i = \sum_j Q_{ij}\alpha_j + y_i b - 1 = 0 \qquad \forall\, i \in S$$

$$h = \sum_j y_j \alpha_j = 0$$

Depois da perturbação causada pelos novos dados as derivadas parciais são:

$$g_i = \sum_j Q_{ij}\alpha_j + \sum_{k\in S} Q_{ik}\Delta\alpha_k + \sum_{l\in U} Q_{il}\Delta\alpha_l + y_i(b+\Delta b) - 1 = 0 \;\forall\, i \in S$$

$$h = \sum_j y_j\, \alpha_j + \sum_{k\in S} y_k \Delta\alpha_k + \sum_{l\in U} y_l \Delta\alpha_l = 0$$

Sendo $Q_{ij} = y_i y_j K\left(x_i, x_j \middle| \theta\right)$. Pode-se simplificar a incrementação de novos valores da seguinte forma, onde f(x) é a função de decisão normal e f'(x) é a função de decisão com a incrementação dos novos dados:

$$f(x) = \sum_{i\in E} y_i \alpha_i K(x_i, x|\theta) + \sum_{j\in S} y_j \alpha_j K\left(x_j, x\middle|\theta\right) + b$$

$$f'(x) = \sum_{i\in E} y_i \alpha_i K(x_i, x|\theta) + \sum_{j\in S} y_j(\alpha_j + \Delta\alpha_j) K\left(x_j, x\middle|\theta\right) + \sum_{k\in U} y_k \Delta\alpha_k K(x_k, x|\theta) + b + \Delta b$$

Para coeficientes de vetores do conjunto *U*, isto é, não categorizados, que causam perturbações pequenas, até quase inexistentes, os vetores de reserva e de erro não sofrem

alterações de categorias. Para uma dada perturbação dos coeficientes dos vetores não categorizados $\{\Delta\alpha_l: \forall\ l\ \in U\}$, o objetivo é determinar a alteração necessária nos coeficientes dos vetores de margem $\{\Delta\alpha_k: \forall\ k\ \in S\}$ e no parâmetro de ajuste Δb que permite preservar as condições de KKT em todos os dados já aprendidos (Cauwenberghs e Poggio, 2003).

Estratégia Warm Start

Shilton et al. (2005) propuseram um algoritmo diferente de SVM incremental. Este algoritmo é adaptado para situações em que os dados são obtidos sequencialmente e com variações rápidas nos parâmetros de restrições. Este método envolve a utilização de um algoritmo de "Warm-Start", isto é, uma inicialização rápida a partir de um modelo já criado. Este processo utiliza o método de "Otimização do Conjunto Ativo de Restrições" para remover pontos desnecessários e diminuir a dimensionalidade dos dados para problemas de otimização lineares com restrições. Dois conceitos são importantes para se entender esta técnica, nomeadamente:

- A aprendizagem incremental envolve voltar a treinar rapidamente um modelo de SVM após a adição de novos vetores ao conjunto de treino, anteriormente utilizado para a aprendizagem de um modelo SVM já existente.
- Da mesma forma, o problema da variação rápida das restrições envolve de uma forma rápida voltar a treinar um modelo de SVM já existente. Para tal, é utilizado o mesmo conjunto de treino mas com parâmetros de restrição diferentes.

Em ambos os casos utiliza-se o método de otimização do conjunto ativo que demonstra uma superioridade computacional do treino incremental sobre o método tradicional de SVM usado quando surgem novos dados. Nos modelos tradicionais de SVM, o treino é sempre realizado com todo o conjunto de dados de treino. Se mais dados de treino forem posteriormente adicionados, ou se for necessário testar diferentes parâmetros de restrição, o modelo SVM volta a ser treinado a partir do zero, com todos os dados do conjunto. Mas se adicionar um volume pequeno de novos dados a um conjunto de treino extenso, assume-se que provavelmente terá um efeito mínimo sobre a função de decisão. Porém, voltar a treinar um modelo do zero é um desperdício de tempo e de capacidade computacional.

A alternativa aqui proposta de inicialização rápida (Warm-Start) permite reutilizar um modelo SVM antigo na criação de um novo modelo. Neste método os valores de α_i, coeficientes dos multiplicadores de Lagrange, dos pontos que são vetores de suporte possuem valores $\alpha_i \neq 0$. Uma caraterística apelativa dos SVM é que os vetores de suporte são apenas uma pequena parte do conjunto total de treino. Isto porque, com a utilização do método do conjunto ativo, é possível reduzir a dimensionalidade do conjunto de treino que é normalmente enorme comparado com o número de vetores de suporte resultante do modelo SVM. No método do conjunto ativo, as restrições são divididas em dois conjuntos, o conjunto de restrições ativas (conjunto ativo) e o conjunto de restrições inativas (conjunto inativo). O algoritmo, em seguida e de forma iterativa percorre o conjunto de treino ajustando o conjunto de restrições ativas após cada etapa até encontrar o conjunto ativo ótimo. Em todas as iterações, cada etapa é calculada

tratando as restrições ativas como restrições de igualdade, descartando as restrições inativas, e assim resolvendo o problema de otimização irrestrita.

Outras Técnicas Incrementais do Modelo SVM

Um algoritmo de aprendizagem SVM incremental pressupõe que se possa dividir o conjunto de dados em subconjuntos, para que se possam "encaixar" em memória. Em seguida, em cada iteração do método incremental, a classificação dos dados observados é dada por um conjunto de vetores de suporte e pelo hiperplano separador das classes. Esses vetores de suporte são incorporados com o novo subconjunto de entrada de dados para fornecer os dados de treino para a próxima iteração.

O SVM incremental permite, pois, proporcionar uma representação compacta e precisa do conjunto de dados com um número pequeno de vetores de suporte, em comparação com o SVM tradicional. É razoável esperar que o modelo construído de forma incremental e os seus resultados não sejam muito diferentes do modelo construído com o conjunto completo de dados de uma só vez (modo batch). Isto deve-se ao facto de em cada iteração incremental, o SVM memoriza os vetores de suporte da iteração anterior, e esta informação contribui adequadamente para gerar o classificador na iteração seguinte. Uma vez que um novo subconjunto de dados é carregado na memória, existem diferentes possibilidades para a atualização do modelo atual. De seguida são apresentadas quatro técnicas diferentes. Para todas as técnicas, em cada iteração, apenas o subconjunto de vetores de suporte é mantido na memória (Domeniconi e Gunopulos, 2001). Algumas das técnicas propostas por Domeniconi e Gunopulos (2001) são de seguida apresentadas:

- **Técnica orientada por erro (ED).** Nesta técnica, a cada iteração do SVM, é guardada uma percentagem dos dados incorretamente classificados para utilizar no processamento de treino incremental. A técnica orientada por erro mantém assim apenas os dados classificados incorretamente. Dada a iteração SVMt com o número da iteração t, os novos dados são carregados na memória e classificados utilizando o SVMt. Se os dados são classificados incorretamente são guardados. Os dados corretamente classificados são descartados das seguintes iterações. Quando um determinado número *N* de dados mal classificados é obtido, é realizada a atualização do SVMt. Os vetores de suporte de SVMt, juntamente com os *N* pontos erroneamente classificados são usados como dados de treino para obter o novo modelo da iteração SVM$t+1$.
- **Técnica de partição fixa (PF).** Nesta técnica, o conjunto completo de dados é dividido em subconjuntos com tamanhos fixos. Quando um novo subconjunto de dados é iterado pelo SVM e carregado na memória, é adicionado ao conjunto atual de vetores de suporte. O conjunto resultante oferece o conjunto de vetores de suporte que serão usados para treinar o novo modelo. O conjunto de vetores de suporte obtidos a partir deste processo são a nova representação dos dados observados até ao momento e são mantidos em memória.
- **Técnica da excedência da margem (EM).** Dado o modelo SVMt com o número da iteração t, novos dados $\{(x_i, y_i)\}$ são carregados na memória. O algoritmo verifica se

(x_i, y_i) excede a margem definida na iteração SVM*t*. Ou seja, $y_i f_t((x_i)) \leq 1$. Se a condição for satisfeita o ponto é mantido, senão é descartado. Quando um determinado número *N* de pontos que excedem a margem são registados, a atualização do SVM*t* ocorre. Os vetores de suporte de SVM*t*, em conjunto com os *N* pontos, são utilizados como dados de treino para obter o novo modelo SVM*t+1*.

- **Técnica da excedência da margem + técnica de erros (EM + E).** Dado o modelo SVM*t* com o número da iteração *t*, novos dados $\{(x_i, y_i)\}$ são carregados na memória. O algoritmo verifica se (x_i, y_i) excede a margem definida na iteração SVM*t*. Ou seja, $y_i f_t((x_i)) \leq 1$. Se a condição for satisfeita o ponto é mantido, senão ele é classificado usando o SVM*t*. Se for classificado incorretamente ele é mantido, senão é descartado. Quando um determinado número *N* de dados, seja excedendo a margem ou classificados incorretamente, é obtido, a atualização de SVM*t* ocorre. Os vetores de suporte de SVM*t*, juntamente com os *N* pontos, são usados como dados de treino para obter o novo modelo da iteração SVM*t+1*.

5.2. Definição de um critério de atualização

Até ao momento, foram introduzidos alguns métodos de modelação SVM incremental. Agora, é necessário definir um critério para saber quando se deve atualizar um modelo. Algumas das técnicas referidas anteriormente somam o número N de exemplos mal classificados, e atingido um determinado número N, o modelo é atualizado. Outros contabilizam os pontos que excedem a margem, e da mesma forma, quando um determinado número de pontos é atingido, realiza-se a atualização do modelo.

No contexto das ETAR a melhor forma para se determinar o melhor momento para se atualizar um modelo é quando um determinado número de exemplos é mal classificado. Realizar a previsão do valor de um constituinte dos efluentes faz sentido se o valor de previsão for obtido em tempo real. Sendo assim, o conjunto de teste é constituído por um ou por relativamente poucos exemplos. Neste caso em concreto seria contabilizado o número de exemplos mal classificados e quando N exemplos forem mal classificados, o modelo é incrementalmente atualizado. Outra forma não tão viável para uma organização como as ETAR seria a utilização do valor da acuidade. Uma previsão obtém sempre um certo grau de assertividade chamado Acuidade. A acuidade é um conceito apresentado a seguir e representa a percentagem (%) de exemplos bem classificados. Quando um determinado limite mínimo de acuidade é atingido, o modelo é atualizado. Este método necessita de um conjunto de teste com algum volume pois, como é óbvio, não é possível determinar a acuidade de um modelo apenas com um exemplo de previsão.

5.3. Avaliação dos Modelos

Ao longo do documento, por várias vezes, se referiram os conceitos de conjunto de treino e conjunto de teste. Dado que os modelos de classificação são desenvolvidos para serem aplicados a dados diferentes daqueles que foram usados para os construir, torna-se essencial

avaliá-los para medir até que ponto eles são capazes de fazerem as previsões corretamente. Essa avaliação é tipicamente feita separando o conjunto de dados disponíveis em 2 partes: o conjunto de treino e o conjunto de teste. O modelo é construído com base no conjunto de treino. Depois é aplicado aos dados de teste e compara-se o valor da classe de cada exemplo neste conjunto com o que se obtém na previsão. Esta técnica serve para avaliar o modelo e estimar a incerteza das suas previsões. Existem várias medidas de avaliação dos modelos que permitem saber a consistência de cada modelo e avaliar o seu desempenho na classificação dos dados. De seguida apresentam-se os conceitos das metodologias e algumas medidas de avaliação de desempenho de um modelo de *Data Mining*.

5.3.1. Metodologias de Avaliação do Desempenho

As dimensões de análise que as metodologias de avaliação do desempenho de um modelo de *Data Mining* pretendem avaliar são a taxa de erro, o tempo de aprendizagem, ou a complexidade do modelo. A metodologia mais exata e precisa é das taxas de erros que calcula, entre outros erros, o desvio entre os valores previstos e os valores reais. A análise das taxas de erros inclui a utilização de um ou vários conjuntos de testes. Os métodos para a criação dos conjuntos de teste são semelhantes na medida que retiram exemplos do conjunto de treino para serem utilizados como conjunto de teste. Este processo de separação do conjunto total de dados em conjuntos de treino e conjuntos de teste é realizado de diversas formas apresentadas a seguir (Han, Pei e Kamber, 2006).

Validação Cruzada

O método de Validação Cruzada (*Cross Validation*) é uma forma amplamente aceite para dividir um conjunto total de dados em vários subconjuntos de teste estatisticamente independentes. Cada subconjunto é utilizado isoladamente como conjunto de teste enquanto os outros subconjuntos formam o conjunto de treino. Isso permite a construção de intervalos de confiança para uma determinada medida de desempenho definida como critério de avaliação.

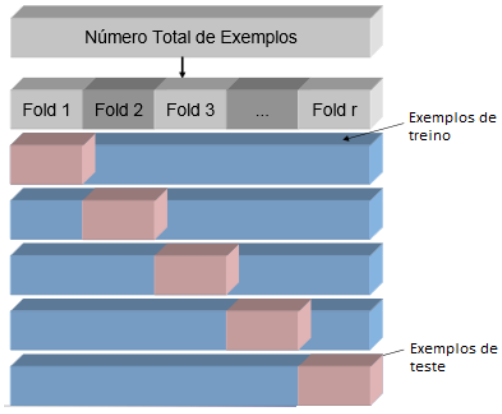

Figura 16 – Representação da divisão do conjunto total de dados através do método k-fold Cross Validation – figura extraída e adaptada de (Baranauskas, 2006).

O método *k-fold Cross Validation* divide o conjunto de dados total em *K* subconjuntos, *K > 1.* O modelo é treinado com todos os subconjuntos exceto um que será o conjunto de teste. Este

procedimento é repetido em *K* tentativas, utilizando em cada iteração um subconjunto de teste diferente. O desempenho do modelo é verificado realizando a média do erro quadrático de cada iteração do processo. Uma desvantagem deste processo é que o modelo tem que ser treinado *K* vezes o que representa um custo computacional elevado. Quando o conjunto total de dados é relativamente pequeno pode ser utilizado uma variante da validação cruzada chamada *Leave-one-out*, disponível no RapidMiner. Este método divide os n exemplos em n subconjuntos e testa cada exemplo de cada vez. Neste caso, cada exemplo de teste é previsto individualmente.

O *Stratified Cross Validation* é outra vertente de validação cruzada que tem em consideração a distribuição das classes no conjunto total de dados. Por exemplo, se a percentagem de distribuição dos dados pelas classes é classe +1 = 30% e classe -1 = 70%, então os subconjuntos também terão aproximadamente a mesma percentagem de distribuição pelas classes.

Holdout

O método *Holdout* divide o conjunto total de dados em dois subconjuntos, conjunto de treino e conjunto de teste. A divisão dos dados atribui 2/3 do total dos dados ao conjunto de treino e 1/3 ao conjunto de teste. A divisão pode ser feita como ilustrada na figura seguinte ou pode ser realizada sob a forma de partições aleatórias, mas com as mesmas proporções, do conjunto total de dados.

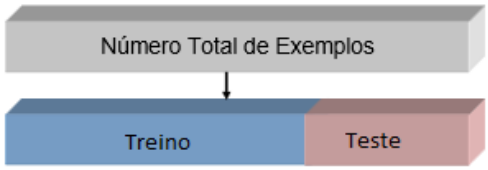

Figura 17 - Representação da divisão do conjunto total de dados através do método *Holdout* – figura extraída e adaptado de (Baranauskas, 2006).

Bootstrap

Se a validação cruzada divide os dados em vários subconjuntos sem que o mesmo exemplo possa estar duplicado no mesmo subconjunto de treino, o método *Bootstrap* permite a repetição de dados nos seus subconjuntos. Um conjunto de dados com n exemplos é utilizado n vezes com o objetivo de formar um novo conjunto com n exemplos, entre eles, exemplos duplicados. Este novo conjunto é utilizado como conjunto de treino. Os exemplos que não entram no conjunto de treino são utilizados no conjunto de teste. Tal como a validação cruzada, Este método itera *k* vezes. A taxa de erro é calculada com base na média das avaliações das *k* iterações.

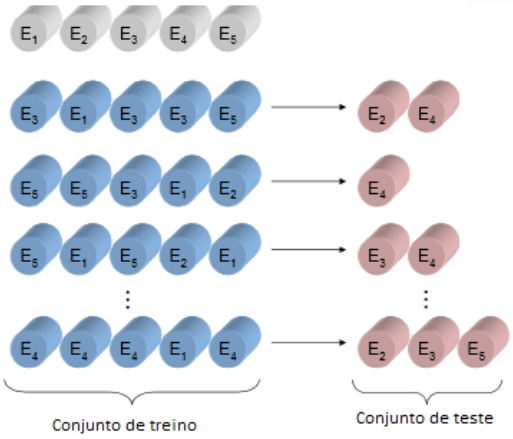

Figura 18 - Representação da divisão do conjunto total de dados através do método *Bootstrap* - figura extraída e adaptada de (Baranauskas, 2006).

5.3.2. Medidas de Avaliação de Modelos

Como se acabou de constatar, para o mesmo objetivo de previsão podem ser utilizadas diversas técnicas de aprendizagem para vários tipos de modelos. Torna-se necessário, assim, avaliar os modelos criados para se conhecer o modelo mais adequado para cada caso em concreto. Existem várias medidas de desempenho que permitem avaliar um modelo. Algumas destas medidas são no entanto restritas a alguns modelos, como por exemplo a Acuidade, a matriz de confusão, e a margem que só podem ser obtidas a partir de modelos de classificação. Isto porque, por exemplo, avaliar a distância da margem de separação entre duas classes, como é óbvio, apenas é possível em casos em que se pretende prever a classe de um registo. Nos casos de regressão pretende-se obter valores de previsão numéricos e contínuos, os dados não são divididos em duas classes.

Medidas de Desempenho para Classificação

Matrizes de Confusão

Uma matriz de confusão é uma tabela usada em classificação e apresenta os resultados das previsões realizadas. As linhas da tabela representam os valores previstos de uma classe enquanto as colunas representam os valores reais de uma classe. Uma das vantagens da matriz de confusão é que é de fácil análise. Desta forma, podemos considerar que a matriz de confusão é uma tabela com duas linhas e duas colunas que apresenta o número de Verdadeiros Negativos (VN), Falsos Positivos (FP), Falsos Negativos (FN) e Verdadeiros Positivos (VP).

Valor Previsto (predito pelo teste)		Valor Verdadeiro (confirmado por análise)	
		positivos	negativos
	positivos	**VP** Verdadeiro Positivo	**FP** Falso Positivo
	negativos	**FN** Falso Negativo	**VN** Verdadeiro Negativo

Figura 19 - Modelo de Matriz de Confusão

Acuidade

É a precisão do sistema de previsão considerando a quantidade de valores corretamente classificados no total dos registos. Esta medida é calculada através da fórmula:

$$\text{Acuidade} = \frac{VP + VN}{VP + VN + FP + FN}$$

Margem de Separação de Duas Classes

Outra medida de desempenho utilizada pelo RapidMiner para modelos de classificação é a margem de separação de duas classes. Como foi apresentado no capítulo anterior, os modelos de classificação separam os dados em duas classes distintas. A distância entre os pontos mais próximos das duas classes é a chamada margem de separação. Ora quanto maior a margem de separação (maximização da margem) mais assertivo é o modelo na previsão da classe de cada exemplo. Comparar as margens dos modelos é uma possível medida do desempenho da aprendizagem. O modelo com maior margem de separação é teoricamente o mais assertivo nas suas previsões.

Medidas de Desempenho para Regressão.

Quando se utiliza um modelo é necessário comparar o resultado obtido com o valor real. Isso possibilita medir a assertividade de previsão aproximada ao quantificar a diferença com o valor estimado. Estas medidas de desempenho podem ser obtidas através de escalas contínuas e numéricas, e retratam a aproximação entre os valores previstos e os valores reais. Os modelos de regressão permitem selecionar o modelo que realiza a previsão de valores mais próximos dos dados reais. A diferença entre o valor real *y* e o valor previsto $\hat{y}$ é o denominado erro ou desvio. Algumas das medidas de desvio mais utilizadas são o Desvio médio absoluto (*MAD - Mean Absolute Deviation*); Soma do Erro Quadrático (*SSE – Sum Squared Error*); Erro Quadrático Médio (*MSE - Mean Squared Error*) e a Raiz Quadrada do Erro médio (*RMSE - Root Mean Squared Error*).

$$MAD = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n}$$

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$$MSE = \frac{SSE}{n}$$

$$RMSE = \sqrt{MSE}$$

Destas medidas, a que foi utilizada, nomeadamente para comparar os modelos durante o processo de seleção de atributos foi a raiz quadrática do erro médio (*RMSE – Root Mean Squared Error*). Esta medida consiste no cálculo da média do somatório das diferenças entre os valores reais e os valores previstos. Em que y representa os valores reais, $\hat{y}$ representa os valores previstos e n representa o número de exemplos do conjunto testado. Outra medida utilizada no mesmo processo de seleção de atributos foi a comparação dos coeficientes de correlação entre os atributos de cada modelo. O Coeficiente de correlação de Pearson é calculado a partir da seguinte fórmula:

$$R = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{(\sum (x_i - \bar{x})^2)(\sum (y_i - \bar{y})^2)}}$$

em que x_i e y_i são os valores das variáveis X e Y. E $\bar{x}$ e $\bar{y}$ são as respetivamente as médias de x_i e y_i.

