FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

FEUP

Formação Automática de Grupos para Projetos Estudantis

Ana Catarina Silva Carvalho

Mestrado Integrado em Engenharia Informática e Computação

Orientador: Ana Paula Rocha (Doutora)

12 de julho de 2012

Formação Automática de Grupos para Projetos Estudantis

Ana Catarina Silva Carvalho

Mestrado Integrado em Engenharia Informática e Computação

Aprovado em provas públicas pelo Júri:

Presidente: Henrique Daniel de Avelar Lopes Cardoso (Doutor)

Arguente: Paulo Jorge Freitas Oliveira Novais (Doutor)

Vogal: Ana Paula Cunha da Rocha (Doutora)

12 de julho de 2012

Resumo

A formação de grupos tem sido desde sempre um assunto investigado em diferentes áreas, de entre as quais se destacam a Sociologia, Psicologia, entre outras. Quando o número de estudantes é consideravelmente elevado, este problema assume largas proporções em termos de complexidade.

Com esta dissertação, tentam-se resolver dois problemas: avaliar quais fatores são relevantes na formação de um grupo e como formar estes grupos tendo toda a informação. Na formação de grupos de estudantes, os fatores a ponderar podem ser sociais, como interesses e amizades, ou pessoais, como a idade, localidade. Pretende-se estudar a relevância destes dois tipos de informação na escolha do grupo de trabalho mais adequado a um estudante.

A outra parte importante e desafiante é a extração de informação, principalmente a de tipo social existente em redes sociais, visto que as redes sociais (como o Facebook) usam bases de dados não-convencionais, o que eleva a extração desses dados a outro nível. Depois da extração de todos os dados, é mandatório definir o critério, por outras palavras, que fatores são mais relevantes que os outros e de acordo com isto, implementar uma metodologia adequada usando estes critérios.

Com esta dissertação, o problema com a formação de grupos da unidade curricular Projeto FEUP, leccionada na Faculdade de Engenharia da Universidade do Porto, será resolvido usando métodos automáticos (algoritmos de clustering), permitindo que a avaliação seja mais justa, visto que os grupos serão mais equilibrados ao contrário do método atual que aleatoriamente aloca estudantes num grupo.

Abstract

Group formations has always been an issue under study in different areas such as Sociology, Psicology, amont others. When the number of students is considerable large, this problem assumes large proportions in complexity.

With this dissertation, two problems are trying to be solved: evaluate which factors are relevant in this group formation and how to form those groups having all the information. Which factors are important in a group formation of students? Only social aspects as interests and friends or the age and location are relevant too?

The other important and challenging part is the extraction of this data since social networks (as Facebook) have non-convencional databases, which takes the extraction os those data to another level. After collecting all the data, is mandatory to define the criteria, or in another words, which factors are more relevant then the others and according to this, implement a methodology using those criteria.

In this dissertation, the problem with the group formation in the course Projeto FEUP, taught in Faculdade de Engenharia da Universidade do Porto, will be resolved using automatic methods (clustering algorithms), allowing the evaluation to be more fair, since the groups will be more balanced unlike the actual method which randomly assigns students to a group.

Agradecimentos

Gostaria de agradecer a todos os que contribuiram para esta dissertação, tanto diretamente como indiretamente.

Em primeiro lugar, o meu obrigada à minha orientadora, Professora Ana Paula Rocha, por todas as sugestões, ideias e contributo para que esta dissertação de tornasse possível. Ao Professor Armando Jorge Sousa, pelas informações cedidas relativas à unidade curricular Projeto FEUP. À minha família pelo seu amor e apoio de sempre.

Finalmente, um obrigada a todos os meus amigos que me ajudaram nestes cinco anos de faculdade.

Catarina Carvalho

"You're never fully dressed without a smile :)."

Martin Charnin

Conteúdo

Lista de Figuras

Lista de Tabelas

Abreviaturas e Símbolos

API	Application Programming Interface
CSV	Comma-separated values
DM	Data Mining
EDM	Educational Data Mining
FEUP	Faculdade de Engenharia da Universidade do Porto
FQL	Facebook Query Language
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
KDD	Knowledge Discovery in Databases
NUTS	Nomenclatura Comum das Unidades Territoriais Estatísticas (do francês, Nomenclature commune des Unités Territoriales Statistiques)
REST	Representational State Transfer
SiFEUP	Sistema de Informação da FEUP
SQL	Structured Query Language
XML	Extensible Markup Language

Capítulo 1

Introdução

1.1 Contexto/Enquadramento

Os projetos de grupo são uma componente importante na vida de um estudante. Estudantes a colaborarem na resolução de problemas em grupo tem muitas vantagens. Comparativamente com turmas grandes, os estudantes em grupo expoêm as suas dúvidas e problemas entre todos e debatem-nos mais frequentemente.

Esta interação é normalmente cooperativa em vez de competitiva, criando oportunidades de adquirir habilidades de *teamwork* e colaboração e desenvolver habilidades interpessoais[CP07].

A formação de grupos é um problema bastante conhecido e estudado. Como podemos formar grupos para um projeto garantindo que estes são equilibrados ou baseados em dados critérios? Um exemplo deste problema é a unidade curricular Projeto FEUP leccionada no primeiro ano dos cursos da Faculdade de Engenharia da Universidade do Porto. Os estudantes entram pela primeira vez nesta faculdade e a grande maioria desconhece o resto dos colegas, o que requere a criação dos grupos de trabalho por parte de outros. Mas a questão é como podemos agrupar estudantes tão diferentes em pequenos gruposde forma justa e equilibrada, em termos de competências e saber?[BDC09]

Um grupo de trabalho encoraja a aprendizagem e o apoio dos colegas permitindo aos estudates a oportunidade de clarificar e melhorar a sua percepção de determinados conceitos na discussão com os pares. Contudo, certos factores foram estudados como exercendo influência na dinâmica e performance do grupo.

Independentemente do critério usado, alocar estudantes num grupo é um problema complexo que não pode ser bem resolvido sem o suporte computacional.

1.2 Motivação e Objetivos

A motivação desta dissertação baseia-se em duas questões fundamentais: Como organizar os grupos de acordo com dados critérios, ou seja, como se organizam dezenas de estudantes considerando factores como os referidos anteriormente, e como é que estes factores realmente influenciam o sucesso dos estudantes em dado grupo.

A semântica pode ser usada no processo de identificação e extracção de informação útil para o reconhecimento e identificação de características do indivíduo (estudante). Neste cenário, as redes sociais são um conceito útil na identificação do círculo social a que um indivíduo pertence usando a informação retirada num contexto social. É importante considerar esta informação proveniente das redes sociais no processo de formação dos grupos, tentando agrupar estudantes que possuem personalidades compatíveis.

Toda esta informação (social, pessoal, interesses e conhecimento) é usada pelos algoritmos de clustering que tentam optimizar a formação dos grupos.

Este trabalho tem como primeiro objetivo o aprofundamento dos conhecimentos no domínio em que se insere, bem como fazer uma revisão dos trabalhos já desenvolvidos na área.

Outro dos objetivos deste trabalho passa pela identificação dos dados relevantes, por outras palavras, de toda a informação possivel de obter das redes sociais e do SiFEUP, encontrar os fatores relevantes e que são importantes para a formação dos grupos.

O objetivo seguinte passa pela extracção desses dados da rede social e do SiFEUP. Após a recolha de todos os dados, serão implementados algoritmos de clustering de modo a agrupar os indivíduos de acordo com os dados critérios. Segue-se a aplicação desta abordagem ao cenário da unidade curricular Projeto FEUP extensível posteriormente a outros cenários.

Para validação destas ideias, serão efetuados testes no cenário da unidade curricular Projeto FEUP e uma análise crítica dos resultados obtidos.

O aspecto inovador desta dissertação é o facto de usar informação de bases de dados não convencionais como é o caso das bases de dados das redes sociais.