Capitulo 6
Análise e Preparação dos Dados

6.1. Conceitos importantes para a preparação dos dados

Recolha dos dados

A recolha dos dados numa ETAR é realizada ao longo do processo de tratamento, nos chamados pontos de amostragem através de sensores que realizam a medição dos valores físicos, químicos e biológicos dos efluentes. Os pontos de amostragem (SP) encontram-se entre cada etapa do tratamento. O processo de tratamento em algumas ETAR é composto por duas linhas de tratamento da fase liquida. Porém com a finalidade de simplificar o conjunto de dados, grande parte dos estudos relacionados com este processo de tratamento reduz o conjunto de dados a uma só linha de tratamento. Devido a realização de diversas leituras diárias, o volume de dados aumenta muito rapidamente. Alguns estudos sugerem para controlar este enorme fluxo de entrada de dados que cada registo represente a média dos valores obtidos para cada variável ao longo de um dia.

Analise dos dados

O processo de análise dos dados pressupõe o estudo de todas as caraterísticas apresentadas pelo conjunto de dados com a finalidade de se dar ordem, estrutura e significado aos dados. Conhecer as caraterísticas das variáveis é essencial para proceder a preparação dos dados. Cada modelo de *Data Mining* necessita de determinadas condições sobre os dados para que possa ser executado. As caraterísticas dos dados são definidas segundo vários conceitos (Cortez, 2013):

- **Tipo de dados -** Cada atributo que constitui o conjunto de dados possui um tipo de dados. Os dados podem ser qualitativos, isto é não numéricos ou quantitativos caso sejam valores numéricos. Os atributos qualitativos são definidos por várias categorias, ou seja, representam uma classificação dos indivíduos. Podem ser nominais ou ordinais.

    - **Variáveis nominais**, em que não existe ordenação dentre as categorias. Exemplos: sexo, cor dos olhos.

- **Variáveis ordinais**, em que existe uma ordenação entre as categorias. Exemplos: temperatura (baixa, média, alta), mês do ano (janeiro, fevereiro, dezembro).

Os atributos quantitativos são as caraterísticas que podem ser medidas numa escala quantitativa, ou seja, apresentam valores numéricos que fazem sentido. Podem ser contínuas ou discretas.

- **Variáveis contínuas** - São caraterísticas mensuráveis que assumem valores numa escala contínua (na reta real), incluindo valores decimais. Normalmente são medidas através de algum instrumento. Exemplos: peso, altura, tempo, pressão arterial, idade.
- **Variáveis discretas -** São caraterísticas mensuráveis que podem assumir apenas um número finito ou infinito contável de valores inteiros. Geralmente são o resultado de contagens. Exemplos: número de filhos, número de bactérias por litro de água.

- **Valores Nulos -** Um valor nulo é um valor ausente no conjunto de dados mas existente no contexto em que a medição foi realizada. Uma quantidade significativa de valores nulos pode tornar um conjunto de dados inutilizável.
- **Outliers -** A ocorrência de valores irregulares, designados de *outliers*, devido a erros na recolha dos dados é comum, por exemplo, em sistemas em que a recolha de dados é feita através da inserção por parte de um utilizador ou em casos de avarias nos instrumentos de medição. Como não se enquadram no domínio dos valores relacionados com o fenómeno que se pretende estudar, prejudicam a aprendizagem, devido ao aumento de complexidade.
- **Correlação entre atributos -** A correlação entre atributos visa estudar a interdependência dos dados. Esta correlação é importante pois permite uma melhor compreensão dos dados descobrindo algumas tendências nos dados. Os valores da correlação quadrada R^2 vão de 0 a 1, onde 0 aponta para a inexistência de correlação linear, enquanto que 1 indica que os atributos são extremamente colineares. Um exemplo no caso em estudo é facilmente observável analisando os atributos CBO e SSV. Sempre que o valor de SSV é alto, o valor de CBO também é alto. Isto deve-se ao facto das bactérias que decompõem os sólidos suspensos serem aeróbios e quanto maior a carência bioquímica de oxigênio (CBO), maior a concentração de sólidos suspensos voláteis (SSV).

Preparação de Dados

O processo que advém a seguir à análise dos dados é a preparação dos dados. Depois de compreender os dados e o seu formato, é necessário transformá-los de forma a serem interpretáveis pelos modelos de *Data Mining* que se pretende utilizar. Cada algoritmo de modelação pode possuir restrições específicas sobre as caraterísticas dos dados, nomeadamente:

Tratamento de valores nulos - Existem duas formas de tratar valores nulos. A primeira é eliminar os registos com valores nulos, o que pode tornar o conjunto de dados bastante reduzido, ao ponto de não ser suficiente para se obter bons resultados. A outra opção é a substituição dos valores nulos por um determinado valor. O valor de substituição pode ser um valor dado pelo utilizador, o valor médio do atributo calculado a partir de todos os registos com valores corretos, valores extraídos de outros conjuntos de dados, o valor do registo mais semelhante/próximo, um valor estimado por regressão linear, ou uma combinação dos métodos anteriores (Cortez, 2013).

Remoção de Outliers - Os outliers são facilmente identificados durante a análise do conjunto de dados ou utilizando ferramentas de visualização que permitam ter uma ideia da distribuição dos atributos dos dados no seu domínio. Os outliers encontram-se normalmente em pequenas quantidades o que permite a remoção dos exemplos considerados outliers sem reduzir em demasia o conjunto de dados.

Normalização - Determinados algoritmos de *Data Mining* necessitam que o conjunto de dados esteja normalizado. Porém esse não é o caso das SVM. A normalização dos dados possibilita a alteração do domínio de valores dos atributos do conjunto de dados. Todas as dimensões do espaço de dados ficam assim à mesma escala, dentro dos limites do intervalo de valores.

A normalização dos dados permite uma melhor interpretação do conjunto de dados pelos algoritmos de modelação. Desta forma, os algoritmos poderão alcançar melhores níveis de eficácia com uma maior facilidade de aprendizagem (Pyle, 1999). Por exemplo, dados dois atributos, temperatura e PH, os valores de cada atributo variam em escalas diferentes, 150°C, 60°C, para a temperatura e 8.2, 7.5 para o PH. A normalização altera os valores de ambos os atributos para a mesma escala ficando no mesmo intervalo de valores.

Discretização - Como já foi referido, alguns algoritmos usados para criar modelos de *Data Mining* precisam de tipos de conteúdo específicos para que funcionem corretamente. Assim, alguns métodos de aprendizagem apenas lidam com valores numéricos discretos. Como tal, torna-se necessário converter os valores nominais ou numéricos contínuos em valores numéricos discretos, sob a forma de intervalos ordenados categorizados ou de categorias de valores. Como por exemplo:

Cor: azul -> 1, vermelho -> 2, verde -> 3;

Altura: [1,50m-1,60m] -> 1, [1,61m-1,75m]->2, [1,76m-1,90m]->3.

Seleção de atributos - O propósito da seleção de atributos é resolver o problema da alta dimensionalidade. O processo envolve selecionar um subconjunto de atributos de um conjunto total de dados, e isso permite reduzir a dimensionalidade dos dados, eliminando atributos irrelevantes. Este processo melhora a performance dos algoritmos de aprendizagem e leva a uma representação mais facilmente interpretável do problema. Existem várias técnicas de seleção de atributos. Dos quais os métodos de seleção *Wrappers*, *Filters*, e *Embebbed* (Pyle, 1999).

Os métodos *Wrappers* utilizam os algoritmos de aprendizagem e o seu desempenho preditivo como meio de avaliar o subconjunto de atributos utilizado na aprendizagem.

Neste método, são testados vários subconjuntos de atributos, o subconjunto que conduza a um melhor desempenho preditivo do modelo é o escolhido. Nos métodos *Wrappers* é necessário definir três componentes: a componente de procura e seleção de atributos, que cria os subconjuntos iterativamente; a componente de avaliação do modelo, que avalia o desempenho do modelo criado com cada subconjunto; e o modelo de previsão que será utilizado na modelação. A primeira componente de procura de atributos pode ser de dois tipos:

- **Procura para a frente (Forward),** em que a procura é iniciada sem atributos no subconjunto e os mesmos são adicionados um a um isoladamente e o conjunto resultante é avaliado segundo um determinado critério. O atributo que produz o melhor resultado é incorporado.
- **A procura para trás (Backward),** na qual a procura é iniciada com o conjunto total de atributos. A cada passo, os atributos menos relevantes são eliminados do subconjunto um a um até que apenas fique os atributos que melhorem o desempenho do modelo de previsão.

As componentes de avaliação do modelo, e a componente do algoritmo de modelação são definidas pelo utilizador. Os métodos *Wrappers*, pelo facto de criar e testar iterativamente vários subconjuntos, têm um alto custo computacional. Nos métodos *Filters*, ao contrário dos métodos *Wrappers*, o processo de seleção dos atributos é independente do algoritmo de modelação. A seleção dos atributos acontece antes do processo de aprendizagem. Os métodos *Filters* utilizam medidas específicas para avaliar a qualidade dos atributos disponíveis. Esses métodos podem avaliar cada atributo independente dos outros, determinando o grau de correlação entre cada atributo e a classe (Yang e Pedersen, 1997), ou podem avaliar subconjuntos de atributos, buscando através de estratégias e heurísticas, aqueles que, em conjunto, melhor predizem as classes.

Os métodos Embebbed são métodos incorporados no próprio algoritmo de previsão. Estes métodos selecionam o conjunto de atributos no próprio processo de construção do modelo de previsão, durante a fase de treino, e são geralmente específicos para um dado algoritmo de classificação. Existem ainda outros métodos não supervisionados de seleção de atributos. Um exemplo é o método baseado na correlação entre os atributos (CFS) que é utilizado neste trabalho. Neste método, são incorporados os atributos que maior correlação tem com o atributo alvo (*label*). Como já foi referido, O índice de correlação varia entre 0 e 1. Em alguns casos o utilizador pode escolher o valor mínimo da correlação dos atributos a serem incorporados no subconjunto.

6.2. A Variável de Interesse

Um dos objetivos deste trabalho é a previsão dos valores das variáveis que medem a qualidade dos efluentes tratados numa ETAR. Conhecer as caraterísticas das águas residuais no final do tratamento é primordial, caso contrário correr-se-ia um risco enorme ao libertar essas massas hídricas para o meio ambiente. Como tal, pretende-se neste trabalho melhorar este

processo de medição da qualidade do efluente após a etapa final do tratamento. Dos parâmetros avaliados para a qualidade das águas destacam-se a Carência Química de Oxigênio (CQO), a Carência Bioquímica de Oxigénio (CBO) e os Sólidos Suspensos (SS), visto que são parâmetros com valores máximos recomendados e regulados por lei (Decreto-Lei nº 152/97 de 19 de Junho, 1997).

O parâmetro alvo de previsão neste trabalho e presente em ambos os conjuntos de dados utilizados é o CBO. Esta variável será a variável de interesse, devido ao prazo demorado de 5 dias após a recolha para se obter, em laboratório, os seus valores. Como tal, esta parece ser a variável ideal para ser alvo deste estudo. A CBO é também uma variável com uma correlação muito elevada com várias outras variáveis, tais como o PH e o SS. Este facto compreende-se devido a necessidade de oxigénio no tratamento aeróbio que decompõe as matérias orgânicas e sólidas em suspensão.

6.3. Ferramentas Utilizadas na Preparação de Dados

A ferramenta utilizada na preparação dos dados foi o RapidMiner, desenvolvido pela empresa Rapid-i. Esta ferramenta, desenvolvida em Java, é um ambiente *open-source* que permite explorar os dados de diversas maneiras por meio de operadores interligados (sejam eles para pré-processamento, aprendizagem, validação, etc.). O principal motivo para a utilização desta ferramenta neste trabalho foi o grau de familiarização com a ferramenta que foi obtido durante o percurso académico.

O RapidMiner torna-se muito fácil de utilizar após alguma prática com a própria ferramenta. Para além disso, dispõe de um amplo leque de conceitos e instruções de como utilizar cada operador ou algoritmo de aprendizagem. Possui ainda uma interface gráfica bastante intuitiva, que inclui uma componente de correção que sugere alterações nos esquemas dos operadores quando algum erro é apresentado. Talvez uma das poucas limitações do RapidMiner encontra-se na variedade das tarefas. Apesar de ser um ambiente *open-source* não possui todas as tarefas necessárias por exemplo para a realização desta dissertação. Deste modo, não possui algoritmos SVM incrementais, nem permite a atualização de um modelo SVM já criado. No entanto, Os conhecimentos adquiridos anteriormente sobre a sua utilização e a possibilidade de integrar tarefas de outras ferramentas no RapidMiner fizeram com que este fosse o eleito para a realização desta etapa do projeto.

6.4. O 1º Conjunto de dados

Caraterização dos dados

Este conjunto de dados, composto por 34 atributos, possuía inicialmente 53 registos relativos a atividade de uma ETAR durante um período de 7 meses. Cada registo corresponde a um dia de tratamentos. Os valores de um registo resultam da média das várias medições que são diariamente realizadas para cada parâmetro. Como já foi referido, algumas ETAR procedem ao

tratamento paralelo em duas linhas de tratamento. No entanto, vários estudos simplificam o conjunto de dados obtido através das medições numa só linha de tratamento. Isto permite reduzir a dimensionalidade dos dados porém em casos com muitos dados podem distorcer os dados. O conjunto de dados é maioritariamente composto por atributos relativos as medições realizadas em cada ponto de amostragem (*SP*). Este conjunto de dados é composto por medições realizadas em 6 pontos de amostragem (*SP*), em que *SP1* corresponde ao afluente que dá entrada na ETAR. O tratamento é realizado em três tanques principais, o Decantador primário, o tanque de arejamento e o decantador secundário, na saída de cada um dos 3 tanques existe um ponto de amostragem onde é realizada medições de vários parâmetros. No fim de cada um destes tanques são realizadas as amostragens de *SP2*, *SP3* e *SP5* respetivamente, sendo o *SP4* uma amostragem de recirculação das águas que necessitam de voltar a ser tratadas e voltam ao início do tratamento. O ponto de amostragem *SP6* realiza medições da qualidade das águas residuais já tratadas e prontas para serem libertadas. No entanto, existem também 2 atributos relativos a Data e a estação do ano. Visto que cada registo corresponde a um dia de medições, e que nenhum dia se repete, a data é um atributo importante pois representa o identificador único de cada registo.

Atributo	Descrição	Escala de medição
Season	Estação do ano	*Não aplicável*
Date	Data	*Não aplicável*
SP1_Redox	Potencial de redução do afluente à entrada da ETAR.	*(mV)*
SP1_pH	Indica a acidez, neutralidade ou alcalinidade do afluente à entrada da ETAR.	*Não aplicável*
SP1_CQO	Indica a Carência Química de oxigénio do afluente à entrada da ETAR.	*(mg/l)*
SP1_CBO	Indica a Carência Bioquímica de oxigénio do afluente à entrada da ETAR.	*(mg/l)*
SP1_TSS	Indica o valor Total de Sólidos Suspensos no afluente à entrada da ETAR.	*(mg/l)*
SP1_VSS	Indica a quantidade de Sólidos Suspensos Voláteis do afluente à entrada da ETAR.	*(mg/l)*
SP2_pH	Indica a acidez, neutralidade ou alcalinidade das águas residuais à saída do decantador primário.	*Não aplicável*
SP2_DO2(AZ)	Indica a quantidade de oxigénio dissolvido, em zona aeróbia, das águas residuais à saída do decantador primário.	*(mg/l)*
SP2_DO2(AxZ)	Indica a quantidade de oxigénio dissolvido, em zona Anóxica, das águas residuais à saída do decantador primário.	*(mg/l)*
SP2_Redox(AZ)	Potencial de redução em zona aeróbia à saída do decantador primário.	*(mV)*
SP2_TSS	Indica o valor Total de Sólidos Suspensos das águas residuais à saída do decantador primário.	*(mg/l)*
SP2_VSS	Indica a quantidade de Sólidos Suspensos Voláteis das águas residuais à saída do decantador primário.	*(mg/l)*
SP3_pH	Indica a acidez, neutralidade ou alcalinidade das águas residuais à saída do tanque de arejamento.	*Não aplicável*
SP3_DO2	Indica a quantidade de oxigénio dissolvido das águas residuais à saída do tanque de arejamento.	*(mg/l)*
SP3_Redox	Potencial de redução das águas residuais à saída do tanque de	*(mV)*

	arejamento.	
SP3_V30	Indica o volume do lodo sedimentado das águas residuais à saída do tanque de arejamento após 30 minutos.	*(mg/l)*
SP3_TSS	Indica o valor Total de Sólidos Suspensos das águas residuais à saída do tanque de arejamento.	*(mg/l)*
SP3_VSS	Indica a quantidade de Sólidos Suspensos Voláteis das águas residuais à saída do tanque de arejamento.	*(mg/l)*
SP4_Redox	Indica o Potencial de redução das águas residuais que voltam para o início do tratamento.	*(mV)*
SP4_TSS	Indica o valor Total de Sólidos Suspensos das águas residuais que são sujeitas a recirculação.	*(mg/l)*
SP4_VSS	Indica a quantidade de Sólidos Suspensos Voláteis das águas residuais que voltam para o início do tratamento.	*(mg/l)*
SP4_VSS/TSS%	Indica o rácio entre os sólidos suspensos voláteis e o total de sólidos suspensos das águas residuais que voltam para o início do tratamento.	*%*
SP5_pH	Indica a acidez, neutralidade ou alcalinidade das águas residuais à saída do decantador secundário.	*Não aplicável*
SP5_CQO	Indica a Carência Química de oxigénio das águas residuais à saída do decantador secundário.	*(mg/l)*
SP5_CBO	Indica a Carência Bioquímica de oxigénio das águas residuais à saída do decantador secundário.	*(mg/l)*
SP5_TSS	Indica o valor Total de Sólidos Suspensos das águas residuais à saída do decantador Secundário.	*(mg/l)*
SP5_VSS	Indica a quantidade de Sólidos Suspensos Voláteis das águas residuais à saída do decantador Secundário.	*(mg/l)*
SP6_pH	Indica a acidez, neutralidade ou alcalinidade das águas residuais no final do tratamento.	*Não aplicável*
SP6_CQO	Indica a Carência Química de oxigénio das águas residuais no final do tratamento.	*(mg/l)*
SP6_CBO	Indica a Carência Bioquímica de oxigénio das águas residuais no final do tratamento.	*(mg/l)*
SP6_TSS	Indica o valor Total de Sólidos Suspensos das águas residuais no final do tratamento.	*(mg/l)*
SP6_VSS	Indica a quantidade de Sólidos Suspensos Voláteis das águas residuais no final do tratamento.	*(mg/l)*

Tabela 5 - Variáveis do conjunto de dados - primeiro conjunto.

Atributo	Pontos de amostragem
Season	Externo
Date	Externo
Redox	SP1, SP2(AZ), SP3, SP4
PH	SP1, SP2, SP3, SP5, SP6
CQO	SP1, SP5, SP6
CBO	SP1, SP5, SP6
TSS	SP1, SP2, SP3, SP4, SP5, SP6
VSS	SP1, SP2, SP3, SP4, SP5, SP6
DO2	SP2 (AZ, AxZ), SP3
V30	SP3
VSS/TSS%	SP4

Tabela 6 - Pontos de amostragem em que são realizadas as medições de cada parâmetro.

O Enriquecimento dos Dados

O conjunto de dados inicial, composto por 53 registos, possui um volume de dados relativamente pequeno. Tal quantidade de dados não é suficiente para se obter uma aprendizagem de um modelo com um desempenho satisfatório. Deste modo, tornou-se necessário gerar dados a partir dos 53 exemplos. Para isso, tentou-se identificar algumas tendências nos dados para poder passá-las para os novos registos. Um exemplo de tendência identificada era que nas estações do ano mais quentes (Verão e Primavera), os valores de alguns atributos como o PH e o CBO eram altos. Aliás, todas as tendências inicialmente identificadas utilizando apenas a observação do conjunto de dados são comparações relativas ao atributo estação do ano com todos os outros atributos. Isto permitiu identificar os atributos que supostamente são afetados pelas alterações de temperatura. Porém, outras tendências e relações entre atributos que não foram identificadas Estas relações entre atributos podem-se ter perdidas ao gerar os novos registos. Este facto afeta diretamente a correlação dos atributos. Todos os atributos foram gerados aleatoriamente com a utilização da função "*AleatórioEntre*" do *EXCEL* segundo um limite inferior e um limite superior definido para cada atributo. Após a geração aleatória de novos dados através do *EXCEL*, o conjunto de dados é formado por 5995 registos, referentes aos anos 1995 até 2014.

Tratamento de Valores nulos e valores duplicados

Para tratar os chamados valores nulos o RapidMiner permite duas opções, possibilita a remoção dos registos com valores nulos, e permite ainda substituir os valores omissos por valores especificados. Estes valores especificados podem ser a média dos valores do mesmo atributo, o máximo ou o mínimo, ou simplesmente colocar o valor zero. Neste trabalho recorreu-se ao operador de substituição de valores nulos pela média de cada atributo. A utilização deste operador de substituição de valores omissos poderia ser prejudicial, e interferir nos resultados obtidos posteriormente caso existisse um elevado número de valores nulos. Porém, sendo que grande parte dos dados foram gerados aleatoriamente, a percentagem de valores nulos para cada atributo era pouco significativa. Os valores nulos de cada atributo foram substituídos por valores obtidos calculando a média dos valores de cada atributo. O processo de geração de datas preencheu várias instâncias do atributo "Date" com as mesmas datas. Sendo que cada registo corresponde a média diária das leituras dos sensores em cada ponto de amostragem, cada data nunca pode figurar mais do que uma vez no conjunto total dos dados. Para tratar esta anomalia utilizou-se o operador de remoção de duplicados disponível no RapidMiner. Os parâmetros utilizados para tal foram a seleção de um só atributo (*single*) no parâmetro de filtragem dos tipos dos atributos. No parâmetro atributo, selecionou-se o atributo que se pretendia (Date). Este processo removeu todos os registos com datas duplicadas.