1.3 Estrutura do documento

Além da Introdução, esta dissertação contém mais quatro capítulos. No capítulo 2, é descrito o estado na arte e o trabalho relacionado. Neste capítulo pode ler-se sobre os trabalhos relacionados com o tema desta dissertação e a revisão bibliográfica. No capítulo 3, são abordadas as perspectivas de solução as quais contêm mais detalhadamente o problema. De seguida, no capítulo 4, são referenciados todos os detalhes da implementação incluindo frameworks usadas e outras tecnologias. Neste capítulo também são referidas as formas de validação dos resultados. Por último, no capítulo 5 é feita uma análise crítica do trabalho desenvolvido bem como algumas conclusões.

Capítulo 2

Revisão Bibliográfica

Neste capítulo é descrito o estado da arte e apresentados todos os trabalhos relacionados com o mesmo domínio.

2.1 EDM: Educational Data Mining

Segundo o *website* da comunidade de *Educational Data Mining*, www.educationaldatamining.org, a definição de *educational data mining* é a que se segue:

> "*Educational Data Mining* é uma disciplina emergente, preocupada em desenvolver métodos para explorar os típos de dados únicos provenientes de definições educacionais, e usando esses métodos para melhor compreender os estudantes, e os trâmites em que eles aprendem" [BY09]

2.1.1 O que é EDM?

Educational data mining(EDM) é a área que explora algoritmos estatisticos, *machine-learning* e de *data mining*(DM) sobre diferentes tipos de dados provenientes da educação. O objetivo principal é analisar estes tipos de dados de modo a resolver questões na área da pesquisa educacional [HjHAk+09].

Por um lado, o aumento das aplicações educacionais bem como das bases de dados com informações dos estudantes criaram grandes repositórios de dados refletindo o ambiente em que os estudantes aprendem[KCSL08]. Por outro lado, o uso da internet revolucionou a educação criando o conceito de *e-learning* ou educação baseada na *web* onde um enorme conjunto de de informação sobre a interação ensino-aprendizagem foram gerados e disponibilizados [CVNM07]. Toda esta informação aglomerada providencia um precioso repositório de dados educacionais [MB06].

2.1.2 EDM: O antes e visão do futuro

Os métodos de EDM são extraidos de uma variedade de literaturas, incluindo DM e *machine-learning*, psicométricas e outras áreas da estatística, visualização da informação e modelação computacional. Segundo Romero e Ventura [RV10], o trabalho relacionado com EDM pode ser categorizado nas seguintes categorias:

- Estatísticas e visualização
- *Web mining*
    - *Clustering*, classificação
    - *Text mining*

Um segundo ponto de vista sobre EDM é dado por Baker [BY09], que classifica os trabalhos que abordam EDM da seguinte forma (figura 2.1):

- Previsão
    - Classificação
    - Regressão
    - Estimação de densidade
- *Clustering*
- *Mining* relacional
- Descoberta com modelos

2.1.3 EDM e DM: diferenças

De um ponto de vista prático, EDM permite, por exemplo, descobrir novo conhecimento baseado nos dados dos estudantes de modo a validar/avaliar sistemas educacionais que potencialmente melhoram alguns aspetos na qualidade da educação[RV04]. Algumas ideias semelhantes foram já implementadas com sucesso em sistemas de comércio electrónico, sendo esta a primeira e mais popular aplicação de DM[Rag05], de modo a determinar os interesses dos clientes podendo assim aumentar as suas vendas *online*. Contudo, até à data o progresso na área educacional tem sido comparativamente menor, embora esta situação esteja a mudar e atualmente tem havido um aumento de interesse na aplicação de DM ao ambiente educacional[RVPB10]. No entanto, existem alguns tópicos importantes que distinguem a aplicação de DM, especificamente na educação, de como é aplicada a outros domínios[RV07].

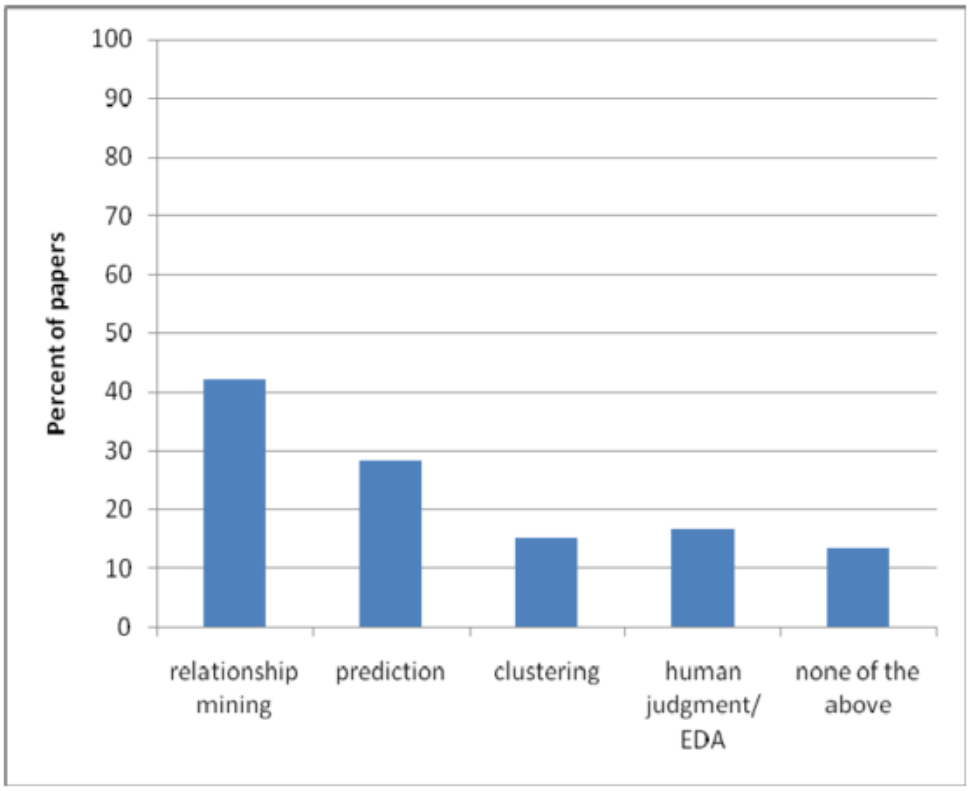


Figura 2.1: Proporção de artigos envolvendo cada tipo de método de EDM em 2007

Objetivo O objetivo de DM em cada área de aplicação é diferente. Por exemplo, em EDM, tanto são aplicados os chamados *research objectives*, como melhorar o precesso de aprendizagem para guiar a aprendizagem dos estudantes, bem como *pure research objectives*, tal como aprofundar o entendimento sobre o fenómeno educacional. Estes objetivos são por vezes dificeis de quantificar e requerem as suas técnicas de medidas especiais.

Dados Em cenários educacionais há uma variedade de tipos de dados disponíveis para *mining*. Estes dados sao específicos a esta área e por conseguinte possui informação semântica intrínsseca, relações com outros dados e múltiplos níveis hierárquicos significativos. Além disso, é também necessário tomar em consideração aspetos pedagógicos do aprendiz (estudante) e do sistema.

Técnicas Os dados e problemas educacionais possuem características especiais, o que requere que sejam tratados e processados de modo diferente. Embora a maioria das tecnicas tradicionais de DM possam ser aplicadas diretamente, outras não podem e têm que ser adaptadas ao problema específico. Para além disto, técnicas específicas de DM podem ser aplicadas para específicos problemas do foro educacional.

EDM envolve grupos diferentes de utilizadores ou participantes. Diferentes grupos avaliam a informação educacional de outra perspectiva, de acordo com a sua missão, visão e objetivos

Utilizadores/ Atores	Objetivos de usar DM
Aprendizes/ Estudantes	Para personalizar o *e-learning*: recomendar atividades aos estudantes, recursos e tarefas que podem melhorar a sua aprendizagem; sugerir experiências de aprendizagem interessantes de outros estudantes; etc
Educadores/ Professores/ Instrutores	Para obter *feedback* sobre o seu estilo de ensino; para analisar a aprendizagem e comportamento dos estudantes; para detetar que estudantes necessitam de mais apoio; para prever a performance dos estudantes; para classificar os aprendizes em grupos; para encontrar os padrões regulares e irregulares dos estudantes; para encontrar os erros mais frequentes; para determinar as atividades mais eficazes; etc
Pesquisadores Educacionais	Para avaliar o curso; para melhorar a aprendizagem dos estudantes; para avaliar a estrutura dos conteúdos do curso e a sua eficácia no processo de aprendizagem; para automaticamente construir modelos de estudantes e de tutores; para comparar as técnicas de DM de modo a ser capaz de recomendar a mais útil em cada tarefa; etc
Organizações/ Universidades	Para aumentar os procesos de decisão nas instituições de ensino superior; para alcançar objetivos específicos; para sugerir certos cursos que possam ser uma mais valia para certas turmas de estudantes; para encontrar o modo mais rentável de aumentar as notas; para ajudar a admitir estudantes que terão sucesso na universidade; etc
Administradores	Para desenvolver o melhor modo de organizar os recursos (humanos e materiais) institucionais e a sua oferta educativa; para utilizar os recursos disponíveis mais eficazmente; para aumentar as ofertas no programa educativo; para avaliar profesores e o seu currículo; etc

Tabela 2.1: Utilizadores de EDM/Intervenientes

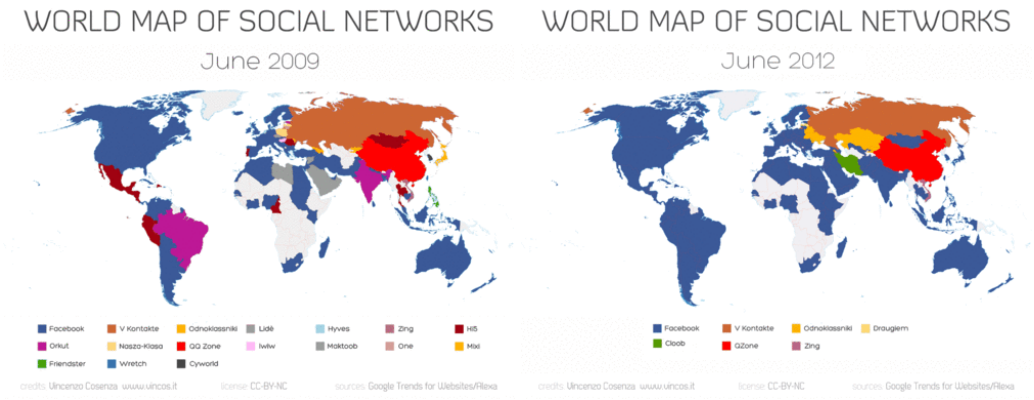
para usar DM[Han04]. Por exemplo, o conhecimento adquirido através de algoritmos de EDM pode ser usado não só para ajudar professores a gerir as suas aulas, a compreender o processo de aprendizagem dos seus alunos e a refletir sobre os seus próprios métodos de ensino, mas também para apoiar as reflexões dos alunos sobre as diversas situações e dar-lhes *feedback*[Mer05].

Embora as considerações iniciais parecerem envolver apenas dois grupos principais: os alunos e os professores, existem de facto mais grupos envolvidos e com mais objetivos como pode ser visto na tabela 2.1 .

2.2 Facebook: uma rede social

"I mean, we've built a lot of products that we think are good, and will help people share photos and share videos and write messages to each other. But it's really all about how people are spreading Facebook around the world in all these different countries. And that's what's so amazing about the scale that it's at today."

Mark Zuckerberg



(a) Redes sociais em junho 2009 (b) Redes sociais em junho 2012

Figura 2.2: Evolução das redes sociais a nível mundial

Facebook é uma popular rede social, cujo nome é oriundo do *nickname* dos diretórios dados aos estudantes universitários para os ajudar a conhecer os restantes estudantes.

2.2.1 A história do Facebook

O Facebook foi inventado pelo estudante de Ciências de Computadores de Harvard Mark Zuckerberg, juntamente com os seus companheiros Eduardo Saverin, Dustin Moskovitz e Chris Hughes. A Origem do seu nome aponta para a história do Facebook, contudo o *website* foi originalmente e aínda que por pouco pouco tempo chamado *Facemash*. Mark Zuckerberg era estudante do segundo ano quando criou o *software* para o *website* do *Facemash*. Usando as suas habilidades de programador para entrar ilegalmente na rede de segurança de Harvard, Mark copiou as fotografias de todos os estudantes e usou-as para preencher o *Facemash*.

O *Facemash* foi aberto a 28 de outubro de 2003 e fechado poucos dias depois por executivos de Harvard. Mark Zuckerberg enfrentou acusações de violação de segurança, violação de direitos de autor, e violação individual de privacidade por obter ilegalmente as fotografias dos estudantes. Foi também expulso da Universidade de Harvard pelas suas ações. Contudo, todas as queixas foram eventualmente arquivadas.

A 4 de fevereiro de 2004, Mark Zuckerberg abriu um novo *website*, *Thefacebook*. Seis dias depois, Mark enfrentou a justiça outra vez quando três *seniors* de Harvard o acusaram de roubar os seus planos para a criação de uma rede social chamada *HarvardConnection*. Contudo, também este assunto foi arquivado pela justiça.

Em 2004, um investidor, Sean Parker (fundador da Napster[1]) tornou-se o presidente da empresa na altura formada por Mark Zuckerberg, Eduardo Saverin, Dustin Moskovitz, Andrew McCollum e Chris Hughes. A empresa mudou o nome de *TheFacebook* para simplesmente *Facebook* como aínda é conhecida atualmente[Bel].

O *Facebook* tem crescido exponencialmente nos últimos anos, como é possível verificar na figura 2.2.1, sendo que atualmente é a rede preferida dos habitantes de 126 países[Vis12].

2.2.2 Facebook como base de dados não-convencional

As bases de dados não convencionais, também chamadas de bases de dados pós-relacionais visam atender as necessidades de gestão de dados de aplicações ditas não convencionais.

Exemplos deste tipo de bases de dados são as bases de dados orientadas a objetos (caso do *Facebook*), temporais, XML, entre outros.

O Facebook usa MySQL, mas principalmente para persistência de chaves-valores (Hashes), movendo lógicas de consultas e JOINS para a camada de aplicação dos servidores web em que as otimizações são mais fáceis de implementar, usando por exemplo caches em memória.

2.2.3 Plataforma do Facebook

A plataforma do *Facebook* providencia um conjunto de APIs e ferramentas que possibilitam programadores terceiros integrar com o *"open-graph"* - quer seja através de aplicações em *Facebook.com* ou *websites* e dispositivos externos. Lançada a 24 de maio de 2007, a plataforma do *Facebook* evoluiu desde permitir o simples desenvolvimento no domínio *Facebook.com* até suportar a integração deste através da *web* e dispositivos.

Estatísticas de maio de 2010 da plataforma do *Facebook*:

- Mais de um milhão de programadores e empresários de mais de 180 países
- Mais de 550,000 aplicações ativas na plataforma do *Facebook*
- Todos os meses, mais de 70% dos utilizadores do *Facebook* ligam-se a aplicaçoes da plataforma
- Mais de 250,000 *websites* integraram-se com a plataforma do *Facebook*
- Mais de 100 milhões de utilizadores do *Facebook* usam o *Facebook* através de *websites* externos todos os meses

Em 29 de agosto de 2007, o *Facebook* alterou a forma como a popularidade das aplicações era medida, dando mais atenção a aplicações que cativavam os utilizadores por mais tempo ao invés

[1] Napster, criado por Shawn Fanning e seu co-fundador Sean Parker, foi o programa de partilha de arquivos em rede P2P que protagonizou o primeiro grande episódio na luta jurídica entre a indústria fonográfica e as redes de partilha de música na internet. Compartilhando, principalmente, arquivos de música no formato MP3, o Napster permitia que os utilizadores fizessem o download de um determinado arquivo diretamente do computador de um ou mais utilizadores de maneira descentralizada, uma vez que cada computador conectado à sua rede desempenhava tanto as funções de servidor quanto as de cliente.

daquelas que tinham maior número de instalações. A partir desse ponto, as empresas começaram a desenvolver as chamadas aplicações "virais", isto é, aplicações cujo propósito é cativarem o utilizador de modo a que este sinta necessidade de a "visitar"regularmente.