Outliers

Como seria de esperar o conjunto de dados encontra-se livre de *outliers*. Isso deve-se ao facto dos dados terem sido gerados com limites inferiores e superiores definidos para cada atributo.

Correlação

O estudo da interdependência entre dois atributos é muito importante, nomeadamente para conhecer como os atributos se relacionam e para saber quais as variáveis que afetam outras variáveis. O RapidMiner permite criar uma matriz de correlação entre os atributos do conjunto de dados. Esta matriz permite visualizar os relacionamentos colineares entre cada par de atributos e classificar a correlação segundo uma escala de 0 a 1, na qual 0 representa a ausência de correlação e 1 significa uma correlação extrema. Neste caso em concreto, apenas se consideram relações com coeficiente superior ou igual a 0,1. A tabela seguinte mostra as relações colineares mais fortes do conjunto de dados. O RapidMiner permite normalizar os pesos de cada relação colinear, porém, é preferível obter os valores reais para o peso de cada correlação.

First Attribute	Second Attribute	Correlation
SP2_TSS	SP3_TSS	0.902
SP3_TSS	SP4_TSS	0.818
SP4_TSS	SP4_VSS	0.765
SP2_TSS	SP4_TSS	0.737
SP4_VSS	SP4_VSS/TSS%	0.687
SP1_Redox	SP2_Redox(AZ)	0.644
SP3_TSS	SP4_VSS	0.621
SP2_TSS	SP4_VSS	0.561
Season	SP3_V30	0.456
Season	SP5_BOD	0.280
SP3_V30	SP5_BOD	0.275
SP4_TSS	SP4_VSS/TSS%	0.227
Season	SP1_pH	0.205
SP1_Redox	SP3_V30	0.198
SP3_TSS	SP4_VSS/TSS%	0.179
Season	SP1_Redox	0.171
SP2_TSS	SP4_VSS/TSS%	0.165
SP1_Redox	SP5_BOD	0.139
SP2_Redox(AZ)	SP3_V30	0.133
SP2_Redox(AZ)	SP3_Redox	0.131
Season	SP5_pH	0.126
SP2_pH	SP3_pH	0.124
SP1_pH	SP2_pH	0.121
Season	SP2_Redox(AZ)	0.115
Season	SP4_Redox	0.108
SP1_pH	SP3_V30	0.104

Tabela 7 - Representação dos pares de atributos com maiores coeficientes de correlação - primeiro Conjunto.

Como se pode verificar pela tabela anterior, os atributos com maior correlação são atributos pertencentes ao mesmo parâmetro *TSS,* porém de pontos de amostragem diferentes. Esta correlação é evidente pois, se por alguma anomalia a quantidade de Sólidos Suspensos (SS) for muito elevada, ou pelo contrário baixa, à entrada da ETAR, os valores de SS serão altos, ou baixos ao longo de vários pontos de amostragem. Isso deve-se a necessidade da realização de várias etapas do tratamento para obter uma remoção completa de todos os Sólidos Suspensos. A relação colinear mais forte entre dois parâmetros diferentes possui um coeficiente de 0,765. Os parâmetros são o Total de Sólidos Suspensos (TSS) e a quantidade de Sólidos Suspensos Voláteis (VSS) no ponto de amostragem *SP4.* Relativamente à variável de interesse (CBO - em

inglês BOD), possui um coeficiente de 0,280 de relacionamento colinear com a Estação do ano (Season) e de 0,275 com o Iodo sedimentado após 30 minutos (V30).

O relacionamento entre os atributos foi afetado pelo facto dos registos de cada atributo terem sido gerados isoladamente dos outros atributos. Se por exemplo, o parâmetro CQO possuísse uma forte corelação com CBO, e numa determinada data os valores de CQO fossem elevados, os valores de CBO também seriam afetados. No entanto, como os atributos foram gerados isoladamente, os valores de um determinado atributo não apresenta a real corelação com outros atributos.

Normalização

As técnicas de normalização de dados que se destacam no RapidMiner foram o *MinMax-1:1* e o *MinMax0:1*. Estas técnicas são relativamente idênticas. Ambas normalizam os atributos para intervalos de valores. No entanto a diferença entre estas duas técnicas encontra-se no próprio intervalo de valores. Enquanto o *MinMax-1:1* normaliza os dados para o intervalo de valores $[-1; 1]$, o *MinMax0:1* normaliza os dados para valores entre $[0; 1]$. Os testes realizados no RapidMiner para verificar os benefícios da normalização dos dados demonstraram uma melhoria pouco significativa dos dados normalizados sobre os dados não normalizados. No entanto, como os algoritmos utilizados para a aprendizagem do modelo de SVM incremental necessita de um conjunto de treino e um conjunto de teste, tornar-se-ia necessário normalizar ambos os conjuntos. Para testar o SVM incremental é necessário incrementar novos dados ao conjunto de treino. Estes novos dados, sendo provenientes das medições realizadas durante o processo de tratamento das águas residuais, encontrar-se-iam numa escala completamente desproporcional a escala normalizada. Como tal, preferiu-se não normalizar nenhum dos conjuntos de dados utilizados neste projeto.

Conversão de Tipo de Atributos

A ferramenta RapidMiner permite converter atributos do tipo nominal para numérico e vice-versa. A maioria dos atributos da fonte de dados são do tipo numérico, os atributos não numéricos são a estação do ano (*Season*) e a data (*Date*). A data é o identificador único (*id*) de cada registo e corresponde ao dia de tratamento que é registado. Como tal não necessita de ser convertido pois o algoritmo de previsão não processa este atributo. A conversão da estação do ano para dados numéricos foi realizada no RapidMiner através do operador "*Nominal to Numeric*". Os parâmetros utilizados permitiram selecionar apenas o atributo relativo a estação do ano "*Season*", assim como, o tipo de conversão realizada. Selecionando a opção "*unique integer*" cada estação foi convertida num numérico inteiro identificador de cada estação do ano nomeadamente Outono = 0, Inverno = 1, Primavera = 2, Verão = 3. Outro método seria a conversão em variáveis binárias. O atributo é dividido em 4 novos atributos, um para cada estação do ano. A estação referente a cada registo apareceria como verdadeira (valor 1) e as restantes como falsas (valor 0). Este método poderia ser o mais lógico porém iria aumentar a dimensionalidade da fonte de dados e assim prejudicar o desempenho e o tempo de execução dos algoritmos.

Discretização

Como se pode verificar através do Anexo A, grande parte dos atributos são do tipo numérico e contínuo. Com o RapidMiner criaram-se dois conjuntos distintos de dados a partir de cada conjunto recolhido. O primeiro não foi sujeito a discretização da variável de previsão, que é do tipo numérico e contínuo, com a finalidade de testar algoritmos de regressão do SVM. Para o segundo conjunto de dados realizou-se a discretização da variável de previsão através do operador "*Discretize by user specification*". Nos parâmetros do operador, selecionou-se o atributo *SP6_CBO* (label), e no parâmetro de definição das classes definiram-se as classes -1 e 1. A classe -1 engloba os valores compreendidos entre $[-\infty; 25]$. Por sua vez a classe 1 engloba os valores entre $[25; +\infty]$. Esta divisão baseia-se nas normas legais relacionadas com os valores máximos recomendados (em mg/l) para o parâmetro CBO (Carência Bioquímica de Oxigénio).

Seleção de Atributos

Com a finalidade de reduzir a quantidade de dados irrelevantes para a previsão da variável de interesse, bem como a dimensionalidade do conjunto de dados, procedeu-se a seleção de atributos. Para fazer essa seleção foram testados vários métodos, nomeadamente técnicas de *wrapper* e *filter*. A tabela 8 apresenta os resultados obtidos na aprendizagem dos modelos de SVM e SVR para cada técnica de seleção de atributos utilizados.

Técnica	Nº atributos selecionados	e-SVR		C-SVM			
		Root Mean Squared Error	Correlation	Root Mean Squared Error	Correlation	Acuidade	Margin
Weight by correlation	11	34.044 ± 0.000	**0.008**	0.444 ± 0.000	0.000	76.74%	0.288
Forward Selection	2	**32.278 ± 0.000**	0.000	**0.433 ± 0.000**	0.000	**75.06%**	0.256
Sem seleção de atributos	34	34.047 ± 0.000	0.000	0.444 ± 0.000	0.000	74.68%	**0.345**

Tabela 8 - Resultados com e sem seleção de atributos para o 1º conjunto de dados.

Os resultados apresentados na tabela 8 são relativos a utilização de 2 técnicas de seleção de atributos. A técnica de seleção pelo peso atribuído à correlação de cada atributo com a variável de previsão é uma técnica de *filter*. Nesta técnica, o operador "*Weight by correlation*" atribui um peso a cada atributo consoante a sua correlação com a variável de previsão (*label*). Depois o operador "*Select by weights*" seleciona os atributos com um peso superior ou igual a um valor definido pelo utilizador, neste caso selecionou-se o valor de *0.1*.

A técnica de *Wrapper* utilizada foi a "*Forward Selection*" que, como já foi referido, inclui obrigatoriamente no seu processo a aprendizagem de um modelo. Para estes testes, realizou-se a aprendizagem de um modelo de SVR (*e-SVR*) e outro de SVM (*C-SVM*). Para se poder visualizar o desempenho dos modelos criados a partir de cada técnica de seleção de atributos, utilizou-se o método de "*Cross Validation*" que através de um operador de "*Performance*" avalia a aprendizagem do modelo segundo vários critérios. Os critérios selecionados para a avaliação dos modelos de *e-SVR* criados foram as métricas de desempenho *Root Mean Squared Error* e o *Coeficiente de correlação* (*Correlation*). Para os modelos de *C-SVM* criados, a avaliação do desempenho foi realizado com as mesmas métricas e adicionalmente a acuidade da previsão (Accuracy) e a margem obtida (*Margin*). Os modelos de C-SVM obtiveram melhores resultados,

com valores para a Raiz Quadrada do Erro Médio (*Root Mean Squared Error*) muito inferiores aos modelos de e-SVR.

Dos modelos de C-SVM criados, o que obteve melhores resultados e desempenho foi o modelo criado a partir do método de seleção de atributos "Forward Selection", que selecionou apenas dois atributos relevantes para a previsão dos valores da variável de interesse. De notar que o modelo de C-SVM criado sem seleção de atributos foi o que obteve maior distância de margem de separação das classes apesar da alta dimensionalidade do conjunto de treino. Porém também foi o que mais tempo de execução necessitou para a aprendizagem do modelo devido grande quantidade de dados, logo envolveu também um maior custo computacional. Devido às distorções nos dados e perdas de relacionamento entre atributos provocadas pela geração automática de dados, este conjunto de dados obtém resultados bastante fracos sobretudo com modelos de SVM para regressão. Deste modo, este conjunto não será utilizado nos testes para os modelos de SVM incremental.

6.5. O 2º Conjunto de dados

Caraterização dos dados

O conjunto de dados aqui descrito armazena os valores das medições realizadas por uma ETAR durante o seu processo de tratamento das águas residuais. Inicialmente, o volume de dados foi relativamente pequeno, que apesar do conjunto de dados ser composto por 38 atributos, este possuía inicialmente apenas 527 registos. Analisando o conjunto de dados podemos observar que tinha algumas semelhanças com o conjunto analisado anteriormente. Os parâmetros de medição são semelhantes. Apesar de algumas variáveis não serem as mesmas, como por exemplo a quantidade de Zinco e a condutividade elétrica da água, a variável de previsão (CBO) mantém-se. Porém, este conjunto de dados não toma em consideração a estação do ano associada a cada registo mas regista o fluxo de água que entra diariamente na ETAR. A semelhança do outro conjunto de dados, analisado na secção anterior, cada registo possui uma data, isto é, cada registo é a média das medições realizadas ao longo de um dia de tratamento. O atributo "data" (*Date*) representa assim o identificador único de cada registo. O conjunto de dados também possui atributos relativos a avaliação do desempenho de várias etapas do processo de tratamento dos parâmetros SS, DBO, DQO e SED. De notar que todos os atributos do conjunto de dados são do tipo numérico e continuo.

Atributo	Descrição	Escala de medição
inQ-E	Fluxo de águas residuais que entra na ETAR.	m^3/dia
inZN-E	Indica a quantidade de Zinco à entrada da ETAR.	*(mg/l)*
inPH-E	Indica a acidez, neutralidade ou alcalinidade do afluente à entrada da ETAR.	*Não aplicável*
inDBO-E	Indica a Carência Bioquímica de oxigénio do afluente à entrada da ETAR.	*(mg/l)*
inDQO-E	Indica a Carência Química de oxigénio do afluente à entrada da ETAR.	*(mg/l)*

inSS-E	Indica o valor Total de Sólidos Suspensos no afluente à entrada da ETAR.	*(mg/l)*
inSSV-E	Indica a quantidade de Sólidos Suspensos Voláteis do afluente à entrada da ETAR.	*(mg/l)*
inSED-E	Indica a quantidade de Sólidos sedimentáveis do afluente à entrada da ETAR.	*(mg/l)*
inCOND-E	Indica a condutividade elétrica das águas residuais à entrada da ETAR.	*µS/cm*
PH-P	Indica a acidez, neutralidade ou alcalinidade do afluente à entrada do decantador primário.	*Não aplicável*
inDBO-P	Indica a Carência Bioquímica de oxigénio do afluente à entrada do decantador primário.	*(mg/l)*
inSS-P	Indica o valor Total de Sólidos Suspensos no afluente à entrada do decantador primário.	*(mg/l)*
inSSV-P	Indica a quantidade de Sólidos Suspensos Voláteis do afluente à entrada do decantador primário.	*(mg/l)*
inSED-P	Indica a quantidade de sólidos sedimentáveis do afluente à entrada do decantador primário.	*(mg/l)*
inCOND-P	Indica a condutividade elétrica das águas residuais à entrada do decantador primário.	*µS/cm*
inPH-D	Indica a acidez, neutralidade ou alcalinidade das águas residuais à entrada do decantador secundário.	*Não aplicável*
inDBO-D	Indica a Carência Bioquímica de oxigénio das águas residuais à entrada do decantador secundário.	*(mg/l)*
inDQO-D	Indica a Carência Química de oxigénio das águas residuais à entrada do decantador secundário.	*(mg/l)*
inSS-D	Indica o valor Total de Sólidos Suspensos das águas residuais à entrada do decantador secundário.	*(mg/l)*
inSSV-D	Indica a quantidade de Sólidos Suspensos Voláteis do afluente à entrada do decantador secundário.	*(mg/l)*
inSED-D	Indica a quantidade de sólidos sedimentáveis das águas residuais à entrada do decantador secundário.	*(mg/l)*
inCOND-D	Indica a condutividade elétrica das águas residuais à entrada do decantador secundário.	*µS/cm*
outPH-S	Indica a acidez, neutralidade ou alcalinidade das águas residuais no final do tratamento.	*Não aplicável*
outDBO-S	Indica a Carência Bioquímica de oxigénio das águas residuais no final do tratamento.	*(mg/l)*
outDQO-S	Indica a Carência Química de oxigénio das águas residuais no final do tratamento.	*(mg/l)*
outSS-S	Indica o valor Total de Sólidos Suspensos das águas residuais no final do tratamento.	*(mg/l)*
outSSV-S	Indica a quantidade de Sólidos Suspensos Voláteis das águas residuais no final do tratamento.	*(mg/l)*
outSED-S	Indica a quantidade de sólidos sedimentáveis das águas residuais no final do tratamento.	*(mg/l)*
outCOND-S	Indica a condutividade elétrica das águas residuais no final do tratamento.	*µS/cm*
perfinpRD-DBO-P	Indica a percentagem de eficácia do tratamento da Carência Bioquímica de Oxigénio para o decantador primário.	*%*
perfinpRD-SS-P	Indica a percentagem de eficácia do tratamento de sólidos suspensos para o decantador primário.	*%*
perfinpRD-SED-P	Indica a percentagem de eficácia do tratamento de sólidos sedimentáveis para o decantador primário.	*%*
perfinpRD-DBO-S	Indica a percentagem de eficácia do tratamento da Carência Bioquímica de Oxigénio para o decantador secundário.	*%*
perfinpRD-DQO-S	Indica a percentagem de eficácia do tratamento da Carência Química de Oxigénio para o decantador secundário.	*%*

globalRD-DQO-G	Indica o desempenho global do tratamento da Carência Química de oxigénio.	*%*
globalRD-DBO-G	Indica o desempenho global do tratamento da Carência Bioquímica de oxigénio.	*%*
globalRD-SS-G	Indica o desempenho global do tratamento de sólidos suspensos.	*%*
globalRD-SED-G	Indica o desempenho global do tratamento de sólidos sedimentáveis.	*%*
Date	Data de cada registo	*Não aplicável*

Tabela 8 - Descrição dos Atributos do conjunto de dados – o segundo conjunto.

Para além disso, o número de pontos de amostragem são inferiores. As medições foram realizadas em 4 pontos de amostragem diferentes. Na tabela seguinte pode-se verificar quais os parâmetros medidos em cada ponto de amostragem. Os pontos de amostragem estão representados da seguinte forma: *E,* entrada da ETAR; *P,* entrada do decantador primário; *D,* entrada do decantador secundário; e *S,* final do tratamento.

Parâmetro	Pontos de amostragem
Q	*E;*
ZN	*E;*
PH	*E,P,D,S;*
DBO	*E,P,D,S;*
DQO	*E,D,S;*
SS	*E,P,D,S;*
SSV	*E,P,D,S;*
SED	*E,P,D,S;*
COND	*E,P,D,S;*

Tabela 9 - Pontos de medição de cada parâmetro.

Enriquecimento dos Dados

Apesar do conjunto de dados possuir um maior número de registos do que o conjunto anterior, pensamos que 527 registos possam ser insuficientes para uma aprendizagem eficiente de um modelo de previsão. Para além disso, se o pretendido é testar os modelos criados, uma das métricas de desempenho utilizadas é o tempo de aprendizagem do modelo. Ora quanto maior for o conjunto de treino mais exigente será o processo de aprendizagem. Conjuntos com poucos registos podem ser criados em poucos segundos o que não permite tirar conclusões precisas. Pretende-se então aumentar o tamanho do conjunto de dados. Porém, com a preocupação de evitar a perda de correlação entre atributos, semelhante à ocorrida no conjunto anterior pela geração de novos registos atributo por atributo. Deste modo, o meio mais simples de aumentar o conjunto de dados é copiar os registos sucessivamente. Este método cria novos registos que não trazem enormes melhorias para a aprendizagem do modelo, pois são valores repetidos, mas criam uma exigência suplementar e um maior esforço computacional. Ao proceder a este método, o conjunto de dados ficou com 5178 registos. De referir que este método utilizado para aumentar o volume deste conjunto de dados levou ao aparecimento de datas duplicadas. Como a data é o identificador único de cada registo, este não pode ocorrer em

dois registos distintos. Esta anomalia foi tratada alterando as datas duplicadas dos registos copiados.

Valores Nulos

O conjunto de dados apresentava valores nulos em alguns registos. Estes valores nulos foram tratados com a utilização do RapidMiner. O operador utilizado, a semelhança do conjunto de dados anterior, foi o "*Replace Missing Values*" com o método de substituição de valores nulos pela média de cada atributo.

Outliers

Os *outliers* são considerados valores fora do contexto real e são prejudiciais pois afetam a aprendizagem de um modelo. Com a utilização do Rapidminer procedeu-se a deteção de *outliers* e consequente remoção. O operador "*Detect Outliers*" pela distância dos valores identifica *N outliers* no conjunto de dados com base na distância dos seus *K* pontos vizinhos mais próximos. As variáveis *N* e *K* podem ser especificadas pelo utilizador nos parâmetros do operador. Os valores utilizados para estes parâmetros foram os valores do operador por defeito, *N = 10* e *K = 10.* A função de distância utilizada foi a função Euclidiana. O histograma apresentado na Figura 19 mostra o domínio no espaço da variável de previsão e permite identificar a presença de *outliers*.

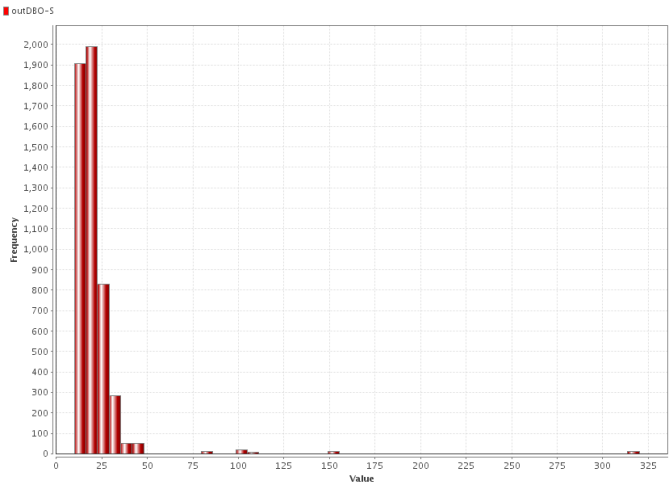

Figura 20 - Histograma que identifica *outliers* na variável de previsão.