A criação das aplicações "virais"é um método que tem certamente empregado inúmeros programadores de aplicações para o *Facebook*. A Universidade de Stanford chegou mesmo a oferecer em 2007 uma unidade curricular intitulada ”Criar Aplicações Web Cativantes Usando Métricas e Aprendizagem o Facebook”[2]. Numerosas aplicações criadas pela turma foram altamente sucedidas, competindo com as aplicações de topo do *Facebook*, algumas delas atingindo os 3,5 milhões de utilizadores mensalmente.

2.3 Algoritmos de Clustering

2.3.1 O que são os algoritmos de clustering?

Clustering pode ser considerado um dos mais importantes problemas de aprendizagem não supervisionada. Aprendizagem não supervisonada refere-se ao problema de tentar descobrir padrões ou relações entre dados com base em processos de auto-organização. Como todos os outros problemas deste género, lida com a procura de uma estrutura numa coleção de dados não identificados. Uma definição mais vaga poderia ser "o processo de organizar objetos em grupos cujos membros são semelhantes entre si de algum modo".

Um *cluster* é uma coleção de objetos mais semelhantes aos objetos do mesmo grupo do que aos que pertencem aos outros *clusters*.

O conceito de "centroide"é relevante no ambiente de algoritmos de *clustering*. Em certos algoritmos de *clustering* tona-se necessário definir um ponto de partida, isto é, o objeto ideal para um dado *cluster*. Esse objeto definido inicialmente chama-se centroide.

2.3.2 Objetivos do Clustering

O objetivo de *clustering* é determinar os grupos de um conjunto de dados não ”etiquetados”. Mas como se pode decidir o que constitui um bom *clustering*? Está demostrado que não há um critério que pode ser aplicado e que seria independente do objetivo final do *clustering*. Consequentemente, é o utilizador que deve fornecer este critério de modo a que o resultado de *clustering* satisfaça as suas necessidades.

2.3.3 Problemas

Há inúmeros problemas no uso de *clustering*. Entre eles salientam-se:

1. as técnicas atuais de *clustering* não respondem a todos os requisitos adequadamente (e concorrentemente);

[2] originalmente, ”Create Engaging Web Applications Using Metrics and Learning on Facebook”

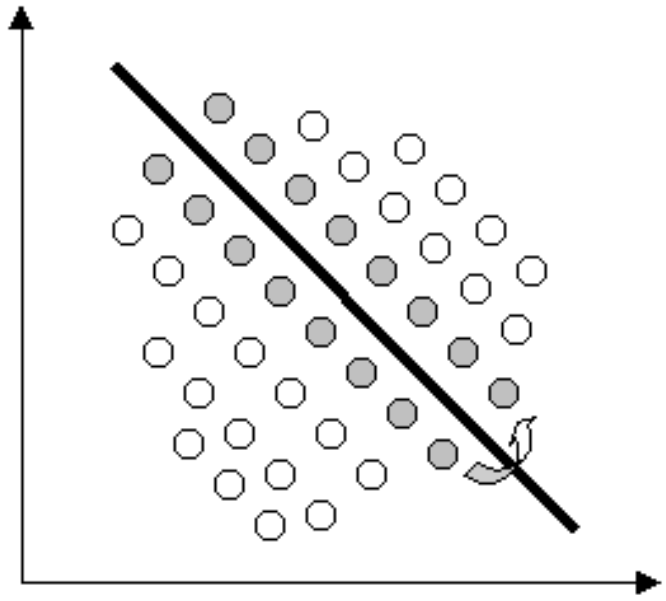


Figura 2.3: Exemplo de um agrupamento exclusivo

2. lidar com um grande número de dimensões e de dados pode ser problemático devido à complexidade temporal;

3. se uma medida de distância entre dois objetos não existe então é preciso defini-la, o que não é linear por vezes, especialmente em cenários multidimensionais.

2.3.4 Classificação

Os algoritmos de *clustering* podem ser classificados da seguinte forma:

- **Exclusivos (*Exclusive Clustering*)** - os dados são agrupados de um modo exclusivo, de modo a que se um dado objeto pertence a um *cluster*, não pode ser incluido noutro;
- **Fuzzy (*Overlapping Clustering*)** - usa conjuntos difusos para agrupar os dados, de modo a que cada objeto pode pertencer a dois ou mais *clusters* com diferentes graus de associação, assim os dados são associados a um dado valor de pertença;
- **Hierárquicos (*Hierarchical Clustering*)** - é baseado na união de dois *clusters* próximos. A condição inicial passa por definir cada objeto como um *cluster*. Após algumas iterações os grupos finais são encontrados;
- **Probabilísticos (*Probabilistic Clustering*)** - usa uma metodologia completamente probabilística.

Para explicar melhor cada tipo de algoritmos, serão explicados quatro dos algoritmos mais usados:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2 ,$$

Figura 2.4: Função objetivo do algoritmo K-means

- *K-means*
- *Fuzzy C-means*
- *Hierarchical Clustering*
- *Mixture of Gaussians*

Cada um destes algoritmos está associado a um dos tipos referidos anteriormente. O algoritmo *K-means* é um algoritmo do tipo exclusivo, *Fuxxy C-means* é um algoritmo do tipo fuzzy, e o *Hierarchical Clustering* e *Mixture of Gaussians* pertencem aos tipos hierárquico e probabilístico, respetivamente. Estes algoritmos serão discutidos nas secções seguintes.

2.3.4.1 K-Means

K-means (MacQueen, 1967) é um dos algoritmos de aprendizagem não supervisionada mais simples que solve o conhecido problema de *clustering*. O procedimento segue uma forma simples e fácil de classificar um certo conjunto de dados num dado número de *clusters* (admitam-se *k clusters*) fixados *a priori*. A ideia principal é definir *k* centoides, um para cada *cluster*. Estes centroides devem ser colocados de forma astuta por locais diferentes consequentemente implicam resultados diferentes, e por isso, a melhor forma é localizá-los o mais longe possível de cada um. Quando todos os centroides estão definidos, o primeiro passo está completo e o primeiro agrupamento feito. Nesta altura é necessário recalcular *k* novos centroides como "centros de gravidade"dos *clusters* resultantes do passo anterior. Depois de obtidos estes *k* novos centroides, terá de ser feita feita uma nova ligação entre os mesmos pontos do conjunto de dados e novo centroide mais próximo. Um ciclo é gerado. Como resultado deste ciclo podemos reparar que os novos *k* centroides mudam a sua localização passo-a-passo até não ocorrerem novas alterações. Finalmente, este algoritmo tem como objetivo a minimização de uma função objetivo, neste caso uma função de erro.

Embora possa ser provado que o procedimento irá sempre terminar, o algoritmo *k-means* não encontra necessariamente a melhor solução, correspondendo ao valor minimo da função objetivo global. O algoritmo é também significativamente sensível ao valor inicial gerado aleatoriamente para os centroides. O algoritmo *k-means* pode executar múltilplas vezes de forma a reduzir este efeito.

Resumidamente, o algoritmo é composto pelos seguintes passos:

1. Colocar K pontos no espaço representado pelos objetos que vão ser agrupados. Estes pontos representam centróides iniciais do grupo;

$$J_m = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^{m} \left\| x_i - c_j \right\|^2 \quad , \quad 1 \leq m < \infty$$

Figura 2.5: Função objetivo do algoritmo Fuzzy C-means

2. Associar cada objeto ao grupo com o centróide mais próximo;

3. Quando todos os objetos forem atribuídos, recalcular as posições dos K centróides;

4. Repetir as etapas 2 e 3 até que os centróides já não mudem. Isto produz uma separação dos objectos em grupos a partir da qual a métrica a ser minimizada pode ser calculada.

2.3.4.2 Fuzzy C-Means

Fuzzy C-means (FCM) é um método de *clustering* que permite que um objeto pertença a dois ou mais *clusters*. Este método (desenvolvido por Dunn em 1973 e melhorada por Bezdek em 1981) é frequentemente utilizado em padrões de reconhecimento. É baseado na minimização da seguinte função objectivo:

A partição *fuzzy* é levada a cabo através de uma optimização iterativa da função objectivo apresentada de seguida, com o valor de pertença u_{ij} e centróides c_j por:

Esta iteração termina quando a diferença do valor de u_{ij} de dois passos consecutivos pertence ao intervalo [0,1].

Sucintamente, o algoritmo segue os seguintes passos:

1. Inicializar a matriz $U = [u_{ij}]$, $U^{(0)}$;

2. No k passo: calcular os vectores centrais $C^{(k)} = [c_j]$ usando $U^{(k)}$;

3. Atualizar $U^{(k)}$,$U^{(k+1)}$;

4. Se a diferença dos valores calculados em 3 dor menor que 1, STOP; caso contrário retorna ao passo 2.

2.3.4.3 Hierarchical Clustering Algorithms

Dado um conjunto de N items a serem agrupados, e uma distância na matriz de N*N (ou semelhante), o processo básico de *clustering* hierárquico (definido por S. C. Johnson em 1967) é o seguinte:

1. Começar por endereçar cada elemento a um *cluster*, de forma a que, se existirem N items, passa-se a ter N *clusters*, cada um contendo apenas um item. Deixar as distâncias (semelhanças) entre os *clusters* iguais às distâncias (semelhanças) entre os items que estes contêm.

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}} \qquad c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} \cdot x_i}{\sum_{i=1}^{N} u_{ij}^{m}}$$

Figura 2.6: Função para cálculo do valor de pertença de um objeto a um grupo do algoritmo Fuzzy C-means

2. Encontrar o mais próximo (mais semelhante) par de *clusters* e fundir os seus elementos num *cluster* único, de forma a que tenhamos menos um *cluster*.

3. 3. Computar distâncias (semelhanças) entre o novo *cluster* e cada um dos antigos *clusters*.

4. 4. Repetir os passos 2 e 3 até que todos os items estejam agrupado num único *cluster* de tamanho N.

O passo 3 pode ser realizado de maneiras diferentes, que é o que distingue uma única ligação de ligação completa e *clustering* média-ligação. Num *clustering* de ligação única (também denominado de "sem conexão" ou "método mínimo"), considera-se que a distância entre um *cluster* e outro *cluster* é igual á distância mais curta entre qualquer membro de um *cluster* para outro membro de outro *cluster*. Se os dados consistirem em semelhanças, considera-se que a semelhança entre um *cluster* e outro *cluster* é igual á distância mais longa entre qualquer membro de um *cluster* para outro membro de outro *cluster*. Numa *clustering* de ligação completa (também denominada de "diâmetro" ou "método máximo"), considera-se que a distância entre dois *clusters* é igual á distância maior entre qualquer membro de um *cluster* para outro membro de outro *cluster*. No *clustering* de média-ligação, considera-se que a distância entre dois *clusters* é igual à média das distâncias entre qualquer membro de um *cluster* para outro membro de outro *cluster*.

2.3.4.4 Mixture of Gaussians

Há uma outra forma de lidar com problemas de *clustering*: uma abordagem baseada em modelo, que consiste no uso de determinados modelos de *clusters* e tentar otimizar o ajuste entre os dados e o modelo. Na prática, cada grupo pode ser matematicamente representado por uma distribuição paramétrica, como uma gaussiana (contínua) ou Poisson (discreta). O conjunto inteiro de dados é, por conseguinte, modelado por uma mistura destas distribuições. Uma distribuição individual usada para modelar um *cluster* específico é frequentemente referida como uma distribuição de componentes.

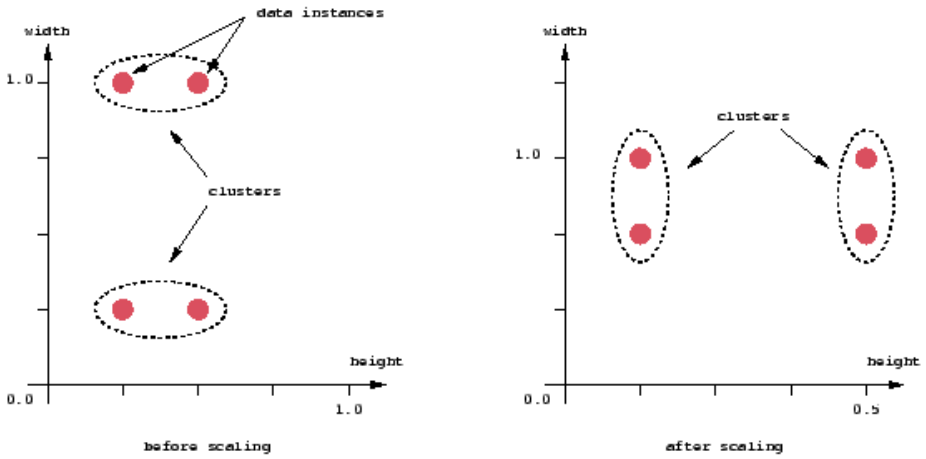


Figura 2.7: Avaliação da distância de dois grupos

2.3.5 Função distância

Um componente importante de um algoritmo de *clustering* é a medida da distância entre um par de dados. Se os componentes dos vectores instância de dados estão todos nas mesmas unidades físicas, então é possível que a métrica de distância Euclidiana simples seja suficiente para agrupar as instâncias de dados semelhantes com sucesso.

No entanto, mesmo neste caso, a distância Euclidiana pode às vezes ser incorreta. A figura 2.7 representada abaixo ilustra este com um exemplo da medida da largura e altura de um objeto. Apesar de ambas as medidas terem as mesmas unidades físicas, uma decisão informada tem que ser feito quanto à escala relativa. Como mostra a figura, escalas diferentes podem levar a agrupamentos diferentes.

De notar, contudo, que esta não é apenas uma questão gráfica: o problema surge a partir da fórmula matemática usada para combinar as distâncias entre os componentes individuais dos vetores de dados numa única medida de distância que pode ser usado para fins de agrupamento: fórmulas diferentes conduzem a diferentes agrupamentos. Novamente, o conhecimento do domínio deve ser utilizado para orientar a formulação de uma medida de distância adequada para cada aplicação particular.

2.4 Trabalho relacionado

A formação automática de grupos é um problema frequentemente estudado na ótica de diferentes áreas, como Psicologia, Sociologia e Ciências da Educação.

Um método de atribuição simples é distribuir aleatoriamente os alunos em grupos [Smi89]. Este método tem a vantagem de não necessitar de qualquer planeamento antecipado, é simples e transparente para os alunos, mas as diferenças dos alunos e as competências não são levados em conta e pode levar a grupos desequilibrados[BSA01].

A maioria das aplicações existentes segue uma perspectiva de auto-seleção [TB11], onde o aluno seleciona outros potenciais estudantes que podem ajudá-lo a alcançar o seu objetivo de aprendizagem, permitindo que os alunos selecionados aceitem ou rejeitem a conexão com o grupo [WP01]. Este método também não requer planeamento antecipado e tem a vantagem de deixar os estudantes escolher os colegas com motivações semelhantes.