Neste histograma é possível identificar a presença de um exemplo irregular na variável de previsão com valor próximo dos 325 mg/l. Esta anomalia refere-se a um dia de tratamento em que os valores de Carência Bioquímica de Oxigénio (*DBO*) no final do tratamento são muito acima do máximo recomendado (25 mg/l). Esta ocorrência pode ser uma consequência de, por exemplo, falhas no instrumento de medição ou no próprio processo de tratamento. Outros casos de presença de *outliers* foram identificados e tratados - os histogramas referentes a esses casos encontram-se nos anexos desta dissertação. No final desta análise, removeram-se os registos considerados *outliers*. Consequentemente, o conjunto de dados ficou com 5168 registos.

Correlação

À semelhança do processo realizado para o primeiro conjunto de dados, realizou-se o estudo do relacionamento colinear entre os atributos. O procedimento seguido foi idêntico ao descrito na secção anterior em que se utilizou uma matriz de correlação. Para este conjunto de dados o critério de interesse é o mesmo. São consideradas importantes correlações entre atributos com um coeficiente de correlação igual ou superior a 0,1. Neste caso, e como se pode verificar na tabela seguinte, existem mais relações colineares com coeficiente superior ou igual a 0,1. Este facto pode ser devido ao aumento do número de registos do conjunto de dados não ter sido realizado por geração automática de atributos, mas sim copiando os registos dos dados. Este método apesar de pouco recomendado era o mais viável para o contexto do projeto e, mesmo assim, é o mais fiável relativamente à preservação das tendências implícitas dos dados e da correlação entre os atributos.

First Attribute	Second Attribute	Correlation
inCOND-P	inCOND-E	0.973
inCOND-D	outCOND-S	0.948
inCOND-P	inCOND-D	0.947
inCOND-D	inCOND-E	0.940
inCOND-P	outCOND-S	0.915
inCOND-E	outCOND-S	0.903
inPH-P	inPH-E	0.902
inPH-P	inPH-D	0.894
inSS-P	inSS-E	0.878
inSSV-E	inSSV-P	0.872
inSED-E	inSED-P	0.847
outSS-S	outDBO-S	0.840
inPH-D	inPH-E	0.822
globalRD-DBO-G	perfinpRD-DBO-S	0.790
perfinpRD-DQO-S	globalRD-DQO-G	0.784
inSS-P	inSED-P	0.725
inDBO-D	inDQO-D	0.719
inDBO-E	inDBO-P	0.710
inSSV-D	inSSV-P	0.691
inSSV-E	inSSV-D	0.683
outSS-S	outDQO-S	0.677
outSED-S	outDBO-S	0.670
inDQO-D	inDQO-E	0.641
outSS-S	outSED-S	0.620
inDBO-D	inDBO-P	0.615
inSS-P	perfinpRD-SS-P	0.598
globalRD-DBO-G	globalRD-SS-G	0.594
perfinpRD-SS-P	perfinpRD-SED-P	0.594

Tabela 10- Representação dos pares de atributos com maiores coeficientes de correlação - segundo conjunto.

Como se pode verificar na Tabela 11, os pares de atributos com coeficientes de correlação mais altos são atributos relativos a medições do mesmo parâmetro, mas em pontos de amostragem diferentes. Este tipo de correlação é normal visto serem provenientes de medições dos mesmos efluentes porém em etapas do tratamento diferentes. No entanto o par de atributos de parâmetros diferentes com maior coeficiente de correlação é o par [*outSS-S; outDBO-S*] com um coeficiente de correlação de 0,840. Como já foi referido, a degradação dos Sólidos Suspensos depende diretamente da presença de oxigénio, daí esta forte correlação entre estes dois atributos.

Discretização

Neste conjunto de dados, todos os atributos são do tipo numérico e contínuo. Deste modo, e como os algoritmos de SVM aprendem sobre dados numéricos, apenas se discretizou a variável de interesse para se poder testar os algoritmos de SVM para a classificação. No entanto guardou-se o conjunto original com a variável de previsão contínua para se poder testar algoritmos de SVM para a regressão. A discretização da variável de previsão foi realizada através do operador "*Discretize by user specification*". Nos parâmetros do operador, selecionou-se o atributo *outDBO-S* (label), e no parâmetro de definição das classes definiram-se as classes -1 e 1. A classe -1 engloba os valores compreendidos entre $[-\infty; 25]$. Por sua vez a classe 1 engloba os valores entre $[25; +\infty]$. Como já foi mencionado no capítulo 2, os valores máximos recomendados de DBO reguladas por lei são *25 mg/l*. Esta divisão baseia-se assim nas normas legais (em mg/l) para o parâmetro DBO (Carência Bioquímica de Oxigénio).

Seleção de Atributos

Este conjunto de dados possui uma dimensionalidade considerada alta para o âmbito deste projeto. Deste modo, pretendeu-se reduzir a quantidade de dados irrelevantes para o objetivo de prever os valores da variável de interesse. O método escolhido foi o mesmo que no primeiro conjunto de dados. Propositadamente, utilizaram-se as mesmas técnicas de seleção de atributos para que, no final, se pudesse comparar os resultados obtidos por ambos os conjuntos de dados. As técnicas utilizadas foram então uma técnica de Wrapper, apresentada no capítulo anterior, chamada "*Forward Selection*", e uma técnica de Filter que seleciona os atributos consoante um peso atribuído, neste caso o peso do coeficiente de correlação. Nesta técnica são selecionados para o subconjunto de dados os atributos com um coeficiente de correlação superior ou igual a 0,1. Para se obter valores concretos do desempenho de cada conjunto de dados realizou-se o processo de aprendizagem de um modelo de SVM para classificação (*C-SVM*) e outro de um modelo de SVM para regressão (*e-SVR*)

Técnica	Nº atributos selecionados	*e-SVR*		*C-SVM*			
		Root Mean Squared Error	Correlation	Root Mean Squared Error	Correlation	Acuidade	Margin
Weight by correlation	17	15.021 ± 0.000	0.464	0.275 ± 0.000	0.971	89.27%	0.349
Forward Selection	3	**12.052 ± 0.000**	**0.583**	**0.274 ± 0.000**	**0.998**	**90.65%**	**0.574**
Sem seleção de atributos	38	15.839 +/- 0.000	0.389	3.277 ± 0.000	0.990	84.74%	0.331

Tabela 11 Resultados com e sem seleção de atributos para o 2º conjunto de dados

Os resultados apresentados na tabela 12 são relativos à utilização de 2 técnicas de seleção de atributos. A técnica de seleção pelo peso atribuído da correlação do atributo é uma técnica de *Filter*. Nesta técnica, o operador "*Weight by correlation*" atribui um peso a cada atributo consoante a sua correlação com a variável de previsão (*label*). Depois o operador "*Select by*

weights" seleciona os atributos com um peso superior ou igual a um valor definido pelo utilizador, neste caso selecionou-se o valor de *0.1*.

A técnica de *Wrapper* utilizada, como já foi referido, inclui obrigatoriamente no seu processo a aprendizagem de um modelo. O desempenho dos modelos foi avaliado com a utilização do método de "*Cross Validation*" que através de um operador de "*Performance*" avalia a aprendizagem do modelo segundo vários critérios. Os critérios selecionados para a avaliação dos modelos de *e-SVR* criados foram as medidas de desempenho *Root Mean Squared Error* e o *Coeficiente de correlação* (*Correlation*). Para os modelos de *C-SVM* criados, a avaliação do desempenho foi realizado com as mesmas métricas e adicionalmente a precisão da previsão (*Acuidade*) e a margem obtida (*Margin*).

Os modelos de SVM para classificação obtiveram resultados bastante superiores aos resultados dos modelos de SVM para regressão. A técnica de seleção de atributos "Forward Selection" obteve uma Raiz Quadrática do Erro Médio (*Root Mean Squared Error*) inferior, e um coeficiente de correlação superior às restantes técnicas de seleção de atributos, e isso tanto nos modelos de C-SVM como os modelos de e-SVR. As medidas de desempenho "Margin", que representa a margem obtida na criação do hiperplano, e a acuidade apenas são calculáveis para modelos de classificação. Neste caso em concreto, A técnica que obteve a maior distância de margem entre as classes e a maior acuidade na previsão continua a ser a técnica "Forward Selection". Comparando os dois conjuntos de dados analisados, nota-se uma grande superioridade no desempenho dos modelos criados com o segundo conjunto de dados. Os modelos de classificação também se mostraram melhores preditores do que os modelos de regressão. Como tal, os resultados dos modelos apresentados a partir deste ponto para algoritmos de SVM incremental serão construídos com a utilização do segundo conjunto de dados como conjunto de treino do modelo de *Data Mining*. Sempre que possível será apresentado uma comparação entre os modelos de classificação e os modelos de regressão.

Formato dos Dados

Durante a realização do projeto, na fase de criação do modelo de SVM incremental, foi necessário realizar a conversão do formato do conjunto de dados. As ferramentas que tinham os algoritmos utilizados para a modelação incremental do SVM não eram compatíveis com conjuntos de dados em formato CSV. Apenas aceitavam conjuntos de dados no formato adotado pela libraria do SVM chamado *LibSVM*. O formato Libsvm é representado da seguinte forma:

<Classe> <nº atributo: valor atributo> <nº atributo: valor atributo> <nº atributo: valor atributo>…
<Classe> <nº atributo: valor atributo> <nº atributo: valor atributo> <nº atributo: valor atributo>…
<Classe> <nº atributo: valor atributo> <nº atributo: valor atributo> <nº atributo: valor atributo>…
<Classe> <nº atributo: valor atributo> <nº atributo: valor atributo> <nº atributo: valor atributo>…

Por exemplo:

1 1:7.8 2:97 3:0.01 4:73.1 5:79 6:92.7 7:100
-1 1:7.8 2:61 3:0.01 4:71.9 5:78.7 6:86.9 7:94
1 1:7.7 2:126 3:0.01 4:60.6 5:67.1 6:85 7:69
-1 1:7.8 2:129 3:0.01 4:80.9 5:82.7 6:92.3 7:34
1 1:7.8 2:158 3:0.01 4:77.3 5:91.5 6:88 7:87

Para proceder a conversão do formato *csv* para o formato *libsvm* utilizou-se um programa, desenvolvido em Python por *Zygmunt*, chamado csv2libsvm. Após a conversão é criado um ficheiro.data com o conjunto de dados em formato libsvm. A execução do programa foi sempre realizada através da linha de comandos com o comando a baixo apresentado:

```
Python csv2libsvm.py ficheiro.csv ficheiro.data "nº da label" true
```

O número da *label* é a posição da coluna que representa a variável de previsão. A contagem das colunas deve iniciar sempre em zero. Por exemplo, dado um conjunto de treino em formato csv chamado *training*, no qual a variável de previsão é a primeira coluna, a conversão é feita da seguinte forma:

```
C:\Users\user\Documents\NetBeansProjects\convert csv to libsvm format\src\python\python csv2libsvm.py training.csv training.data 0 true
```

Da execução do comando anterior resulta um ficheiro training.data, que representa o conjunto de treino em formato libsvm.

Capítulo 7
A Modelação

7.1. Ferramentas e Técnicas de modelação

Os resultados da aplicação de técnicas de SVM são comparáveis e muitas vezes superiores aos obtidos por outros algoritmos de aprendizagem. O desempenho, a adaptabilidade, e a margem de progressão e evolução desta técnica tem chamado a atenção de muitos profissionais da área. Como resultado desse crescente interesse pelas técnicas de SVM, nos dias de hoje, existem diversas ferramentas de *Data Mining* que disponibilizam algoritmos de SVM. Neste projeto, a produção de modelos estritamente não incrementais realizou-se com o Rapidminer. No entanto, esta ferramenta não possibilita a aprendizagem incremental dos modelos de SVM. Como tal, foi necessário pesquisar outras ferramentas direcionadas para esta vertente incremental das técnicas de SVM. A descoberta e implementação de ferramentas que disponibilizam algoritmos de SVM incremental, ao contrário do que se pode idealizar, foi uma tarefa bastante demorosa e trabalhosa, isso pelo facto de ser uma técnica relativamente recente e ainda alvo de diversos estudos.

RapidMiner

Neste projeto, os modelos de SVM na sua vertente não incremental foram construídos a partir do RapidMiner. Esta ferramenta possui vários algoritmos de SVM tanto para classificação como para regressão. Os algoritmos de SVM utilizados são implementações baseadas na biblioteca LibSVM (Chang, Lin, 2011), e denominam-se *C-SVC* e *e-SVR*. Os resultados apresentados no capítulo anterior, nas tabelas 8 e 12, são resultantes da utilização desses mesmos algoritmos de classificação e de regressão respetivamente. O conceito de "universalidade da máquina", introduzido por Vapnik (1998), refere que a aprendizagem dos modelos pode ser realizada com funções de kernel diferentes. Como tal, O RapidMiner permite escolher as funções de kernel que um modelo envolve na sua aprendizagem nomeadamente RBF, polinomial, sigmoid e pré-computado. Neste projeto, todos os modelos de SVM tradicionais e incrementais produzidos envolvem a função de kernel RBF. Esta escolha deve-se a dois fatores. Primeiramente, após uma pesquisa sobre qual a melhor função de kernel a ser utilizada, vários autores revelam que a função RBF é uma boa primeira opção para a produção de um modelo de SVM. Outro fator importante é o facto das ferramentas utilizadas para a implementação de modelos de SVM incremental não suportarem ou não serem estáveis em testes realizados com outros tipos de Kernel.

LIBLINEAR

A LIBLINEAR é uma biblioteca Open Source de algoritmos SVM para aprendizagem de máquina. Inicialmente a ferramenta LIBLINEAR, desenvolvida em C++, era apenas constituída por um algoritmo classificador linear que evoluiu ao longo dos anos. Atualmente fornece já algumas extensões e funcionalidades ao LibSVM. Segundo o autor da ferramenta, Chih-Jen Lin, o objetivo do desenvolvimento da LIBLINEAR foi criar uma ferramenta capaz de lidar com uma grande quantidade de dados. A LIBLINEAR suporta regressão logística e o parâmetro de perda das SVM lineares. A criação da LIBLINEAR baseia-se na libraria LIBSVM (Chang e Lin, 2011) e, como tal, herda várias caraterísticas, tais como a fácil utilização, vasta documentação, e uma licença *open source*. Uma das extensões do LIBLINEAR desenvolvida por Tsai, Lin, C.Y., e Lin, C. J. é uma vertente incremental do SVM. Esta extensão utiliza a estratégia de "Warm-Start" apresentada no capítulo 5 que consiste em atualizar eficientemente um modelo já treinado.

No âmbito deste projeto, os testes com esta ferramenta foram realizados através da linha de comandos. A LIBLINEAR divide-se em dois comandos básicos: *"train"* e *"predict"*. O comando *"train"* realiza a aprendizagem do modelo com o conjunto de treino, e o comando "*predict*" realiza a previsão dos dados de teste e avaliação do modelo. Relativamente a avaliação dos modelos, a LIBLINEAR retorna o valor da "Acuidade" e possui ainda o método *k-Fold Cross Validation* da validação cruzada. No ficheiro gerado automaticamente pelo comando "*predict*" é possível verificar os valores de previsão e a apreciação da probabilidade estimada para cada classe de cada valor. Relativamente aos dados, o formato adotado pela ferramenta LIBLINEAR é o formato LIBSVM. Um exemplo da utilização do LIBLINEAR tem a seguinte forma:

1. *C:\...\liblinear\train [opções] trainset.file*
2. *C:\...\liblinear\predict [opções] test.file model.file output.file*

As opções de treino e de previsão do modelo podem ser consultadas na Tabela 13.

Train		
Opção	**Descrição do comando**	**Exemplo**
-s n	Escolha do algoritmo de SVM	-s 0; -s 11
-c n	Custo da violação de restrições "Parâmetro C"	-c 10 ; -c 5
-p n	Sensibilidade da perda do SVR	-p 1; -p 2
-e n	Critério de paragem	-e 0.01; -e 0.1
-wi n	Peso de ajuste do parâmetro C para diferentes classes	-w1 2; -w2 5; -w3 2
-i	Atualização incremental de um modelo	-i modelo.data
-v n	Avaliação com o método k-fold Cross Validation.	-v 10
-q	Modo sem ficheiro de output	-q
Predict		
Opção	**Descrição do comando**	**Exemplo**
-b	Ficheiro de output com (1) ou sem (0) probabilidade estimada de cada valor de previsão.	-b 0; -b 1

Tabela 12 - Opções para os comandos do LIBLINEAR.

A adição de novos exemplos no conjunto de dados pode trazer variações nas tendências e nos padrões assimilados na aprendizagem do modelo. Quando tal acontece, a LIBLINEAR fornece duas opções, *batch* e incremental, para atualizar o conhecimento extraído dos dados. Primeiramente, o modo *batch* é voltar a treinar um modelo com o novo conjunto de treino (dados antigos + dados novos). O método incremental utiliza a estratégia "*Warm-Start*" para criar um novo modelo atualizado a partir de um modelo já existente evitando um reprocessamento de todo o conjunto de dados. De notar que atualmente, a ferramenta LIBLINEAR apenas suporta aprendizagem incremental de modelos lineares (sem dualidade) para classificação"*-s 0" e "-s 2"*. Um exemplo da utilização da LIBLINEAR é:

1. *C:\...\liblinear\train –s 0 –v 10 trainset.file*

2. *C:\...\liblinear\predict –b 1 testset.file trainset.model resultado.output*

A instrução 1 utiliza o algoritmo de classificação linear para realizar a aprendizagem do modelo "trainset.model". É ainda utilizado o método *k-fold Cross Validation* com k = 10 para avaliar o modelo e retorna o valor da acuidade do método utilizado de validação cruzada. Na instrução 2, é realizada a previsão dos valores do conjunto "*testset.file*" utilizando o modelo de previsão "*trainset.model*". Esta instrução retorna um ficheiro "*resultado.output*" com os valores de previsão com probabilidade estimada e o valor da Acuidade da previsão do conjunto de teste.

Modo *Batch*

No modo *batch*, quando se pretende atualizar um modelo, os novos dados são adicionados ao conjunto de treino criando o ficheiro "*trainset+newdata.file*" e todo o processo de aprendizagem é executado do início criando um modelo completamente novo.

O procedimento para atualizar o modelo no modo batch é o seguinte:

3. *C:\...\liblinear\train -s 0 -v 10 trainset+newdata.file*

4. *C:\...\liblinear\predict -b 1 testset.file trainset+newdata.model resultado.output*

Modo Incremental

No modo incremental, a atualização do modelo de previsão começa a partir de um modelo já criado. O conjunto de dados de treino é formado pelos dados antigos e pelos novos dados tal como no método anterior. O procedimento incremental para atualizar um modelo quando surgem novos dados é o seguinte:

5. *C:\...\liblinear\train -s 0 -v 10 -i trainset.model trainset+newdata.file*

6. *C:\...\liblinear\predict -b 1 testset.file trainset+newdata.model resultado.output*

Os ficheiros do tipo *model* gerados na aprendizagem do modelo de SVM guarda informações acerca do modelo de SVM, nomeadamente o algoritmo utilizado, o número de classes, as possíveis classes (-1 e +1), o número de atributos, o valor da Bias e o peso de cada atributo. O descritivo do modo de utilização, a lista de comandos da ferramenta LIBLINEAR disponibilizados pelos autores, e exemplos de ficheiro output e modelo encontram-se nos anexos.

TinySVM

A ferramenta TinySVM é uma implementação de SVM proposto por Vapnik (Vapnik,1995 e 1998) para o problema de reconhecimento de padrões. As SVM são uma nova geração de algoritmos de aprendizagem, resultantes dos avanços recentes na teoria de aprendizagem estatística, que hoje são já aplicadas a um grande número de aplicações do mundo real. Esta ferramenta suporta aprendizagem de modelos de SVM para classificação e para regressão. Uma vantagem relativamente a ferramenta LIBLINEAR é que com esta ferramenta é possível produzir modelos com diferentes tipos de Kernel, nomeadamente Linear, Polinomial, RBF, Neural e Anova. Os algoritmos SVM suportados são o *C-SVM* e o *C-SVR*. A semelhança do LIBLINEAR, a aprendizagem incremental dos modelos realiza-se iniciando com um modelo já criado. O formato de dados suportado é o formato libsvm. Nesta ferramenta existem três comandos básicos, nomeadamente "*SVM_Learn*", "*SVM_Classify*", *"SVM_Model*".

SVM_LEARN

1. *C:\...\TinySVM\svm_learn [opções] trainset.file model.file*

Esta instrução realiza a aprendizagem do modelo de SVM. Nesta instrução é necessário nomear um modelo que será criado durante a sua execução. As opções são os parâmetros de ajuste do modelo. A execução desta instrução gera um ficheiro.model, e apresenta na linha de comandos o número de iterações, o número de registos do conjunto de treino, o tempo de execução, o valor máximo de violação das KKT (condições de Kuhn Tucker) e a taxa de acerto.

SVM_CLASSIFY

1. *C:\...\TinySVM\svm_classify [opções] testset.file model.file*

Esta instrução realiza a previsão dos valores do conjunto de teste a partir de um dado modelo de previsão. A execução desta instrução pode retornar, consoante as opções selecionadas, a acuidade da previsão e os valores da matriz de confusão.

SVM_MODEL

2. *C:\...\TinySVM\svm_model [opções] model.file*

Esta instrução retorna as informações guardadas no modelo "*model.file*". As opções representam as informações que se pretende visualizar. A tabela seguinte apresenta todas as opções para cada instrução anteriormente apresentadas para o TinySVM.