Alternativamente, os professores podem utilizar o critério baseado nas notas dos estudantes para formar os grupos, sendo os métodos mais comuns notas altas-notas altas e notas altas-notas baixas [JJS91]. Ambos os métodos são simples de colocar em prática, mas exigem a presença de notas prévias dos estudantes em outras disciplinas ou em exames anteriores da atual disciplina[Del01].

De acordo com Redmond [Red01], em termos de formação de grupos, muitas vezes aparecem estudantes que não pertencem a nenhum grupo. Estes estudantes "órfãos"são excluídos por não terem nada em comum com os estudantes já agrupados ou simplesmente porque eles não mostram qualquer interesse nos grupos existentes.

Seguindo a ideologia de Tobar e Freitas [TF07], a mais refinada é a informação para definir os grupos, a mais dificil e demorada é a definição manual dos grupos.

Comparativamente à modelação de caraterísticas dos estudantes, estes autores consideram uma vasta gama de funcionalidades para diferentes formações de grupos, usando uma série de domínios ontológicos, que podem formar perfis de estudantes de forma fiável e dinâmica [ODM]. Neste contexto, a modelação semântica fornece descrições de estudantes e as relações entre eles. Exemplos destas carateristicas moldadas são os detalhes pessoais, curso, interesses, preferências, amigos, colegas e assim por diante [ODM].

Capítulo 3

Modelação do Problema

3.1 Formação Automática de Grupos

Como parte integrante da vida de um estudante, os projetos de grupo ajudam a desenvolver as habilidades e as personalidades dos estudantes.

> ”Formar tais grupos não é nem pedagogicamente nem computacionalmente fácil. Primeiro, que tipo de estudantes devem pertencer ao mesmo grupo de modo a maximizar a eficácia da aprendizagem? Segundo, após o critério estar definido, como é encontrado o grupo ótimo?” [HÏ0]

Há inúmeros fatores que se podem considerar aquando da formação de grupos para projetos estudantis. Exemplos deles são os conhecimentos ou habilidades dos estudantes, as suas amizades e interesses, bem como a sua personalidade. Torna-se necessário avaliar estes dados de modo a formar os grupos seguindo um dado padrão. Estes fatores necessariamente a ponderar na formação de um grupo podem ser classificados em dois tipos: de tipo social (dados provenientes de uma rede social, como o *Facebook*) e de tipo pessoal (dados retirados de uma base de dados com informação de estudantes, como o SiFEUP).

3.2 Metodologia

A metodologia usada no desenvolvimento deste trabalho envolve fases distintas. Uma primeira parte, sobre os critérios, envolve a escolha das caraterísticas a usar aquando da definição do perfil dos estudantes e o critério da formação dos grupos. Seguidamente existe o problema da seleção e extração destes dados. Posteriormente, torna-se necessário organizar e processar estes dados de modo a serem os dados de entrada do algoritmo de *clustering*. Por último segue-se a aplicação deste processo a um cenário real: o Projeto FEUP.

3.2.1 Critérios importantes

De todos os critérios possíveis de implementar, foi feita uma seleção. Visto que a aplicação a desenvolver é em primeira instância destinada à unidade curricular Projeto FEUP, foram considerados importantes fatores como a idade do aluno, local de residência, a última escola frequentada e os interesses pessoais (componente social). Analisando cada fator individualmente, enumera-se nos pontos seguintes a razão desta escolha:

Idade A idade é um dos fatores que influenciam indiretamente a personalidade de um estudante. Maioritariamente, estudantes mais velhos têm uma maior abertura para trabalhos em grupo visto serem mais ponderados, sendo que, por isso a idade é um fator a ter em conta aquando da formação dos grupos.

Local de Residência O local de residência de um estudante pode influenciar a dinâmica de um grupo na medida em que há maior probabilidade de dois estudantes se conhecerem se forem da mesma área de residência do que se forem de diferentes origens.

Escola Frequentada À semelhança do fator local de residência, também o fator escola frequentada toma a mesma importância.

Interesses Pessoais Este é um dos fatores mais valorizado nesta dissertação. A componente social da vida de um estudante tem um grande peso aquando da sua tomada de decisões. Estudantes com interesses semelhantes, à partida possuem uma maior compatibilidade no que toca a tomada de decisões simples. Por exemplo, dois estudantes com interesse num lado mais artístico serão mais compatíveis quando toca à criação de uma apresentação.

3.2.2 Extração e Organização dos dados

O objetivo primário desta dissertação incluía a obtenção de dados através do SiFEUP e do *Facebook*. Por motivos de confidencialidade, os dados de tipo pessoal que seriam retirados do SIFEUP (idade, escola frequentada e local de residência) não puderam ser facultados, pelo que estes mesmos dados tiveram que ser obtidos através do *Facebook*. Deste modo de todos os dados possíveis de extração do *Facebook* foram selecionados apenas os alunos que apresentavam os quatro fatores necessários.

Após os dados serem recolhidos, é necessário organizá-los de acordo com certos parâmetros, isto é, tornar os dados compatíveis com os requisitos necessários usados no algoritmo escolhido.

3.2.3 Formação dos Clusters

Outra parte do problema a resolver é a formação dos *clusters*. De uma forma resumida, um *cluster* é um grupo de algo, neste caso estudantes, com características que seguem um dado padrão. O problema da formação de *clusters* com atributos específicos é o facto de ser preciso definir, em alguns casos, uma função de distância, isto é, uma função que calcule quão distantes dois

indivíduos (estudantes) estão um do outro. Especificamente para este problema, é necessário encontrar um algoritmo de *clustering* que se adeque às especificações do conjunto de atributos escolhido.

3.2.4 Aplicação: Projeto FEUP

Nas primeiras edições da unidade curricular Projeto FEUP, usou-se o método de auto-seleção de grupos, isto é, os estudantes escolhiam o seu grupo. Embora este processo traga algumas vantagens, os estudantes habituam-se a trabalhar com os mesmos colegas enquanto frequentam a universidade e, em alguns casos, negoceiam entre eles o seu envolvimento em cada unidade curricular, o que é um efeito colateral altamente indesejável. Adicionalmente, alguns estudantes reivindicam que estudantes acima da média tendem a ficar juntos no mesmo grupo.

Embora o método de auto-seleção seja uma prática comum na Faculdade de Engenharia da Universidade do Porto, os responsáveis da unidade curricular aperceberam-se do ”ponto fraco” do método. Por conseguinte, os responsáveis decidiram experimentar um novo método baseado no perfil dos estudantes. Neste método, é pedido a cada estudante para responder a um questionário com o objetivo de avaliar o seu papel natural enquanto inserido num grupo de trabalho. Como resultado deste questionário, a cada estudante é atribuido um dos oito perfis: presidente, estratega, inteletual, avaliador, operacional, trabalhador de equipa, explorador e retocador. Por exemplo, um estudante competente a delinear linhas de ação para o projeto teria uma avaliação mais elevada no parâmetro estratégico. De notar que alguns estudantes podem não ter um perfil predominante, apresentando pontuações elevadas para algumas variáveis. Com este método, os estudantes têm que aprender a trabalhar com colegas com os quais é improvável que já tenham trabalhado antes. Este facto ajuda os estudantes a desenvolver competências de trabalho em equipa que serão importantes na sua vida profissional [BDC09].

A última fase do problema é a aplicação de todo este processo à unidade curricular Projeto FEUP. Usando todos os alunos inscritos nesta unidade curricular (cerca de 1000 alunos), processando os seus dados e formando os respetivos *clusters*, será possível formar grupos de 5/6 elementos sendo que todos os grupos serão similares. Esta metodologia difere da usada atualmente na unidade curricular na medida em que em primeiro lugar não são considerados os cursos dos alunos mas sim a sua componente social.