SVM_LEARN		
Comando	**Descrição do comando**	**Exemplo**
-l n	Escolha do algoritmo de SVM	-l 0; -l 1
-t n	Escolha do tipo de kernel	-t 0 ; -t 4
-c n	Custo da violação de restrições "Parâmetro C"	-c 1; -c 2
-e n	Critério de paragem	-e 0.01; -e 0.1
-I	Cria um ficheiro Model.idx para a atualização incremental	-I trainset.file
-M	Atualização incremental de um modelo	-M modelo.model
SVM_CLASSIFY		
Comando	**Descrição do comando**	**Exemplo**
-V	Visualização dos resultados detalhados	-V
SVM_MODEL		
Comando	**Descrição do comando**	**Exemplo**
-n	Apresenta o número de Vetores de Suporte	-n
-l	Apresenta o Risco Empírico	-l
-t	Apresenta o tamanho do conjunto de treino	-t
-m	Apresenta a margem estimada	-m
-d	Apresenta a dimensão VC estimada	-d
-x	Apresenta o coeficiente α para cada x_i	-x

Tabela 13- Opções de configuração para os comandos do TinySVM

Tal como no LIBLINEAR, o TinySVM suporta aprendizagem incremental de modelos de SVM. De seguida é apresentada um exemplo de atualização do modelo em modo batch e em modo incremental.

Modo Batch

1. *C:\...\TinySVM\svm_learn -l 0 -t 3 trainset+newdata.file modelo.model*

2. *C:\...\TinySVM\svm_classify testset.file modelo.model*

A atualização em modo *i* do modelo de SVM cria um novo modelo completamente independente de qualquer outro modelo já criado.

Modo incremental

1. *C:\...\TinySVM\svm_learn -l 0 -t 3 -I trainset.file modelo.model*

Esta instrução realiza a aprendizagem de um modelo de SVM tradicionalmente efetuada. No entanto, a execução desta operação com a opção "-I" gera dois ficheiros para o modelo de SVM,

modelo.model e *modelo.idx.* Esses ficheiros são essenciais para a aprendizagem incremental de um modelo.

2. *C:\...\TinySVM\cat newdataset.file >> trainset.file*

O comando "*cat*" realiza a união dos dois conjuntos de dados no conjunto "trainset.file".

3. *C:\...\TinySVM\svm_learn -l 0 -t 3 -M modelo.model -I trainset.file modelo2.model*

Esta instrução realiza a aprendizagem incremental do modelo "*modelo2.model*" a partir do "*modelo.model"* e do conjunto de treino alterado no comando anterior *"trainset.file".* A opção "-I*"* deve ser sempre selecionada para posteriores atualizações dos modelos, caso contrário, não será gerado o ficheiro *"modelo.idx*" que é necessário para a atualização incremental de um modelo. A opção *"-M*" especifica qual o modelo de previsão que servirá de condição inicial para a criação do modelo incremental.

4. *C:\...\TinySVM\svm_classify testset.file modelo2.model*

A instrução *4* realiza a previsão dos valores do conjunto de teste *"testset.file"* utilizando o modelo de previsão SVM *"modelo2.model".*

A semelhança da ferramenta LIBLINEAR, o TinySVM suporta a aprendizagem incremental de um modelo de SVM. O procedimento é semelhante pois ambos reutilizam modelos para a produção de um novo modelo atualizado. Porém a grande diferença reside no próprio modelo que é gerado. O TinySVM gera um modelo com toda a informação relativa às opções dos parâmetros de otimização do modelo, tais como o algoritmo de SVM selecionado, o tipo de *kernel*, o parâmetro C, entre outros. Para além dos parâmetros, o modelo guarda ainda todos os vetores de suporte, e para cada um deles, o peso do seu coeficiente α, introduzido no capítulo 5. O ficheiro *modelo.idx* que permite a aprendizagem incremental é um ficheiro de anexo que lista o coeficiente α e o gradiente *G* de todos os exemplos de treino.

Resumindo, neste projeto foram utilizadas três ferramentas que permitem realizar a aprendizagem de um modelo de previsão, nomeadamente o RapidMiner, a LIBLINEAR, e o TinySVM. O RapidMiner é, sem dúvida alguma, a ferramenta mais completa, pois possui diversos operadores com algoritmos de SVM e de SVR dos quais se pode otimizar todos os parâmetros manual e automaticamente, como por exemplo o tipo de kernel. O RapidMiner é também a ferramenta mais intuitiva e fácil de utilizar de entre as três ferramentas e permite um vasto leque de opções para realizar a avaliação dos modelos produzidos. No entanto, esta ferramenta não é adequada para os objetivos deste projeto, pelo simples facto de que não permite realizar a atualização incremental dos modelos de SVM. Relativamente as ferramentas LIBLINEAR e TinySVM, as suas execuções são realizadas através da linha de comandos o que não é de todo intuitivo. Para além disso, são ferramentas bastante limitadas a nível de funcionalidades. Uma vantagem comparativamente ao RapidMiner é a possibilidade de realizar a aprendizagem incremental de modelos de previsão.

O LIBLINEAR possui diversos algoritmos de SVM, no entanto, apenas permite produzir modelos lineares, não sendo possível alterar o tipo de *kernel* da aprendizagem. O mesmo acontece com o procedimento incremental que não suporta algoritmos de classificação com dualidade e algoritmos de SVM para regressão. Apenas pode ser utilizado com algoritmos de classificação primal. Relativamente a avaliação dos modelos, o LIBLINEAR realiza o cálculo de algumas medidas de desempenho e apresenta algumas caraterísticas dos modelos que permite comparar os vários modelos criados, tais como a acuidade, o número de iterações do processo de aprendizagem, o número de vetores de suporte.

O TinySVM comparativamente ao LIBLINEAR não possui tantos algoritmos SVM. Apenas possui um algoritmo de classificação e outro de regressão. No entanto, o TinySVM permite a escolha do tipo de kernel para além dos demais parâmetros de configuração da aprendizagem. Relativamente a avaliação dos modelos, as medidas de desempenho que podem ser calculadas no TinySVM são a acuidade, a matriz de confusão, o tempo de aprendizagem, o número de vetores de suporte, o número de iterações, a margem de separação, a dimensão VC, o valor do coeficiente de Lagrange α de cada registo.

Os modelos criados por ambas as ferramentas também diferem entre si. Enquanto os modelos de classificação produzidos pelo LIBLINEAR armazenam apenas os parâmetros que são definidos pelo utilizador e o peso de cada atributo. O TinySVM guarda nos seus modelos o algoritmo selecionado, o tipo de kernel, e todos os parâmetros utilizados para a modelação. Para além disso, os modelos produzidos pelo TinySVM armazenam todos os vetores de suporte, e para cada um, o seu coeficiente do multiplicador de Lagrange α. Para além da informação contida no próprio modelo, o TinySVM gera um ficheiro índex, chamado *modelo.idx*, que armazena o valor de α e o gradiente *G* de todos os dados do conjunto de treino. No capítulo seguinte são apresentados e comparados os resultados dos testes realizados em cada ferramenta.

Capítulo 8

Testes e resultados

Neste capítulo analisa-se os modelos produzidos através dos processos de *Data Mining* descritos no capítulo anterior e compara-se os resultados obtidos por cada um dos modelos de SVM. Para isso, serão analisadas as várias medidas de desempenho dos modelos como o coeficiente de correlação, a margem de separação das duas classes, e o tempo de aprendizagem do modelo. Pretende-se, pois, produzir modelos SVM para a variável de previsão CBO, em cada uma das três ferramentas descritas no capítulo anterior e comparar os resultados obtidos pelos modelos. De notar que os modelos de previsão incrementais apenas podem ser produzidos nas ferramentas LIBLINEAR e TinySVM. Como a ferramenta LIBLINEAR não suporta aprendizagem incremental de modelos de regressão apenas serão produzidos modelos de classificação.

Para a realização destas tarefas considerou-se apenas o segundo conjunto de dados, extraído de *UCI Machine Learning Repository*. O primeiro conjunto de dados inicialmente analisado não era adequado para as tarefas que foram realizadas devido a perda de relacionamento entre os atributos causada pela geração automática de cada atributo isoladamente. O conjunto de dados selecionado para a realização destas tarefas de modelação foi sujeito ao processo de preparação de dados no qual se efetuou a seleção de atributos. O método de seleção de atributos utilizado foi o "*Select by weight*" em que o peso de cada atributo é obtido comparando a correlação de cada atributo com a variável de previsão. O método "*Forward Selection*" obteve resultados ligeiramente superiores, porém a quantidade de dados, após a seleção de atributos, era pouco significativa, reduzindo a dimensionalidade dos dados em demasia. Os atributos selecionados foram os atributos com um coeficiente de correlação superior ou igual a 0,1. Vários autores afirmam que um bom índice de correlação com a variável de previsão é essencial para se obter um modelo de previsão eficiente e com um bom desempenho. Os atributos mais corelacionados com a variável CBO (do inglês - DBO) foram os listados na tabela seguinte.

attribute	weight
outSS-S	1
globalRD-SED-G	0.853
outSED-S	0.798
globalRD-SS-G	0.789
outDQO-S	0.707
globalRD-DBO-G	0.608
perfinpRD-DBO-S	0.559
globalRD-DQO-G	0.381
perfinpRD-DQO-S	0.284
outPH-S	0.176
inDQO-D	0.175
inDBO-D	0.152
inSS-D	0.126
inDBO-E	0.110
perfinpRD-DBO-P	0.109
inPH-D	0.108

Tabela 14- Atributos selecionados para as tarefas de modelação para a variável de previsão CBO.

Como se pode verificar na Tabela 15, os atributos mais correlacionados com a Carência Bioquímica de Oxigénio (CBO, do inglês DBO) são variáveis dos parâmetros de medições relativas aos sólidos, suspensos e sedimentados. Um dos métodos utilizado pelas ETAR para dissolver estes sólidos utiliza bactérias aeróbias para os decompor. Este processo sendo aeróbio consume grandes quantidades de oxigénio. Para além destes atributos, pode-se verificar que a Carência Química de Oxigénio (CQO, do inglês DQO) e o PH também se encontram fortemente corelacionada com a variável de previsão. Outro atributo selecionado para o conjunto de dados é o atributo *Date* que é o identificador único de cada registo.

Para avaliar os modelos criados com o RapidMiner não foi necessário criar um conjunto de teste pois a própria ferramenta possui os seus métodos de partição do conjunto de dados e de avaliação dos modelos. Porém, a ferramenta LIBLINEAR apenas possui um método *k-fold Cross Validation*. Os seus resultados da avaliação com recurso a este método de validação cruzada são posteriormente comparados com os resultados obtidos com um conjunto de teste criado manualmente. Já a ferramenta TinySVM obriga a criação manual de um conjunto de teste visto não possuir nenhum método de divisão do conjunto total de dados.

Conjunto de Teste

O conjunto de teste é um conjunto de dados utilizado para testar um modelo de previsão. Os valores que serão previstos pelo modelo são os valores da variável de previsão do conjunto de teste. A avaliação do modelo de previsão é realizada comparando os valores previstos com os valores reais do conjunto de teste. Este conjunto deve possuir os mesmos atributos que o conjunto utilizado para a aprendizagem do modelo. Neste caso, o conjunto de teste foi desenvolvido a partir do conjunto de treino. Para tal foram copiados alguns exemplos aleatoriamente do conjunto de treino. Estes exemplos sofreram depois algumas alterações nos seus valores, nomeadamente no atributo Date que, como já foi mencionado, é o identificador de cada registo e como tal não pode ser duplicado. O conjunto de treino é então composto por 5168 registos e o conjunto de teste por 898 registos.

8.1. Previsão de CBO

RapidMiner

Os modelos produzidos no RapidMiner são modelos relativos à previsão de CBO. Com estes modelos pretende-se comparar os métodos de SVM para classificação e métodos de SVM para regressão. As medidas de desempenho utilizadas para comparar os modelos foram o RMSE, o coeficiente de correlação, o erro absoluto e o número de vetores de suporte. Para os modelos de classificação são ainda apresentados os valores das medidas da margem e da acuidade para posteriormente compara-los com os valores obtidos no TinySVM e no LIBLINEAR. Os algoritmos utilizados para a aprendizagem dos modelos de classificação e para os modelos de regressão foram respetivamente o *C-SVC* e o *e-SVR*.

A seleção dos parâmetros de configuração dos modelos de SVM e do próprio *kernel* é necessária para obter um bom desempenho previsional. Inicialmente foram produzidos modelos de previsão com os parâmetros predefinidos pelo RapidMiner. Porém o processo de seleção e otimização de parâmetros pode resultar em uma acuidade superior. Existem diversos métodos automáticos para a seleção dos valores dos parâmetros da função de kernel. Uma das técnicas mais utilizadas para a otimização dos parâmetros de forma automática é a procura em grelha. No contexto de aprendizagem de máquina, a procura em grelha é o processo de procura aprofundada sobre um subconjunto do espaço de trabalho.

Neste projeto apenas foram produzidos modelos com o *kernel* do tipo linear e RBF pois é o *kernel* suportado pelas 3 ferramentas de modelação. Diferentemente do *kernel* linear, o *kernel* RBF possibilita a resolução de problemas inicialmente não separáveis linearmente através do mapeamento para um espaço de maior dimensão. No caso específico de seleção dos parâmetros do kernel do tipo *RBF* a procura é realizada num espaço formado por dois parâmetros para otimizar a aprendizagem do classificador, são eles: gama (γ) e custo (C). Cada coordenada desse espaço é formada por um par ordenado (γ, C). O objetivo é procurar exaustivamente o par (γ, C) para o qual a acuidade do modelo é maior. Como o RapidMiner permite a procura em grelha para a otimização de parâmetros de configuração dos modelos, realizou-se a comparação de modelos produzidos com e sem otimização de parâmetros e avaliaram-se os modelos através do método de validação cruzada.

	e-SVR				*C-SVC*					
	Vetores de suporte	**RMSE**	**Correlação**	**Erro absoluto**	**Vetores de suporte**	**RMSE**	**Correlação**	**Erro Absoluto**	**Acuidade**	**Margem**
CBO	4194	15.614	0.400	3.571 ± 15.200	870	0.261	0.936	0.253 ± 0.065	88.38%	0.341

Tabela 15 - Avaliação de modelos produzidos no RapidMiner sem otimização de parâmetros.

Na tabela 16 são apresentados os resultados obtidos na previsão da variável de interesse *outDBO-S,* com a utilização dos algoritmos e-SVR para a produção de modelos de regressão e C-

SVC para modelos de classificação. Os resultados apresentados são resultantes da utilização destes algoritmos de modelação sem a otimização dos parâmetros de configuração dos modelos e do kernel.

A aprendizagem do modelo de *e-SVR* realizou-se com os parâmetros de configuração por defeito. Isto é, o kernel é do tipo *RBF*, o parâmetro de custo da violação das restrições é $C = 0.0$, o valor do parâmetro gama é $\gamma = 0.0$, o critério de paragem da aprendizagem é $e = 0.001$, e a tolerância da função de perda é $p = 0.1$. Como se pode verificar na tabela anterior, a aprendizagem do modelo identificou 4194 vetores de suporte. Este número é elevado, porém expectável por se tratar de um modelo de regressão. O RMSE foi de 15,614 (mg/l) o que representa um desvio médio bastante elevado relativamente aos valores reais das medições de CBO, cujo limite máximo de concentração é 25 mg/l e a média dos valores reais registados nas medições é 20.46 mg/l. O coeficiente de correlação foi de 0.400 e o erro médio absoluto foi de 3.571 ± 15.200.

OS modelos produzidos com o algoritmo de classificação C-SVC obtiveram resultados bastante superiores aos modelos de regressão. O número de vetores considerados vetores de suporte foi de 870. O valor do RMSE para os modelos originados a partir do C-SVC foi 0.261 com uma correlação de 0.936. O desvio médio absoluto foi de 0.253 ± 0.065. Neste caso os valores não são referentes ao desvio da concentração de CBO, mas do desvio da probabilidade estimada para as classes 1 e -1. Outras medidas de desempenho foram registadas para o modelo de C-SVC, nomeadamente a acuidade que obteve um valor de 88.38% exemplos corretamente classificados com uma margem de separação das duas classes de 0.341. Para estes resultados, o modelo foi produzido com uma configuração predefinida dos parâmetros do modelo e do kernel do tipo RBF, nomeadamente $C = 0.0, \gamma = 0.0$,e $e = 0.001$.

Com a finalidade de comparar os benefícios da otimização de parâmetros de configuração dos modelos, procedeu-se a otimização dos parâmetros γ, C, e, e p. Os parâmetros foram automaticamente ajustados através do método de procura em grelha disponível no RapidMiner. Desta operação resultaram melhorias significativas nos modelos de previsão como se pode constatar nas medidas de desempenho apresentadas na tabela seguinte.

	e-SVR				***C-SVC***					
	Vetores de suporte	**RMSE**	**Correlação**	**Erro absoluto**	**Vetores de suporte**	**RMSE**	**Correlação**	**Erro Absoluto**	**Acuidade**	**Margem**
CBO	2286	12.120 $\pm$ 7.711	0.702	2.312 $\pm$ 1.011	941	0.321 $\pm$ 0.002	0.997	0.317$\pm$ 0.002	95.41%	0.635

Tabela 16 - Avaliação de modelos produzidos no RapidMiner com otimização de parâmetros.

Como se pode ver na tabela anterior, os resultados obtidos pelos modelos de previsão com otimização de parâmetros foram superiores aos modelos produzidos com os parâmetros predefinidos pelo RapidMiner. Para os modelos de C-SVC e de e-SVR, realizou-se a otimização da configuração dos parâmetros por procura em grelha. A otimização do valor dos parâmetros γ, *C,*

e, e *p* foi demorado devido ao elevado número de combinações possíveis, sendo que se limitou o valor de cada parâmetro para valores entre 0 e 2. Para o modelo produzido através do algoritmo C-SVC, os valores para cada parâmetro que em conjunto produziu o modelo com melhores resultados foram $C = 0.0$, $e = 0.1$, $\gamma = 0.667$, $p = 0.0$. O modelo de classificação com otimização dos parâmetros de configuração da aprendizagem e do kernel RBF obteve uma acuidade de 95,41%, uma correlação de 0.997, e uma margem de 0.635. No entanto, o RMSE e o Erro Médio Absoluto registaram uma ligeira subida dos desvios entre os valores previstos e os valores reais. Os modelos de regressão com otimização de parâmetros também obtiveram melhores resultados do que os modelos com os parâmetros predefinidos. Neste modelo em concreto, os valores dos parâmetros após otimização foram $C = 2.0$, $e = 0.1$, $\gamma = 0.0$, $p = 0.734$. O modelo produzido apresenta uma diminuição do desvio médio absoluto e do RMSE para 2.312 e 12.120 respetivamente. De notar que também se verifica um aumento do coeficiente de correlação para 0.702.

LIBLINEAR

A produção de modelos de previsão através da ferramenta LIBLINEAR permite comparar os tipos de atualização de modelos várias vezes referidos neste documento, nomeadamente modelos com atualização incremental. Assim, de seguida é feita uma comparação dos processos de atualização dos modelos quando surgem novos dados. A vertente incremental do LIBLINEAR utiliza a estratégia de Warm-Start, isto é, inicialização a quente para criar um novo modelo de SVM a partir de um modelo já criado. O conjunto de teste foi criado a partir de alguns exemplos do conjunto de treino. Estes foram posteriormente alterados para criar novas tendências e padrões nos dados simulando a necessidade real de atualização do modelo de SVM.

Uma das alterações realizadas foi no atributo *outSS-S* que possui um coeficiente de correlação com a variável de previsão igual a *1*, e no qual se aumentou alguns valores para um intervalo de *[200;400]*. Outra alteração foi no valor do pH que diminui com o aumento do CBO. Estas alterações foram feitas em 40% dos registos do conjunto de teste para testar o modelo previsional em caso de novos dados com tendências irregulares. O conjunto de teste é assim composto por 898 registos. A tabela seguinte compara os resultados obtidos pelos modelos, tradicional e incremental, de SVM para classificação.

	Nº de registos	Nº de iterações	Acuidade
Conjunto de treino	5168	22	74,89%
Conjunto de treino +novos dados (batch)	5399	26	81,07%
Conjunto de treino +novos dados (incremental)	5399	14	80,18%
Conjunto de teste	**898**		

Tabela 17 – Resultados da comparação dos modelos produzidos com atualização *batch* e incremental no LIBLINEAR - conjunto de treino com 5168 registos.

O algoritmo utilizado para a aprendizagem dos modelos de previsão foi o *L2-regularized L2-loss support vector classification (primal).*O único parâmetro de configuração do algoritmo manualmente definido foi o critério de paragem $e = 0.0001$. Esta alteração trouxe melhorias significativas na acuidade dos modelos criados na ordem dos 30%. Os valores do custo *C*, da função de perda *p,* e da bias *B foram os valores predefinidos* $c = 1$, $p = 0.1$, $B = -1$. Como se pode analisar na tabela anterior, o processo de aprendizagem terminou ao fim de 22 iterações. O modelo produzido com o conjunto de treino, com 5168 registos, obteve uma acuidade de 74,89%. A acuidade é bastante inferior aos outros modelos devido ao conjunto de teste possuir valores com alterações, como já foi referido, o modelo não se encontra preparado para reconhecer essas novas tendências nos dados, torna-se necessário atualizar o modelo de previsão.

Com a finalidade de testar a atualização dos modelos procedeu-se a adição de 231 novos registos ao conjunto de treino com alterações nos dados semelhantes ao conjunto de teste, isto é, aumento dos valores de *outSS-S* e diminuição dos valores de *outPH-S*. Pode-se verificar que o método de atualização tradicional ou *batch* obteve uma acuidade de 81,07% que é ligeiramente superior ao modelo produzido a partir do método de atualização incremental que apresentou uma acuidade de 80,18%. No entanto, o número de iterações necessárias para a realização da aprendizagem dos modelos é bastante diferente. A aprendizagem incremental do modelo terminou ao fim de 14 iterações enquanto que a aprendizagem *batch* terminou após 22 iterações.