3.3 Resumo

Embora a formação automática de grupos seja um problema bastante estudado, torna-se necessário efetuar uma avaliação dos fatores que influenciam este agrupamento. Para cada cenário deve ser estudado quais e em que medida os fatores têm influência na performance e dinâmica do grupo. Nesta dissertação foram avaliadas as características que são importantes na unidade curricular Projeto FEUP e que são usadas no algoritmo de formação dos grupos. Encontrar este tipo de informação nem sempre é uma tarefa fácil, e neste caso, será usada uma base de dados não convencional como fonte de informação.

Capítulo 4

Implementação

No capítulo anterior foram descritos os principais problemas a ter em conta na implementação da metodologia escolhida. Neste capítulo é descrito o processo de implementação, incluindo os protocolos usados, APIs e ferramentas externas apropriadas para a invocação dos algoritmos de *clustering* a testar.

4.1 RestFB: Uma interface Java para a API do Facebook

A API do *Facebook* baseia-se no protocolo HTTP (Hypertext Transfer Protocol). Os pedidos HTTP são criados escolhendo o método HTTP apropriado de POST, GET e DELETE e criando o URL apropriado que acede aos dados pretendidos. O método POST é usado para enviar dados para o servidor, o GET para solicitar dados ao servidor e o DELETE para excluir objetos deste. Uma vez que estes são métodos HTTP nativos, estão propriamente documentados [FGM+99] e há inúmeros exemplos da sua utilização disponíveis *online*. Inicialmente um servidor REST[1] (REpresentational State Transfer) controlava todos os pedidos da API do *Facebook* mas a mais recente implementação, denominada Graph API, sobrepôs-se à implementação anterior, em prol de respostas orientadas a JSON (JavaScript Object Notation). JSON é um subconjunto da notação de objeto de JavaScript, mas o seu uso não requer JavaScript exclusivamente.

```
1 {
2 { "Aluno" : [
3         { "nome": "JoÃ£o", "notas": [ 8, 9, 7 ]  },
4         { "nome": "Maria", "notas": [ 8, 10, 7 ] },
5         { "nome": "Pedro", "notas": [ 10, 10, 9 ] }
6     ]
```

[1] "A REST (Transferência do Estado Representacional) é pretendida como uma imagem do design da aplicação se comportará: uma rede de websites (um estado virtual), onde o usuário progride com uma aplicação selecionando as ligações (transições do estado), tendo como resultado a página seguinte (que representa o estado seguinte da aplicação) que está sendo transferida ao usuário e apresentada para seu uso."Dr. Roy Fielding

```
7 }
8 }
```

Listing 4.1: Exemplo de um objeto JSON

Para facilitar a ligação entre código Java e a API do *Facebook*, usou-se o *RestFB*.

RestFB é um cliente Java simples e flexivel para *Facebook Graph API*[Fac11b][2]. A versão mais atualizada (1.6.8) foi lançada a 15 de outubro de 2011.[Res]

Enumeram-se seguidamente alguns conceitos base necessários para uma melhor compreensão da interface *RestFB*:

Graph API Cada objeto no grafo social tem um identificador único. Deste modo pode aceder-se às propriedades de um objeto fazendo o pedido $https://graph.facebook.com/ID$.

```
1  {
2     "name": "Facebook Platform",
3     "website": "http://developers.facebook.com",
4     "username": "platform",
5     "founded": "May 2007",
6     "company_overview": "Facebook Platform enables anyone to build...",
7     "mission": "To make the web more open and social.",
8     "products": "Facebook Application Programming Interface (API)...",
9     "likes": 449921,
10    "id": 19292868552,
11    "category": "Technology"
12 }
```

Listing 4.2: Exemplo do pedido $https://graph.facebook.com/19292868552$

Autenticação Para iniciar a extração dos dados do *Facebook*, é necessário, em primeiro lugar, autenticar-se no *Facebook*. Esta autenticação é estabelecida usando OAuth 2.0[HLRH11b][3]: depois de o utilizador ser autenticado, é adquirido um *token* do servidor, sendo a partir daí usado nos pedidos seguintes para identificação do utilizador, enquanto válido. Os *tokens* OAuth têm um tempo de vida e têm que ser renovados depois de expirarem. [Fac11a] [HLRH11a]

Após a autorização ser concedida, a aplicação pode aceder a todas as informações disponíveis sobre o perfil do utilizador, as suas mensagens, fotos, eventos ou amigos. Além disso, é possível publicar novos conteúdos. O exemplo 4.3 mostra como fazer um pedido da lista de amigos de um utilizador usando a interface RestFB.

[2] Apresenta uma vista simples e consistente do gráfico social do Facebook, representando de forma uniforme os objetos do grafo (por exemplo pessoas, fotos, eventos e páginas) e as conexões entre eles (amizades, conteúdo partilhado e identificações em fotos)

[3] *Open standart for authorization*

```
1 User user = facebookClient.fetchObject("me", User.class);
2 Connection<User> myFriends = facebookClient.fetchConnection("me/friends",
      User.class);
```

Listing 4.3: Exemplo de como obter a lista de amigos de um utilizador

Obter objetos simples Para fazer pedidos à API, é necessário indicar ao RestFB como transformar os objetos JSON retornados pelo *Facebook* em objetos Java. No exemplo 4.4, os objetod pedidos devem ser mapeados para os tipos **User** e **Page**.

```
1 User user = facebookClient.fetchObject("me", User.class);
2 Page page = facebookClient.fetchObject("cocacola", Page.class);
3
4 out.println("User name: " + user.getName());
5 out.println("Page likes: " + page.getLikes());
```

Listing 4.4: Exemplo de como obter a lista de amigos de um utilizador

FQL O *Facebook* tem a sua linguagem própria para efetuar pedidos à sua base de dados. A *Facebook Query Language* (FQL) [Fac12], permite usar uma interface do estilo SQL para obter os dados exposto pela Graph API. Adiciona funcionalidades não inseridas na Graph API pois aceita múltiplas *queries* numa única chamada como exemplificado no exemplo 4.5.

As chamadas seguem a forma **SELECT [fields] FROM [table] WHERE [conditions]**, com a única condição de que a cláusula FROM apenas pode conter uma tabela.

```
1 Map<String, String> queries = new HashMap<String, String>();
2 queries.put("users", "SELECT uid, name FROM user WHERE uid=220439 OR uid
      =7901103");
3 queries.put("likers", "SELECT user_id FROM like WHERE object_id=122788341354"
      );
4
5 MultiqueryResults multiqueryResults = facebookClient.executeMultiquery(
      queries, MultiqueryResults.class);
6
7 out.println("Users: " + multiqueryResults.users);
8 out.println("People who liked: " + multiqueryResults.likers);
```