No contexto das ETAR, os conjuntos de dados são em volume muito maiores e surgem novos dados com elevada variedade de valores todos os dias. Pensa-se que com conjuntos de dados de treino com maior volume, a diferença entre os modelos batch e incremental será mais visível. Pretende-se por isso realizar em seguida os mesmos testes, porém com um conjunto de treino com 103350 registos, um conjunto de novos dados adicionados ao conjunto de treino de 5179 registos e um segundo conjunto de novos dados com 529 registos posteriormente adicionados ao conjunto de treino + conjunto novos dados. A tabela seguinte demonstra os resultados obtidos.

	Nº de registos	**Nº de iterações**	**Acuidade**
Conjunto de treino	103350	126	75,34%
Conjunto de treino +novos dados (batch)	103350+ 5179 =108529	132	81.51%
Conjunto de treino +novos dados (incremental)	103350+ 5179 =108529	25	79.98%
Conjunto de treino + novos dados + novos dados (2) (batch)	103350+ 5179+ 529 = 109058	137	81,17%
Conjunto de treino + novos dados + novos dados (2) (incremental)	103350+ 5179+ 529 = 109058	10	80,06%
Conjunto de teste	**898**		

Tabela 18 - Resultados da comparação dos modelos produzidos com atualização batch e incremental no LIBLINEAR - conjunto de treino com 103350 registos.

Os resultados apresentados na tabela anterior resultam dos modelos produzidos com conjuntos de dados de maior volume. Verifica-se um aumento da diferença entre os modelos com atualização *batch* e incremental. O modelo treinado a partir do conjunto de treino com 103350 registos obteve uma acuidade de 75,34% e o processo de aprendizagem terminou ao fim de 126 iterações. A atualização *batch*, como mencionado anteriormente, é o reprocessamento do conjunto total de dados produzindo um novo modelo completamente independente de outros modelos. Realizou-se então a união do conjunto de treino com os novos dados (conjunto de treino + novos dados). A posterior aprendizagem de um novo modelo terminou ao fim de 132 iterações e obteve uma acuidade da previsão do conjunto de teste de 81,51%. Relativamente ao método incremental, a atualização do modelo apresentou um custo computacional bastante reduzido para a extração de conhecimentos sobre os novos dados, comparado com o método *batch*, terminando o seu processo no fim de 25 iterações. No entanto de referir que obteve uma acuidade de 79,98% que é ligeiramente inferior comparado com o modelo *batch* criado de raiz.

Para se ter uma maior certeza sobre qual o melhor processo de atualização para o contexto das ETAR realizou-se o mesmo processo de atualização dos modelos com um novo conjunto mais pequeno de novos dados. As ETAR registam diariamente inúmeras variações nos constituintes dos efluentes o que levaria a necessidade constante de atualização dos modelos de SVM com pequenos conjuntos de dados. Foi este processo que se tentou simular. A adição de um conjunto com 529 registos ao conjunto total de treino com 108529 registos e as posteriores atualizações, *batch* e incremental, dos modelos obteve resultados ainda mais distintos. O processo *batch* de aprendizagem de raiz de um modelo terminou no final de 137 iterações e obteve uma acuidade de previsão do conjunto de teste de 81,17%. Já o processo de atualização incremental do modelo terminou no fim de 10 iterações e obteve uma acuidade de 80,06%.

TinySVM

Outra ferramenta de modelação utilizada neste projeto é o TinySVM. Esta ferramenta permite a produção de modelos de SVM para classificação e para regressão. O algoritmo de SVM para classificação utilizado em todos os modelos de previsão produzidos foi o C-SVM com a função de kernel RBF. Os testes realizados têm como objetivo avaliar os métodos de atualização de modelos de previsão que o TinySVM oferece, nomeadamente o método incremental. Para essa avaliação foram utilizados o tempo de execução do processo de aprendizagem e a acuidade da previsão como medidas de desempenho. Os conjuntos de treino e o conjunto de teste foram os conjuntos utilizados nos testes realizados na ferramenta LIBLINEAR. A tabela seguinte compara os resultados obtidos pelos modelos, batch e incremental, de SVM para classificação.

	Nº de registos	Tempo de execução	Acuidade
Conjunto de treino	5168	00:00:01	72,96%
Conjunto de treino +novos dados(batch)	5399	00:00:02	81,96%
Conjunto de treino +novos dados (incremental)	5399	00:00:02	81,96%
Conjunto de teste	**898**		

Tabela 19 - Resultados da comparação dos modelos produzidos com atualização batch e incremental no TinySVM - conjunto de treino com 5168 registos.

Os resultados apresentados na tabela anterior são relativos aos modelos de previsão produzidos com o algoritmo de classificação C-SVM. Os parâmetros de configuração do algoritmo utilizados foram o kernel do tipo RBF, e os restantes predefinidos pela ferramenta, nomeadamente, $c = 1$,e $e = 0.001$. A opção "–I" é utilizada para produzir o ficheiro *model.idx* que contém todos os coeficientes dos multiplicadores de Lagrange α e os gradientes *G* de todos os registos do conjunto de treino. Esta opção é utilizada para o processo de atualização do modelo incremental. Primeiramente procedeu-se a aprendizagem do conjunto de treino, que possui 5168 registos. Este processo demorou 1 segundo. A avaliação do modelo realizada com o conjunto de teste descrito anteriormente, obteve uma acuidade de 72,96%. Este valor é relativamente baixo devido as alterações realizadas no conjunto de teste. Estas alterações criaram novas tendências nos dados de teste que o modelo não reconhece devido ao facto não existirem os mesmos padrões no conjunto de treino. Torna-se necessário realizar a atualização do modelo.

A atualização em modo batch cria um novo modelo a partir do conjunto de treino mais os 231 novos dados, o conjunto de treino fica assim formado por 5399 registos. O tempo de aprendizagem foi de 2 segundos e a previsão do conjunto de teste obteve uma acuidade de 81,96%. A atualização em modo incremental reutiliza um modelo já existente para a criação de um novo. Neste caso, é necessário especificar o modelo que será o ponto de partida da aprendizagem incremental. A aprendizagem incremental utiliza os vetores de suporte e o respetivo peso de cada um juntamente com os novos dados para realizar a atualização do modelo. Este modelo incremental obteve resultados idênticos ao modelo batch tendo demorado 2 segundos a ser aprendido e obtendo uma acuidade de 81,96%.

Em ambas as vertentes de atualização dos modelos de SVM, a assertividade foi maior do que na aprendizagem do modelo sem atualização. Isso deve-se ao facto do modelo em primeira instancia não se encontrar apto, isto é, não possuir conhecimento suficiente para prever os valores dos novos dados devido a inserção de novas tendências e padrões nos dados de teste. A atualização permitiu que os modelos assimilassem estas novas caraterísticas dos dados. Isso reflete-se na melhoria da assertividade nas previsões realizadas com os modelos SVM atualizados em comparação com o modelo SVM inicial. De seguida pretende-se realizar os mesmos testes porém com um maior volume de dados para se obter resultados mais concisos

sobre os possíveis benefícios da atualização dos modelos, nomeadamente da atualização incremental.

	Nº de registos	Tempo de execução	Acuidade
Conjunto de treino	103350	00:02:07	75,34%
Conjunto de treino +novos dados (batch)	103350+ 5179 =108529	00:02:09	82,40%
Conjunto de treino +novos dados (incremental)	103350+ 5179 =108529	00:01:30	82,18%
Conjunto de treino + novos dados (1) + novos dados (2) (batch)	103350+ 5179+ 529 = 109058	00:02:11	82,40%
Conjunto de treino + novos dados (1) + novos dados (2) (incremental)	103350+ 5179+ 529 = 109058	00:00:01	82,18%
Conjunto de teste	**898**		

Tabela 20- Resultados da comparação dos modelos produzidos com atualização batch e incremental no TinySVM - conjunto de treino com 103350 registos.

Com um conjunto de treino formado por 103350 registos, a aprendizagem do modelo de previsão SVM demorou 2:07min com uma acuidade de 75,34%. Com a realização da atualização com o método batch, o conjunto de treino fica com um volume maior o que resulta num maior tempo de execução de 2:09min. No entanto, a atualização do modelo permitiu a aprendizagem dos novos padrões e tendências dos novos dados e a acuidade aumentou para 82,40%.

Com o processo de atualização incremental, reutilizando o modelo inicialmente treinado com 103350 registos, o tempo de execução diminuiu para 1:30min e a acuidade foi de 82,18% que é ligeiramente inferior a acuidade obtida pelo modelo atualizado com o método *batch*. Um segundo conjunto de novos dados foi adicionado com 529 registos para testar a diferença no tempo de aprendizagem. A aprendizagem batch do modelo de previsão terminou ao fim de 2:11min e manteve uma acuidade de 82,40%. A aprendizagem incremental demorou 1 segundo e manteve a acuidade nos 82,18%. A assertividade dos modelos na primeira adição de novos dados e na segunda obtiveram a mesma acuidade para cada método de atualização. Isso deve-se ao segundo conjunto de novos dados de treino possuir apenas 529 registos sem novas tendências o que é pouco significativo num conjunto de treino com um volume tão elevado. No entanto, a atualização *batch* demorou 2:11min para realizar um reprocessamento do conjunto total de treino, com 109058 registos. Este processo é longo, dispendioso a nível de capacidade de processamento do sistema, e é deveras dispensável pois criou um novo modelo de previsão com exatamente a mesma capacidade preditiva que o modelo criado anteriormente com o mesmo método de atualização.

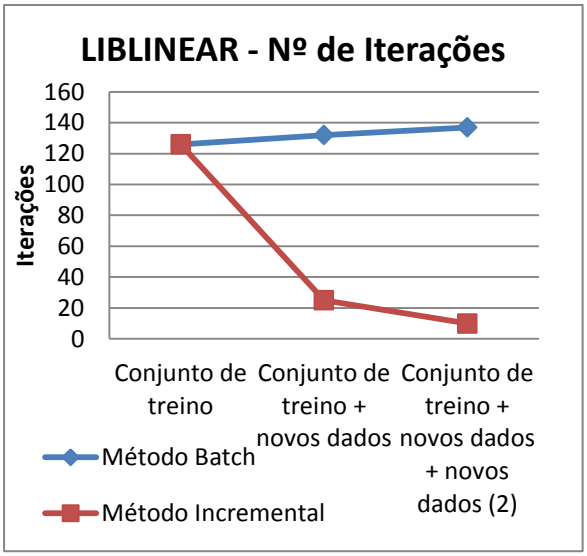

Figura 21 - Gráfico comparativo do número de iterações necessárias para a aprendizagem do modelo para cada método de atualização, batch e incremental. Valores extraídos dos testes realizados com o LIBLINEAR.

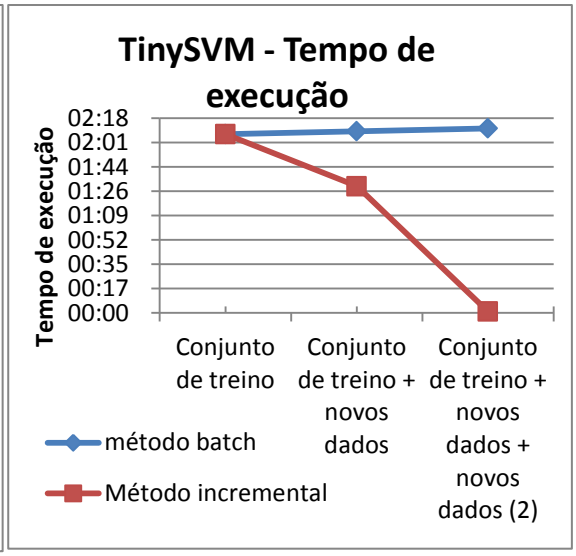

Figura 22 - Gráfico comparativo do tempo de execução da aprendizagem do modelo para cada método de atualização, batch e incremental. Valores extraídos dos testes realizados com o TinySVM.

Os gráficos acima ilustram claramente a diferença no custo computacional exigido na atualização dos modelos de previsão. O método batch produz um modelo independente de outros modelos, logo tem que considerar todo o conjunto de dados no processo de aprendizagem. A medida que surgem novos dados o conjunto total vai aumentando de volume e a exigência e complexidade do processo de atualização aumenta da mesma forma. O método incremental produz um modelo a partir de outro modelo já existente. Isso permite incorporar as tendências e os padrões conhecidos dos vetores de suporte de outros modelos ao conjunto de novos dados utilizados na aprendizagem de um novo modelo.

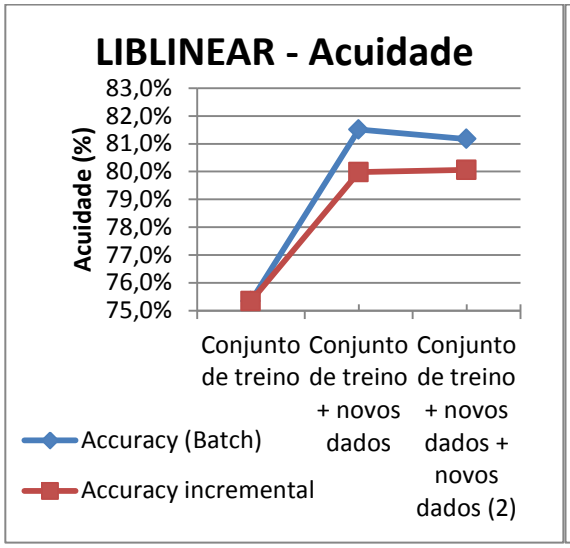

Figura 23 - Gráfico comparativo da acuidade dos modelos batch e incremental. Valores extraídos dos testes realizados com o LIBLINEAR.

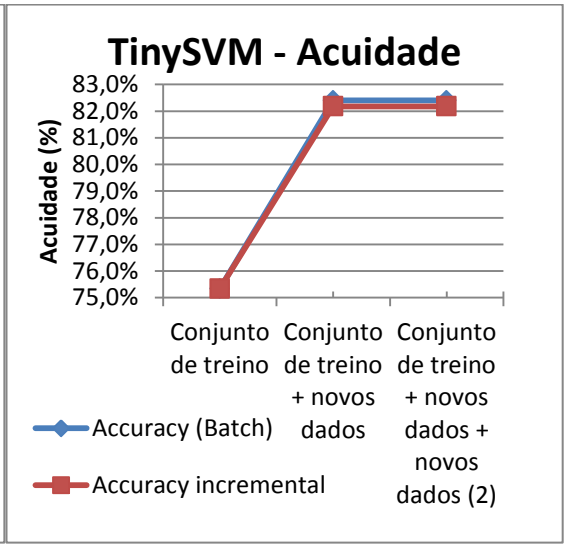

Figura 24 - Gráfico comparativo da acuidade dos modelos batch e incremental. Valores extraídos dos testes realizados com o TinySVM.

Nos gráficos das Figuras 23 e 24 pode-se verificar a diferença na acuidade dos modelos produzidos a partir dos métodos de atualização *batch* e incremental em cada uma das ferramentas utilizadas. Os modelos com atualização *batch* possuem uma maior assertividade nas previsões realizadas. Esta diferença é mais notória nos modelos produzidos no LIBLINEAR. Porém, pode-se verificar uma descida da acuidade na segunda atualização no modelo *batch* e uma ligeira subida no modelo incremental.

Na ferramenta TinySVM, a diferença de acuidade entre o modelo de previsão com atualização batch e o modelo incremental é inferior a 0,5% o que é pouco significativo. A segunda atualização do modelo de previsão não alterou a acuidade que se manteve constante. Isso deve-se, como já foi mencionado, ao segundo conjunto de novos dados ser formado por 529 novos registos com as mesmas caraterísticas dos conjuntos de treino utilizados na aprendizagem dos modelos anteriores. Estes novos dados não trazem novas tendências ou padrões uteis para a previsão dos valores do conjunto de teste.

Pode-se então apurar que em ambas as ferramentas utilizadas o processo incremental é computacionalmente bastante mais rápido, porém obteve na maioria dos testes realizados uma acuidade ligeiramente inferior ao método batch de atualização dos modelos. Em alguns testes realizados com o TinySVM a acuidade dos modelos atualizados com o método batch foi idêntica aos modelos atualizados incrementalmente. Pode-se ainda verificar nos testes anteriores que com o aumento do conjunto de dados, o tempo de execução da aprendizagem dos modelos aumentou consideravelmente. Porém, enquanto nos modelos incrementais o tempo de execução foi diminuindo ao longo das várias execuções, nos modelos com atualização batch o tempo de execução foi aumentando pois foi considerando o conjunto total de dados, novos e antigos.

O reprocessamento da aprendizagem de um modelo realizado pelo método batch é pouco recomendável para conjuntos de dados bastante volumosos onde pode facilmente demorar várias horas ocupando toda a capacidade de processamento de um sistema. Este fator inviabiliza um pouco o método batch para um sistema de controlo e avaliação da qualidade dos efluentes das ETAR, que necessitam de soluções rápidas para os diversos problemas que podem surgir.

Capítulo 9

Conclusões e Trabalhos Futuros

9.1. Conclusões

Este projeto foi desenvolvido com o objetivo de demonstrar os benefícios das técnicas de *Data Mining* no funcionamento de infraestruturas como as ETAR. O processo de tratamento de águas residuais realizado nas ETAR permite manter o ambiente circundante aos meios hídricos mais seguro e agradável e assim conservar a saúde pública. A utilização de técnicas de *Data Mining* para a previsão dos parâmetros físico-químicos, biológicos e microbiológicos permite assegurar altos índices de qualidade do processo de tratamento dos efluentes e na sua monitorização e avaliação. A técnica de *Data Mining* utilizada neste projeto foi a técnica SVM com algumas das suas variantes. Esta técnica, relativamente recente, tem mostrado resultados surpreendentes, de tal modo que tem chamado a atenção de muitos especialistas da área. Um classificador SVM é baseado no princípio de margem máxima (Yu, 1997). Outra propriedade do classificador SVM é o seu bom poder de generalização devido a sua representação dispersa da função objetivo ou de decisão. A técnica de SVM pode ser utilizada para a criação de modelos de previsão para classificação, que prevê a classe dos exemplos de previsão, e de regressão, que faz a previsão do valor com uma determinada precisão (Smola e Scholkopf, 2003).

Depois de identificada a técnica de *Data Mining* utilizada, podemos realçar várias das tarefas realizadas. A necessidade de proceder à análise e preparação dos dados de modo a fornecer dados aos algoritmos SVM adaptados às tarefas preditivas posteriormente realizadas. Neste processo de preparação dos dados removeu-se alguns dados prejudiciais para a aprendizagem dos modelos de previsão como os valores nulos. Também existiu a preocupação de tornar os conjuntos de dados mais eficientes procedendo a redução da dimensionalidade de dados. Este processo permitiu selecionar um conjunto com os valores mais relevantes para as tarefas preditivas descartando os atributos com pouca relevância para a previsão da variável de interesse (Pyle, 1999)

Outra preocupação neste projeto foi a necessidade de atualizar os modelos de previsão quando surgiam dados novos com tendências e padrões desconhecidos pelo modelo de previsão. Foram utilizadas então duas ferramentas de *Data Mining* - o LIBLINEAR e o TinySVM - que permitiram realizar a aprendizagem incremental dos modelos SVM. Por fim e para se poder obter índices de desempenho dos modelos produzidos realizou-se a avaliação dos modelos de previsão. As medidas de desempenho utilizadas permitiram comparar os modelos e tirar diversas conclusões.

Os Conjuntos de Dados

Neste projeto foram utilizados dois conjuntos de dados distintos referentes ao processo de tratamento das águas residuais realizado nas ETAR. O primeiro conjunto de dados possuía inicialmente 53 registos o que não era suficiente para realizar as tarefas de previsão. Como tal, tornou-se necessário aumentar o volume de dados. A forma utilizada para aumentar o número de registos do conjunto de dados foi através da geração automática de valores para cada atributo. Este método realizado no Excel através do operador "AleatorioEntre" gera automaticamente valores compreendidos entre um mínimo e um máximo definido pelo utilizador. Este processo permitiu aumentar o conjunto de dados sem ocorrência de valores nulos nem valores irregulares como os *outliers*. Porém, a geração automática de cada atributo isoladamente revelou-se um método pouco recomendável devido a perda de correlação entre os atributos.

Como os valores foram aleatoriamente gerados, as tendências intrínsecas dos dados, que são difíceis de identificar apenas por observação, deixam de existir. Isso resultou num baixo coeficiente de correlação dos atributos com a variável de previsão CBO. Comparativamente, no segundo conjunto de dados utilizado, que era formado por 527 registos, também se aumentou a quantidade de dados. Porém, a forma utilizada para tal foi a cópia dos registos até se obter um conjunto de dados com registos na ordem dos 5000 exemplos.

De seguida procederam-se a algumas alterações, como por exemplo para evitar valores duplicados em atributos tais como a data que é o identificador único de cada registo. Para este conjunto, a matriz de correlação criada apresentou coeficientes de correlação entre os atributos bastante mais elevados. De notar ainda que essa diferença entre os conjuntos de dados é ainda mais evidente no processo de seleção de atributos, no qual se produziu modelos de previsão para se testar o desempenho dos métodos de seleção de atributos.

Os resultados obtidos pelos modelos de regressão e de classificação produzidos através da aprendizagem do segundo conjunto de dados foram significativamente superiores aos resultados obtidos pelos modelos produzidos com o conjunto com dados gerados aleatoriamente. Como tal, os testes realizados a partir desse ponto foram todos realizados com o segundo conjunto de dados. De referir ainda que em ambos os conjuntos de dados, o método de seleção de atributos, a partir do qual se produziu o modelo com melhor desempenho, foi o método de *Wrapper* chamado *Forward Selection*. Porém, no segundo conjunto de dados, o subconjunto resultante da seleção de atributos com o método *Forward Selection* era apenas constituído por 3 atributos o que representa uma dimensionalidade de dados bastante baixa. No entanto, a dimensionalidade dos dados não deve ser nem alta nem baixa. Deste modo, e como o desempenho dos modelos produzidos a partir do método de seleção de atributos *Weight by*

correlation obtiveram resultados próximos do método *Forward Selection,* optando-se por escolher o subconjunto com 17 atributos selecionado pelo método *Weight by correlation*.

Comparação dos Modelos de Classificação e de Regressão

A utilização da ferramenta RapidMiner, para além de possibilitar a realização do processo de preparação de dados, permitiu realizar a comparação de modelos de previsão para classificação e para regressão. Os algoritmos envolvidos nesta etapa foram o *e-SVR* e o *C-SVC*, que são algoritmos SVM, respetivamente, de regressão e de classificação. Os modelos de previsão criados pretendem prever os valores da variável de interesse CBO (Carência Bioquímica em Oxigénio). Este parâmetro de medição dos efluentes das ETAR é o ideal para a ser alvo de previsão devido a sua grande importância no processo de tratamento, nomeadamente aeróbio, e devido ao longo período de espera de 5 dias para se obter os resultados das medições. Para se realizar esta comparação foi necessário discretizar a variável de interesse para dividir os exemplos em duas classes de valores, -1 e 1. A classe -1 engloba os valores pertencentes ao intervalo $[-\infty\,;25]$, enquanto que a classe 1 é atribuída aos valores de $[25;+\infty]$.

	e-SVR				
	Parâmetros	**Vetores de suporte**	**RMSE**	**Correlação**	**Erro absoluto**
CBO - Sem otimização de parâmetros	*γ = 0.0; C = 0.0; e = 0.001; p = 0.1*	4194	15.614	0.400	3.571
CBO - Com otimização de parâmetros	γ = 0.0; C = 2.0; e = 0.1; p = 0.734	2286	12.120	0.702	2.312

Tabela 21 - Comparação dos resultados dos modelos de regressão com e sem otimização de parâmetros de configuração.

	C-SVC						
	Parâmetros	**Vetores de suporte**	**RMSE**	**Correlação**	**Erro Absoluto**	**Acuidade**	**Margem**
CBO - Sem otimização de parâmetros	*C = 0.0; γ = 0.0; e=0.001*	870	0.261	0.936	0.253	88.38%	0.341
CBO - Com otimização de parâmetros	*C = 0.0; γ = 0.667; e = 0.1*	941	0.321	0.997	0.317	95.41%	0.635

Tabela 22 - Comparação dos resultados dos modelos de classificação, com e sem otimização de parâmetros de configuração.

A divisão é feita nos 25 mg/l pois é o valor máximo aceitável estipulado nas normas legais europeias. Para a avaliação dos modelos de previsão e comparação dos modelos, utilizaram-se as medidas de desempenho RMSE, correlação, e o erro médio absoluto. Apresentaram-se ainda os valores da acuidade e da margem dos modelos de previsão para classificação. Para uma melhor análise desta comparação, realizou-se a otimização dos parâmetros de configuração dos modelos de regressão e de classificação. Os resultados obtidos, com e sem otimização de

parâmetros de configuração dos modelos, demonstraram uma melhor assertividade dos modelos de classificação em comparação com os modelos de regressão. Os modelos produzidos com otimização de parâmetros obtiveram resultados bastante superiores aos modelos sem otimização como se pode ver nas tabelas seguintes.

Comparação dos Métodos de Atualização dos Modelos de Previsão

A principal caraterística que levou a utilização das ferramentas LIBLINEAR e TinySVM foi a possibilidade de proceder a atualização incremental de modelos de previsão quando surgem novos dados. Estes novos dados podem possuir tendências ou padrões diferentes dos que foram aprendidos pelo modelo em aprendizagens anteriores. Isto resulta numa baixa precisão dos valores de previsão devido ao modelo desconhecer as tendências dos novos dados. Tornou-se necessário ensinar o modelo a prever os valores dos novos dados. Este processo de atualização de um modelo pode ser realizado através de dois métodos, batch ou incremental, e realiza uma nova aprendizagem do modelo para assimilar as tendências intrínsecas dos novos dados. Enquanto o método batch realiza a aprendizagem de um novo modelo a partir do conjunto total de dados e independentemente de qualquer outro modelo já criado, o método incremental utiliza um modelo e os seus conhecimentos sobre os dados como condição de partida e realiza a aprendizagem dos padrões e tendências ainda não adquiridas.

Os resultados obtidos e apresentados nas figuras 21,22,23 e 24, no capítulo 8 demonstram que em bases de dados com enormes volumes de dados, como é o caso das ETAR, é vantajoso utilizar o método incremental de atualização dos modelos devido a sua boa eficácia na previsão, que foi neste caso inferior ao método batch apenas entre 0,2% e 1,5%, e sobretudo a sua rapidez de assimilação. A grande vantagem dos métodos incrementais é a rapidez de aprendizagem dos modelos pois é significativamente mais rápida. O custo computacional é muito mais baixo, permite não utilizar toda a capacidade de processamento dos sistemas. Permite realizar a atualização do modelo de previsão sem interferir com outras tarefas de monitorização do processo de tratamento das águas residuais. O método batch é aconselhado em conjuntos de dados com volumes mais baixos pois o seu custo de processamento é elevado e pode interferir ou até mesmo interromper outras tarefas dos sistemas de controlo e avaliação dos processos de tratamento.

Pode-se então concluir que as técnicas de *Data Mining* podem trazer diversos benefícios e resolver vários problemas identificados nas ETAR, como por exemplo avarias nos sensores que acabam por realizar leituras irregulares e criar dados conhecidos como outliers ou valores nulos. Com a análise realizada aos atributos mais pertinentes e mais influentes na previsão de CBO e com a fase de análise e preparação de dados identificaram-se diversas caraterísticas dos constituintes dos efluentes que podem ser de potencial interesse. Nas Estações de Tratamento de Águas Residuais a utilização deste tipo de técnicas para a previsão da qualidade dos seus efluentes representaria uma melhoria significativa, tanto no processo de tratamento, como no processo de avaliação dos riscos de contaminação das águas. Este tipo de infraestruturas possui diariamente um elevado volume de recolha de dados obtidos através dos diferentes sensores de medição. A possibilidade de prever o grau de contaminação da água e escolher o melhor tratamento a ser utilizado para aqueles valores, assim como, a possibilidade de substituir parcialmente alguns sensores defeituosos permite manter o bom funcionamento de toda a

infraestrutura. Estas são algumas das várias possibilidades das técnicas de *Data Mining* no contexto da monitorização da qualidade dos efluentes de uma ETAR.

9.2. Trabalhos Futuros

Esta dissertação teve como tema base a previsão em tempo real da qualidade dos efluentes de uma ETAR. Vários estudos e experiências podem ser desenvolvidos em torno deste contexto. Nomeadamente, a utilização de técnicas de *Data Mining* para o aperfeiçoamento do processo de tratamento das águas residuais. A criação de modelos de previsão para outros parâmetros da qualidade das águas residuais em diferentes etapas do processo de tratamento permitiria de certeza obter mais informações uteis.

Posteriormente, uma possível aplicação das técnicas de SVM incremental utilizadas neste projeto poderia ser realizada com um conjunto de dados real, proveniente de uma ETAR. No caso deste projeto, em concreto, o volume de dados foi pouco significativo, daí a necessidade de enriquecer o conjunto de dados. Este enriquecimento aumentou a quantidade de registos do conjunto de dados mas não a qualidade, uma vez que não aumentou a quantidade de padrões que relacionam as diversas caraterísticas dos dados.

Para além do conjunto de dados, existem várias formas de melhorar os modelos de previsão produzidos através de algoritmos SVM. Outra possível aplicação destas técnicas seria a criação de uma ferramenta mais especializada para o contexto das ETAR e a sua implementação num processo de tratamento de uma ETAR. A utilização de conjuntos de dados mais eficientes e uma ferramenta que permita a produção de modelos de previsão para regressão com atualização incremental seria o ideal para o contexto das ETAR em estudo. Os modelos de regressão permitem prever o valor aproximado, com um determinado desvio, da variável de previsão.

A variante incremental dos modelos SVM é ainda um método relativamente recente e em desenvolvimento. Apesar da grande procura por parte dos utilizadores desta ferramenta ainda não é possível realizar este processo no RapidMiner. Neste projeto não se utilizou técnicas de regressão na ferramenta LIBLINEAR devido a sua incompatibilidade com a atualização incremental dos modelos de previsão. Relativamente a ferramenta TinySVM os modelos de regressão produzidos obtiveram resultados com um grau de assertividade bastante insatisfatório.

Futuramente e com a evolução de ferramentas de *Data Mining* a implementação e a utilização de técnicas SVM incremental irão de certeza chamar a atenção de muitos especialistas da área. Relativamente ao contexto das ETAR, a recolha e análise de requisitos de negócio propostos por decisores de uma ETAR seria um ponto de partida para se conhecer mais projetos que futuramente poderiam ser desenvolvidos.

Bibliografia

Agência Portuguesa do Ambiente,2012. Programa Nacional para o Uso Eficiente da Água

ALES, V.T., 2008. O algoritmo sequential minimal optimisation para resolução do problema de support vector machine: uma técnica para reconhecimento de padrões.

Amo, S., n.d. Idéia geral do algoritmo FUP. pp.1–8.

Araújo, A., Máquinas de Vetores de Suporte Support Vector Machine.

Ayad, A., 2000. A New Algorithm for Incremental Mining of Constrained Association Rules. , 1.

Baranauskas, J.A., 2006. Métodos de Amostragem e Avaliação de Algoritmos.

Barroso, A., 2012. Avaliação do desempenho de uma ETAR de lamas ativadas através do estudo das comunidades microbiológicas do licor misto.

Berkhin, P., 2002 Survey of Clustering *Data Mining* Techniques. pp.1–56.

Boswell, D., 2002. Introduction to Support Vector Machines. pp.1–15.

Burges, C.J.C., 1997. A Tutorial on Support Vector Machines for Pattern Recognition. , 43, pp.1–43.

Chang C.C. & Lin C.-J. 2011 LIBSVM: a library for support vector machines.

Camilo, C.O. & Silva, J.C., 2009. Mineração de Dados : Conceitos, Tarefas, Métodos e Ferramentas.

Cauwenberghs, G. & Poggio,T. 2003. Incremental and Decremental Support Vector Machine Learning.

Cavalcanti, F.T., 2005. Incremental Mining Techniques.

Chung, S. & Mcleod, D., 2005. Dynamic Pattern Mining : An Incremental Data Clustering Approach.

Chung, S., Jun, J. & Mcleod, D., 2005. Incremental Mining from News Streams.

Côrtes, S.D.C., Porcaro, R.M. & Lifschitz, S., 2002. Mineração de Dados – Funcionalidades, Técnicas e Abordagens. *PUC-Rio Informática*, p.35. Available at: ftp://ftp.inf.puc-rio.br/pub/docs/techreports/02_10_cortes.pdf.

Cortez, P. 2013. Classificação e Regressão - Parte 1: Aprendizagem Supervisionada, Processamento de dados e Validação de Modelos, Departamento de Sistemas de Informação. Universidade do Minho - Campus de Azurém Guimarães;

Cruz, P.N., 2009. Tratamento de Águas Residuais Estação de Tratamento de Palmarejo.

Dias, I., 2007. ETAR de Lordelo - Aves, Projeto de execução. Pp –11.

Diehl, C.P. & Cauwenberghs, G., 2003 Svm incremental learning, adaptation and optimization. *Proceedings of the International Joint Conference on Neural Networks, 2003.*, 4, pp.2685–2690. Disponível em: http://www.csie.ntu.edu.tw/~cjlin/libsvm/. Acedido em: 06/2014.

Do, H., Pham, N. & Do, T., 2005. A SIMPLE, FAST SUPPORT VECTOR MACHINE ALGORITHM FOR *DATA MINING*. pp.1–9.

Domeniconi, C. & Gunopulos, D., 2001. Incremental Support Vector Machine Construction. *Techniques in vascular and interventional radiology*, 17(3), p.139. Available at: http://www.ncbi.nlm.nih.gov/pubmed/25247973.

Ester, M. et al., 1998 Incremental Clustering for Mining in a Data Warehousing Environment.

Fayyad, U.M., Piatetsky-Shapiro, G. & Smyth, P., 1996b. From Data Mining to Knowledge Discovery in Databases.

Fernandes, R. et al., 2009. O algoritmo Support Vector Machines (SVM): avaliação da separação ótima de classes em imagens CCD-CBERS-2. pp.2079–2086.

Fung, G. & Mangasarian, O.L., 2001. Incremental Support Vector Machine Classification. pp.247–260.

Gallop, J.R. et al., 2004. The Use of Data Mining for the Monitoring and Control of Anaerobic Wastewater Plants. Em: 4th International Workshop on Environmental Applications of Machine Learning (EAML), 2004.

Gil, A., 2011. O Planeamento de Recursos Hídricos no atual contexto de incerteza : objetivos e metodologias.

Gordon, G., 2004. Support Vector Machines and Kernel Methods.

Han, F. et al., 2012. A New Incremental Support Vector Machine Algorithm. 10(6).

Han, J., Pei. J, & Kamber, M., 2006. *Data Mining Concepts and Techniques* Second Edi.,

Hand, D., Mannila, H. & Smyth, P., 2001. *Principles of Data Mining*.

Hsu, C.-w., Chang, C.-c. & Lin, C.-j., 2010. A Practical Guide to Support Vector Classification.

Ipp, C.I., Azevedo, A. & Santos, M.F., 2008. Kdd, semma and crisp-dm: a parallel overview. , pp.182–185.

Joachims, T., 1998. Making Large-Scale SVM Learning Pratical.

Kinto, E.A., 2011. Otimização e análise das máquinas de vetores de suporte aplicadas à classificação de documentos.

Laskov, P., Geh, C. & Kruger, S., 2006. Incremental Support Vector Learning : Analysis, Implementation and Applications. , 7, pp.1909–1936.

Lazzarotto, l.l., Oliveira, A.D.P.. & Lazzarotto, J.J.., 2006. Aspectos teóricos do *Data Mining* e aplicação das redes neurais em previsões de preços agropecuários.

Leaper, N., n.d. A visual guide to CRISP-DM methodology. p.1.

Leite, S.C. & Neto, R.F., 2007. Algoritmo de Margem Incremental para Classificadores de Larga Margem. 2, p.2001.

Lin, C., 2005. Optimization, Support Vector Machines, and Machine Learning.

Lin, C., 2006. Support Vector Machines.

Lopes, P.,2009. Água no séc. XXI: desafios e oportunidades

Lorena, A.C., Carvalho, A., 2006. Uma Introdução às Support Vector Machines. RITA, XIV(2), pp.43-67.

Luizi,R,. 2012. Operação de Sistemas de Tratamento de Águas Residuais por Lamas Ativadas com Arejamento Prolongado.

Mangasarian, O.L., 2003. *Data Mining* via Support Vector Machines. pp.1–25.

Martin, M., 2002 On-line Support Vector Machine Regression.

Meireles, M., 2011. Otimização da Estação de Tratamento de Águas Residuais de Crestuma.

Mendes, A.B., 2007. Metodologias de *Data Mining*.

Ministério do ambiente, 1997. Decreto-Lei nº 152/97 de 19 de Junho. pp.2959–2967.

Ministério do Ambiente, 1998. Decreto-Lei nº 236/98 de 1 de Agosto.

Ministério do Ambiente, 2004. Avaliação do desempenho ambiental das estações de tratamento de águas residuais urbanas em Portugal continental. pp.1–22.

Ministério do Ambiente, 2007. Decreto-Lei n^{o} 306/2007 de 27 de Agosto.

Müller, R., Mika, S., Ratsch, G., Tsuda, K., and Scholkopf, B. 2001. An introduction to kernel-based learning algorithms. IEEE Transactions on Neural Networks

Nath, B., Bhattacharyya, D.K. & Ghosh, A., 2013. Incremental association rule mining : a survey. , 3(June), pp.157–169.

Nikitidis, S., Nikolaidis, N. & Pitas, I., 2012. Multiplicative update rules for incremental training of multiclass support vector machines. *Pattern Recognition*, 45(5), pp.1838–1852. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0031320311004547.

Pinho, R., 2009 Espaço incremental para a mineração visual de conjuntos dinâmicos de documentos.

Platt, J.C., 1999. Using Analytic QP and Sparseness to Speed Training of Support Vector Machines. Optimization, 11, pp.1-8.

Prati, R.C., Monard, M.C. & Batista, A., 2008 Curvas ROC para avaliação de classificadores. pp.1–8.

Presidente da República, 2011. Decreto Presidencial nº 261/11 de 6 de outubro.

Pyle, D., 1999. *Data Preparation for Data Mining*,

Ribeiro, D., 2012. Support Vector Machines na Previsão do Comportamento de uma ETAR

Rodrigues, C. et al., 2012 Implementação de um sistema de alerta em tempo real da qualidade da massa de água recetora de águas urbanas.

Sabino, V.C., 2006. Categorização de Textos Usando Máquinas de Suporte Vetorial.

Santos, E., dos, 2002. Teoria e Aplicação de Support Vector Machines à Aprendizagem e Reconhecimento de Objetos Baseado na Aparência.

Shah, S., Chauhan, N.C. & Bhanderi, S.D., 2012. Incremental Mining of Association Rules : A Survey. , 3(3), pp.4071–4074.

Shilton, A. et al., 2005. Incremental Training of Support Vector Machines.

Smola, A.J. & Schölkopf, B., 2004. A Tutorial on Support Vector Regression.Sperling, M.v., 2007. Wastewater characteristics, treatment and disposal. London: IWA Publishing.

Stefan, R., n.d. Incremental Learning with Support Vector Machines.

Thomé, A., n.d. Redes neurais - uma ferramenta para kdd e *Data Mining*.

Tsai, C., Lin, C. & Lin, C., n.d. Incremental and Decremental Training for Linear Classification.

Vapnik, V.N., 1979. Support Vector Machines.

Vapnik, V.N., 1995. The Nature of Statistical Learning Theory. 2nd ed. New York: Springer-Verlag.

Vapnik, V.N., 1999. An overview of statistical learning theory. IEEE Transactions on Neural Networks, 10(5), pp.988-999.

Veloso, A. et al., n.d. Parallel, Incremental and Interactive Mining for Frequent Itemsets in Evolving Databases.

Wang, L.-j. & Chen, C.-b., 2008. Support Vector Machine Applying in the Prediction of Effluent Quality of Sewage Treatment Plant with Cyclic Activated Sludge System Process.

Wen, Y. & Lu, B., n.d. Incremental Learning of Support Vector Machines by Classifier Combining.

Yu, H., 1997. *Data Mining* via Support Vector Machine.

Zembrzuski, M.C., 2010. Classificadores Bayesianos.

Anexos

Anexo 1 – Conjuntos de dados

1º Conjunto de dados

Tipo de atributo	Nome Atributo	Tipo de dados	Estatisticas	Intervalo de valores	Valores nulos
label	SP6_BOD	real	avg = 51.420 +/- 30.944	[1.000; 110.000]	0
regular	Season	real	avg = 1.491 +/- 1.119	[0.000; 3.000]	0
regular	Date	polynom	mode = 01-09-1995 (1), least = 01-09-1995 (1)	[01-09-1995; 29-04-2014]	0
Regular	SP1_Redox	real	avg = -172.072 +/- 54.611	[-309.000; 555.000]	0
regular	SP1_pH	polynom	mode = 7,5 (120), least = 8,07 (1)	[7.1; 8.2]	0
regular	SP1_COD	real	avg = 989.929 +/- 346.473	[400.000; 1600.000]	0
regular	SP1_BOD	real	avg = 420.205 +/- 200.561	[75.000; 11111.000]	0
regular	SP1_TSS	real	avg = 462.686 +/- 134.229	[0.000; 1000.000]	0
regular	SP1_VSS	real	avg = 278.643 +/- 123.955	[45.000; 750.000]	0
regular	SP2_pH	polynom	mode = 7,32 (102), least = 6,6 (1)	[6.6; 8.08]	0
regular	SP2_DO2(AZ)	polynom	mode = 0,87 (81), least = 0 (33)	[0; 1]	0
regular	SP2_DO2(AxZ)	polynom	mode = 0,43 (81), least = -0,3 (1)	[-0.01; 0.9]	0
regular	SP2_Redox(AZ)	real	avg = -135.540 +/- 65.674	[-337.000; 565.000]	0
regular	SP2_TSS	real	avg = 2435.409 +/- 906.290	[801.000; 4000.000]	0
regular	SP2_VSS	real	avg = 1938.202 +/- 908.084	[300.000; 3500.000]	0
regular	SP3_pH	polynom	mode = 7,19 (88), least = 6,53 (1)	[6.53; 8.03]	0

regular	SP3_DO2	polynom	mode = 1,92 (74), least = 1,01 (1)	[1.01; 2.99]	0
regular	SP3_Redox	real	avg = -57.252 +/- 71.376	[-251.000; 596.000]	0
regular	SP3_V30	real	avg = 263.949 +/- 123.321	[90.000; 600.000]	0
regular	SP3_TSS	real	avg = 2772.818 +/- 923.752	[948.000; 4485.000]	0
regular	SP3_VSS	real	avg = 2490.540 +/- 873.311	[859.000; 4000.000]	0
regular	SP4_Redox	real	avg = -22.581 +/- 97.200	[-250.000; 150.000]	0
regular	SP4_TSS	real	avg = 5026.770 +/- 1025.611	[2531.000; 7382.000]	0
regular	SP4_VSS	real	avg = 3497.945 +/- 1154.359	[344.000; 6523.000]	0
regular	SP4_VSS/TSS%	real	avg = 68.268 +/- 13.620	[12.000; 92.000]	0
regular	SP5_pH	polynom	mode = 7,35 (156), least = 7,48 (22)	[6.9; 7.65]	0
regular	SP5_COD	real	avg = 70.097 +/- 17.493	[40.000; 150.000]	0
regular	SP5_BOD	real	avg = 30.389 +/- 12.604	[5.000; 60.000]	0
regular	SP5_TSS	real	avg = 34.758 +/- 11.643	[15.000; 55.000]	0
regular	SP5_VSS	real	avg = 29.131 +/- 11.608	[10.000; 50.000]	0
regular	SP6_pH	polynom	mode = 7,48 (135), least = 7,92 (67)	[7.35; 7.95]	0
regular	SP6_COD	real	avg = 73.659 +/- 25.712	[22.000; 120.000]	0
regular	SP6_TSS	real	avg = 30.367 +/- 17.437	[1.000; 63.000]	0
regular	SP6_VSS	real	avg = 39.777 +/- 22.689	[0.000; 80.000]	0

2º Conjunto de dados

Tipo de atributo	Nome Atributo	Tipo de dados	Estatisticas	Intervalo de valores	Valores nulos
label	outDBO-S	integer	avg = 20.456 +/- 12.314	[10.000 ; 160.000]	0
regular	outPH-S	real	avg = 7.710 +/- 0.188	[7.000 ; 9.700]	0
regular	inSED-E	real	avg = 4.601 +/- 2.678	[0.400 ; 36.000]	0

regular	inPH-P	real	avg = 7.833 +/- 0.227	[7.300 ; 8.500]	0
regular	inSED-D	real	avg = 0.419 +/- 0.371	[0.000 ; 3.500]	0
regular	outSS-S	integer	avg = 22.474 +/- 16.201	[9.000 ; 238.000]	0
regular	inSS-P	integer	avg = 254.431 +/- 148.329	[104.000 ; 1692.000]	0
regular	Date	integer	avg = 42477.317 +/- 67035.325	[1190.000 ; 301290.000]	0
regular	inZN-E	real	avg = 2.363 +/- 2.710	[0.100 ; 33.500]	0
regular	outSED-S	real	avg = 0.037 +/- 0.193	[0.000 ; 3.500]	0
regular	perfinpRD-DQO-S	real	avg = 67.837 +/- 11.284	[1.400 ; 96.800]	0
regular	inCOND-P	integer	avg = 1498.787 +/- 403.196	[646.000 ; 3170.000]	0
regular	inSED-P	real	avg = 5.045 +/- 3.283	[1.000 ; 46.000]	0
regular	globalRD-DQO-G	real	avg = 77.858 +/- 8.646	[19.200 ; 98.100]	0
regular	inCOND-D	integer	avg = 1493.240 +/- 400.667	[85.000 ; 3690.000]	0
regular	inCOND-E	integer	avg = 1481.124 +/- 395.473	[651.000 ; 3230.000]	0
regular	globalRD-DBO-G	real	avg = 89.042 +/- 6.729	[19.600 ; 97.000]	0
regular	inDBO-D	integer	avg = 122.753 +/- 36.071	[26.000 ; 285.000]	0
regular	inDQO-D	integer	avg = 274.584 +/- 73.591	[80.000 ; 511.000]	0
regular	inDQO-E	integer	avg = 407.413 +/- 119.883	[81.000 ; 941.000]	0
regular	perfinpRD-SS-P	real	avg = 58.468 +/- 12.773	[5.300 ; 96.100]	0
regular	inDBO-E	integer	avg = 189.230 +/- 60.892	[31.000 ; 438.000]	0
regular	outDQO-S	integer	avg = 87.503 +/- 38.022	[20.000 ; 350.000]	0
regular	globalRD-SS-G	real	avg = 88.961 +/- 8.192	[10.300 ; 99.400]	0
regular	perfinpRD-SED-P	real	avg = 90.505 +/- 8.743	[7.700 ; 100.000]	0
regular	inSSV-E	real	avg = 61.364 +/- 12.327	[13.200 ; 85.000]	0
regular	inSSV-D	real	avg = 72.909 +/- 10.357	[20.200 ;	0

				100.000]	
regular	inDBO-P	real	avg = 206.956 +/- 72.107	[32.000 ; 517.000]	0
regular	perfinpRD-DBO-S	real	avg = 83.513 +/- 8.326	[8.200 ; 94.700]	0
regular	outCOND-S	integer	avg = 1497.359 +/- 388.101	[683.000 ; 3950.000]	0
regular	inQ-E	integer	avg = 37308.048 +/- 6535.400	[10050.000 ; 60081.000]	0
regular	inPH-D	real	avg = 7.815 +/- 0.199	[7.100 ; 8.400]	0
regular	outSSV-S	real	avg = 80.119 +/- 9.032	[29.200 ; 100.000]	0
regular	inPH-E	real	avg = 7.814 +/- 0.246	[6.900 ; 8.700]	0
regular	inSSV-P	real	avg = 60.354 +/- 12.316	[7.100 ; 93.500]	0
regular	inSS-D	integer	avg = 94.420 +/- 23.953	[49.000 ; 244.000]	0
regular	perfinpRD-DBO-P	real	avg = 39.101 +/- 13.931	[0.600 ; 79.100]	0
regular	inSS-E	integer	avg = 227.867 +/- 136.752	[98.000 ; 2008.000]	0
regular	globalRD-SED-G	real	avg = 99.081 +/- 4.361	[36.400 ; 100.000]	0

Anexo 2 - Deteção de outliers

2º Conjunto de dados

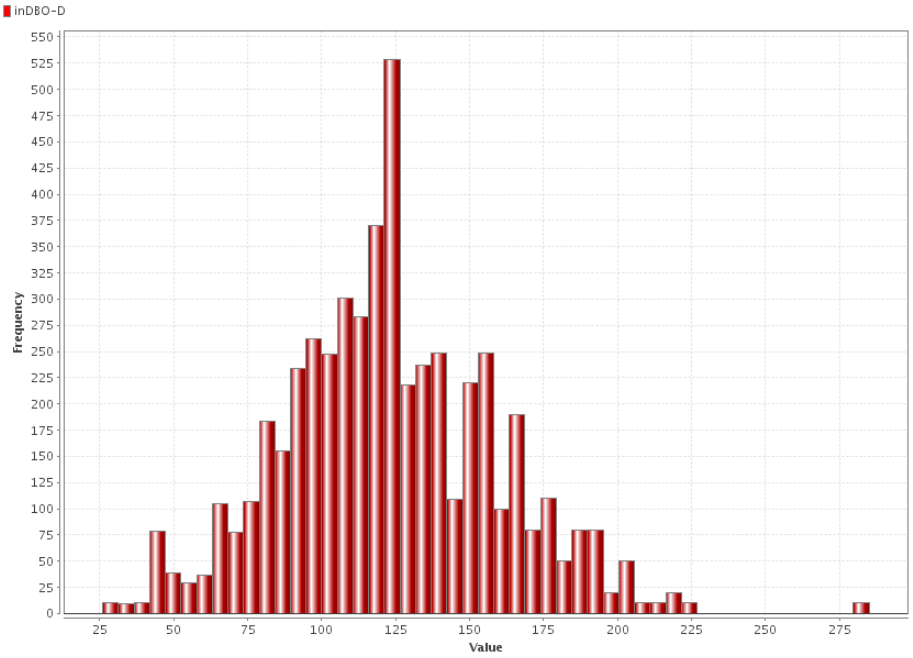

Figura 25 - histograma relativo a existência de outliers no atributo inDBO-D.

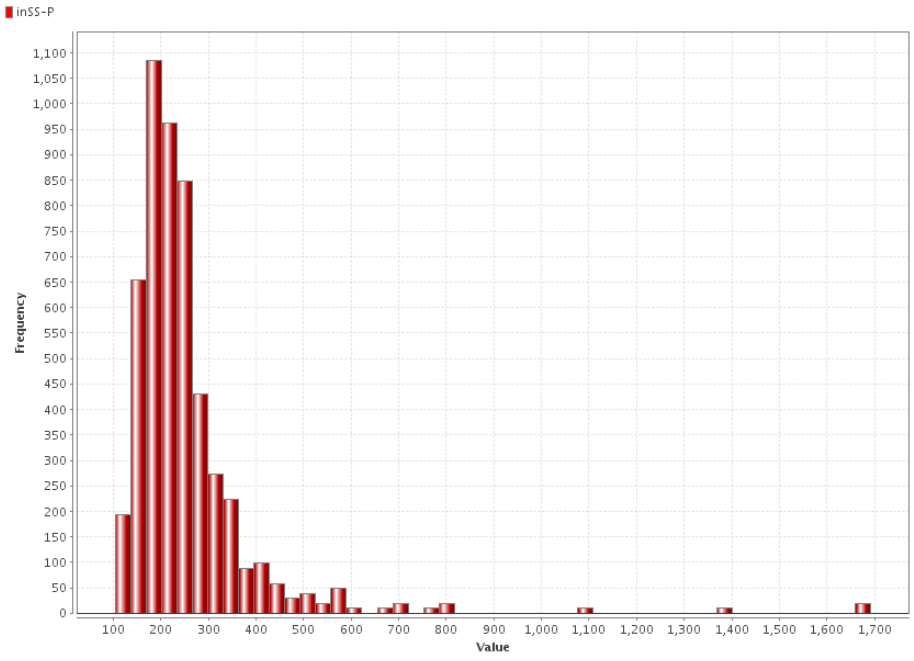

Figura 26 - histograma relativo a existência de outliers no atributo inSS-P.

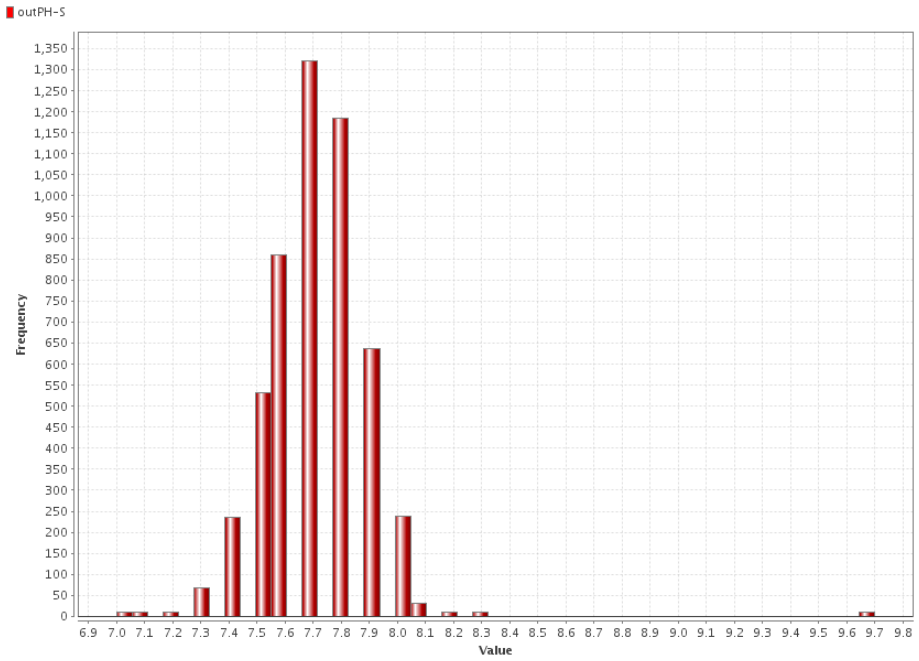

Figura 27 - histograma relativo a existência de outliers no atributo outPH-S.

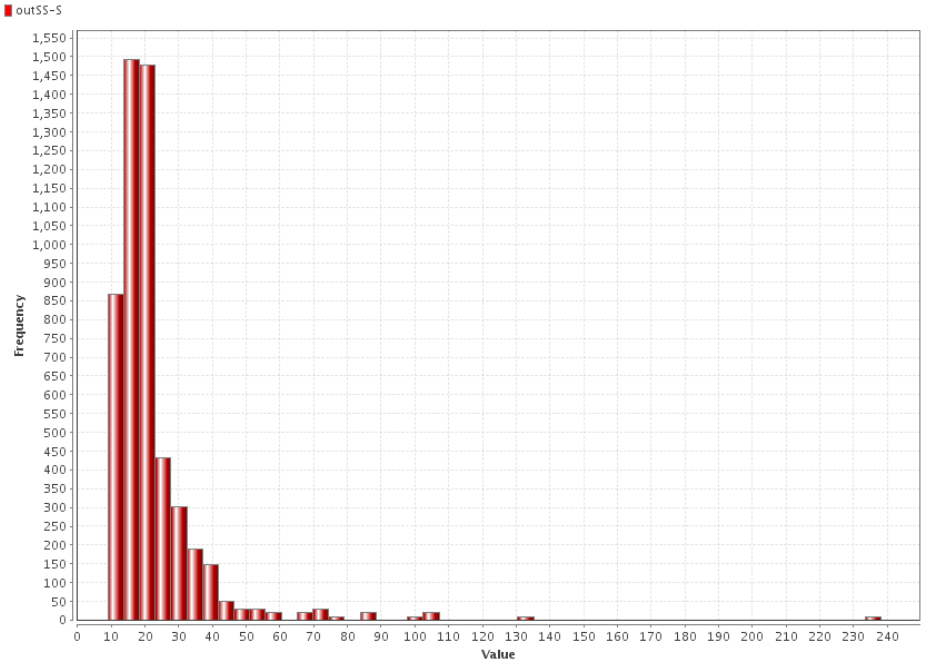

Figura 28 - histograma relativo a existência de outliers no atributo outSS-S.

Anexo 3 - Instruções LIBLINEAR

Train Usage

==========

Usage:

train [options] training_set_file [model_file]

options:

-s type: set type of solver (default 1)

for multi-class classification

0 -- L2-regularized logistic regression (primal)

1 -- L2-regularized L2-loss support vector classification (dual)

2 -- L2-regularized L2-loss support vector classification (primal)

3 -- L2-regularized L1-loss support vector classification (dual)

4 -- support vector classification by Crammer and Singer

5 -- L1-regularized L2-loss support vector classification

6 -- L1-regularized logistic regression

7 -- L2-regularized logistic regression (dual)

for regression

11 -- L2-regularized L2-loss support vector regression (primal)

12 -- L2-regularized L2-loss support vector regression (dual)

13 -- L2-regularized L1-loss support vector regression (dual)

-c cost of constraints violation: set the parameter C (default 1)

-p epsilon: set the epsilon in sensitiveness of loss of support vector regression of epsilon-SVR (default 0.1)

-e epsilon: set tolerance of termination criterion

-s 0 and 2

|f'(w)|_2 <= eps*min(pos,neg)/l*|f'(w0)|_2, where f is the primal function and pos/neg are # of positive/negative data (default 0.01)

-s 11

|f'(w)|_2 <= eps*|f'(w0)|_2 (default 0.001)

-s 1, 3, 4 and 7

Dual maximal violation <= eps; similar to libsvm (default 0.1)

-s 5 and 6

|f'(w)|_inf <= eps*min(pos,neg)/l*|f'(w0)|_inf, where f is the primal function (default 0.01)

-s 12 and 13\n"

|f'(alpha)|_1 <= eps |f'(alpha0)|, where f is the dual function (default 0.1)

-B bias : if bias >= 0, instance x becomes [x; bias]; if < 0, no bias term added (default -1)

-wi weight: weights adjust the parameter C of different classes (see README for details)

-v n: n-fold cross validation mode

-q : quiet mode (no outputs)

Predict Usage
==========
Usage:

predict [options] test_file model_file output_file

options:
-b probability_estimates: whether to output probability estimates, 0 or 1 (default 0); currently for logistic regression only
-q : quiet mode (no outputs)

Note that -b is only needed in the prediction phase. This is different from the setting of LIBSVM.

Examples
========
Train linear SVM with L2-loss function.

> train data_file

Train a logistic regression model.

> train -s 0 data_file

Do five-fold cross-validation using L2-loss svm. Use a smaller stopping tolerance 0.001 than the default 0.1 if you want more accurate solutions.

> train -v 5 -e 0.001 data_file

> train -c 10 -w1 2 -w2 5 -w3 2 four_class_data_file

Train four classifiers:

positive	negative	Cp	Cn
class 1	class 2,3,4.	20	10
class 2	class 1,3,4.	50	10
class 3	class 1,2,4.	20	10
class 4	class 1,2,3.	10	10

> train -c 10 -w3 1 -w2 5 two_class_data_file

If there are only two classes, we train ONE model. The C values for the two classes are 10 and 50.

> predict -b 1 test_file data_file.model output_file

Output probability estimates (for logistic regression only).

Instruções LIBLINEAR Warm-Start

This extension of liblinear supports incremental and decremental learning. If some data are added or removed, it applies a warm-start strategy to efficiently updates the previously trained model. Then you do not need to train the new set from scratch. **Currently, -s 0 (solving primal LR) and -s 2 (solving primal l2-loss SVM) are supported.**

Usage
=====
The usage is the same as liblinear except the following additional option:

-i initial model file: use a previously trained model for incremental/decremental training (only for -s 0 and 2)

Examples
========

> ./train -s 0 heart_scale.sample

> ./train -s 0 -i heart_scale.sample.model heart_scale

Library Function for Incremental and Decremental Learning
==

- Function: model* warm_start_train(const struct problem *prob, const struct parameter *param, const struct model *wsmodel);

This function uses wsmodel for the warm-start training of the given data. It constructs and returns a linear classification or regression model. If wsmodel is NULL, then the regular LIBLINEAR training is conducted.

Anexo 4 - Modelos de previsão

Modelo LIBLINEAR

solver_type L2R_LR
nr_class 2
label 1 -1
nr_feature 17
bias -1
w
0.04514495494728325
0.3922293037591972
0.03822871174685184
-0.096279765539006
-0.1092368501017153
-0.1686695172936437
0.01953562983657866
0.0483107952859587
-0.008995146311635551
-0.1268696294099367
0.2836958855576906
-0.1599238686416402
0.1387220367266134
0.01331640258995657
-0.01381080099132449
-0.03259937275634631
-0.000316141095564821

Modelo TinySVM

TinySVM Version 0.09
3 # kernel type
1 # kernel parameter -d
1 # kernel parameter -g
1 # kernel parameter -s
1 # kernel parameter -r
empty # kernel parameter -u
20 10 50 8.0158216 # number of SVs/BSVs/number of training data/L1 loss
0.03127000900965964 # threshold b
0.9913818628520047 1:7.3 2:21 3:0.02 4:70 5:79.40000000000001 6:89.01364562000001 7:122.3486974 8:280
0.990988538503039 1:7.5 2:17 4:80.8 5:79.5 6:89.01364562000001 7:122.3486974 8:474
0.9910159469910083 1:7.6 2:21 3:0.05 4:52.9 5:75.8 6:89.01364562000001 7:122.3486974 8:272
0.9910159469910083 1:7.6 2:20 4:72.3 5:82.3 6:90.2 7:158 8:376
-0.9688762513920464 1:7.6 2:22 3:0.02 4:71 5:78.2 6:92.09999999999999 7:122.3486974 8:372
0.9911602499266616 1:7.5 2:28 4:78.3 5:73.09999999999999 6:90.09999999999999 7:150 8:460
0.9911602499266616 1:7.5 2:26 3:0.05 4:79.8 5:86.2 6:89.01364562000001 7:192 8:376
-0.9688778543567879 1:7.5 2:18 4:53.7 5:66.90000000000001 6:92.09999999999999 7:181 8:350
-0.9688727204124567 1:7.5 2:19 3:0.03 4:58.2 5:73.8 6:92.2 7:111 8:282
-0.9688732945241766 1:7.6 2:27 3:0.02 4:66.09999999999999 5:69 6:89 7:164 8:463
0.9270191282483583 1:7.6 2:131 3:3.5 4:25.7 5:36 6:19.6 7:172 8:412
0.9096852268048286 1:7.4 2:238 3:2 4:67.81736527 5:77.85657371000001 6:89.01364562000001 7:116 8:276
0.9912308053531928 1:7.5 2:104 3:0.06 4:20.4 5:31.7 6:26.3 7:79 8:216
0.9912314410034954 1:7.6 2:98 4:52.9 5:54.1 6:84.3 7:136 8:325
-0.9688603413039567 1:7.5 2:41 3:0.02 4:61.3 5:71.40000000000001 6:87 7:109 8:243
0.9914923297971016 1:7.6 2:20 4:36.6 5:60.4 6:85.90000000000001 7:118 8:320
-0.9688496889008199 1:7.3 2:20 4:96.8 5:98.09999999999999 6:94.40000000000001 7:138 8:269
0.9915268167580859 1:7.5 2:26 3:0.03 4:70.40000000000001 5:73.40000000000001 6:91.3 7:166 8:419
0.9915843733123155 1:7.5 2:53 3:0.02 4:49.3 5:60.8 6:80.8 7:172 8:345
0.9917917686487449 1:7.5 2:25 4:68.09999999999999 5:72.2 6:87.40000000000001 7:175 8:376
-0.9687861899204235 1:7.1 2:17 4:68 5:81.40000000000001 6:88.40000000000001 7:108 8:194