Listing 4.5: Exemplo de como usar FQL para obter dados de utilizadores e páginas

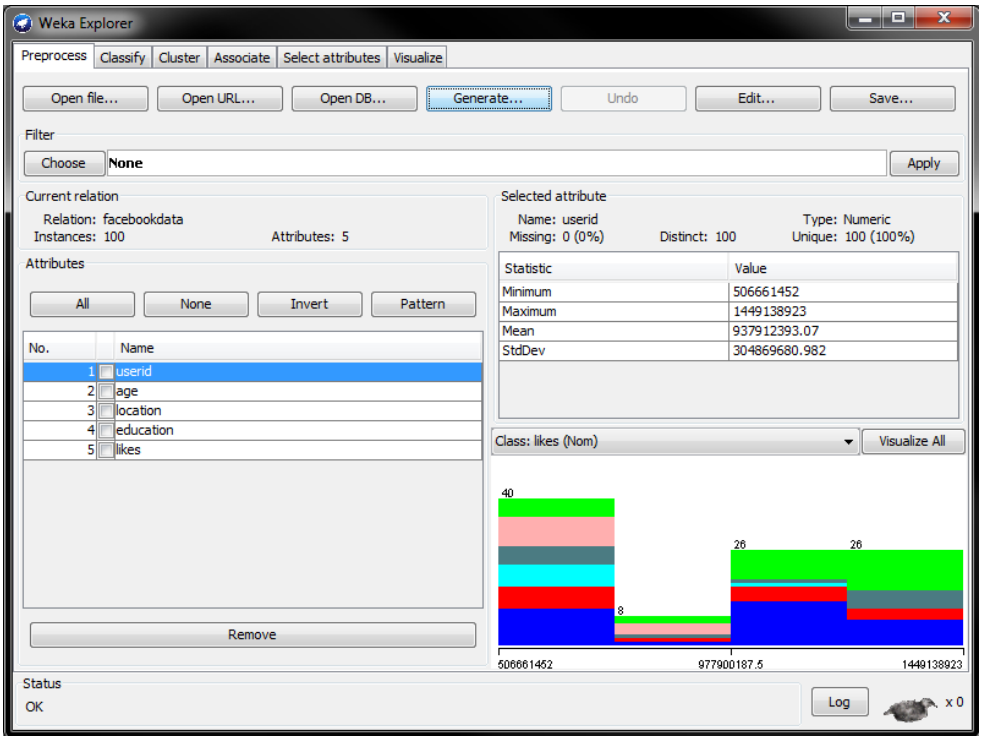


Figura 4.1: Interface Weka

4.2 Weka: Data Mining Software em Java

Weka é um software *open-source* que contém uma coleção de algoritmos de aprendizagem para tarefas de DM. Os algoritmos podem ser aplicados diretamente a um conjunto de dados ou chamados a partir do seu próprio código Java. Weka contém ferramentas para pré-processamento de dados, classificação, regressão, *clustering*, regras de associação e visualização dos resultados. É também ideal para o desenvolvimento de novos sistemas de aprendizagem de *machine learning*[Wek].

4.2.1 Tipo de dados

De modo a serem reconhecidos, os dados que serão agrupados através do algoritmo têm que respeitar certos parâmetros. Estes dados são armazenados num ficheiro que será posteriormente lido. A Weka possui um formato próprio, o ARFF, no qual temos que descrever o domínio do atributo, pois o mesmo não pode ser obtido automaticamente pelo seu valor.

No cabeçalho do ficheiro que contém os dados, devem estar a especificação dos atributos e relações, seguidos dos dados organizados no formato CSV.

Relation O nome da relação (nome do conjunto de dados) é definido na primeira linha do ficheiro ARFF. O formato é o apresentado em 4.2.1 onde $< relation - name >$ é uma *string*.

```
1 @relation <relation-name>
```

Attribute A declaração dos atributos toma a forma de uma sequência de declarações *@attribute*. Cada atributo dos dados tem a sua declaração *@attribute* própria que define unicamente o nome e tipo do atributo. A ordem em que os atributos são declarados indica a posição da coluna na secção *@data* que contém os dados. Por exemplo, se um atributo é o terceiro a ser declarado, a Weka espera que todos os valores correspondentes a esse atributo sejam encontrados na terceira coluna. O formato da declaração *@attribute* é :

```
1 @attribute <attribute-name> <datatype>
```

onde $< attribute - name >$ tem que começar por uma letra.

$< datatype >$ pode ser qualquer um dos quatro formatos suportados pela Weka:

- *numeric* - podem ser números reais ou inteiros;
- *integer* - que é tratado como *numeric*;
- *real* - que é tratado como *numeric*;
- <especificação-nominal> - por exemplo, @ATTRIBUTE classe tipo1,tipo2,tipo3;
- *string* - podem conter valores arbitrários desde que depois sejam implementados filtros na Weka para os manipular;
- *date*[<formato>] - por exemplo, @ATTRIBUTE abertura date [yyyy-MM-dd'T'HH:mm:ss]

Data A declaração *@data* é uma única linha que indica o início do segmento de dados no ficheiro.

Cada instância é representada numa única linha, seguindo o formato CSV. Cada valor em falta deve ser substituido pelo caracter '?'.

```
1  @RELATION iris
2
3  @ATTRIBUTE sepallength  NUMERIC
4  @ATTRIBUTE sepalwidth   NUMERIC
5  @ATTRIBUTE petallength  NUMERIC
6  @ATTRIBUTE petalwidth   NUMERIC
7  @ATTRIBUTE class        {Iris-setosa,Iris-versicolor,Iris-virginica}
8
9  @DATA
10 5.1,3.5,1.4,0.2,Iris-setosa
11 4.9,3.0,1.4,0.2,Iris-setosa
12 4.7,3.2,1.3,0.2,Iris-setosa
13 4.6,3.1,1.5,0.2,Iris-setosa
14 5.0,3.6,1.4,0.2,Iris-setosa
15 5.4,3.9,1.7,0.4,Iris-setosa
```

```
16 4.6,3.4,1.4,0.3,Iris-setosa
17 5.0,3.4,1.5,0.2,Iris-setosa
18 4.4,2.9,1.4,0.2,Iris-setosa
19 4.9,3.1,1.5,0.1,Iris-setosa
```

Listing 4.6: Exemplo de um ficheiro de dados

4.2.2 Estrutura dos dados

Os fatores usados, como referido anteriormente, são a idade, local de residência, escola frequentada e interesses pessoais. Tendo em conta o formato dos dados requeridos pela Weka, os atributos foram definidos da seguinte forma:

```
1 @attribute userid numeric
2 @attribute age numeric
3 @attribute location {Abrantes, Agueda, Aguiar da Beira,...}
4 @attribute education real
5 @attribute likes {music,movies,sports,arts,science,technology}
```

Listing 4.7: Atributos usados

Visto a biblioteca Weka apenas aceitar dados definidos *a priori* para os atributos, para o atributo local de residência (*location*) foi necessário estabelecer que valores seriam aceites. Deste modo, e seguindo a divisão por concelhos estabelecida em Portugal [Wik12], são aceites os 308 concelhos definidos.

De igual modo, os interesses pessoais (*likes* de cada utilizador no *Facebook*) tiveram que ser agrupados em tipos. Por conseguinte, foram definidos os tipos apresentados na tabela 4.7.

Postas todas estas condições, tornou-se necessário definir uma nova função de distância, isto é, uma função que dados dois indivíduos retornasse um valor indicando a distância entre eles. Assim sendo, estabeleceram-se as seguintes premissas:

1. concelhos pertencentes ao mesmo NUTS (Nomenclatura Comum das Unidades Territoriais Estatísticas) de nível III[Wik12] têm como valor de distância *1*;

2. concelhos pertencentes a diferentes níveis NUTS de nível III mas ao mesmo NUTS de nível II[Wik12] têm como valor de distância *3*;

3. concelhos pertencentes a diferentes níveis NUTS de nível II mas ao mesmo NUTS de nível I[Wik12] têm como valor de distância *5*;

Estes valores adicionam-se ao cálculo das distâncias dos outros atributos, sendo que, com um par de objetos, é possível computar a sua distância, construindo assim os grupos.

4.2.3 Agrupamento

4.2.3.1 Escolha do algoritmo

Dos algoritmos apresentados, foi efetuada uma seleção no sentido de escolher o algoritmo mais indicado para a formação deste tipo de grupos. Segundo a análise de Abu Abbas [Abb08], foram analisados três algoritmos: *K-means*, EM (*Expectation Maximization*) e o algoritmo hierárquico.

EM é um algoritmo bem estabelecido na comunidade estatística. É um algoritmo baseado na distância assumindo que cada conjunto de dados pode ser modelado como uma combinação linear de variadas distribuições normais e o algoritmo descobre os parâmetros da distribuição que maximizam a medida de qualidade do modelo, denominado "*log likelihood*".

Os algoritmos foram comparados de acordo com os seguintes fatores:

- número de *clusters*
- tamanho do conjunto de dados

No que respeita ao número de *clusters*, *k*, exceto o algoritmo hierárquico, todos os algoritmos requerem que o valor de *k* seja especificado inicialmente. À medida que os valores de *k* vão aumentando, verificou-se que a performance dos algoritmos *K-means* e EM torna-se superior à do algoritmo hierárquico, como é possível verificar na tabela 4.1.

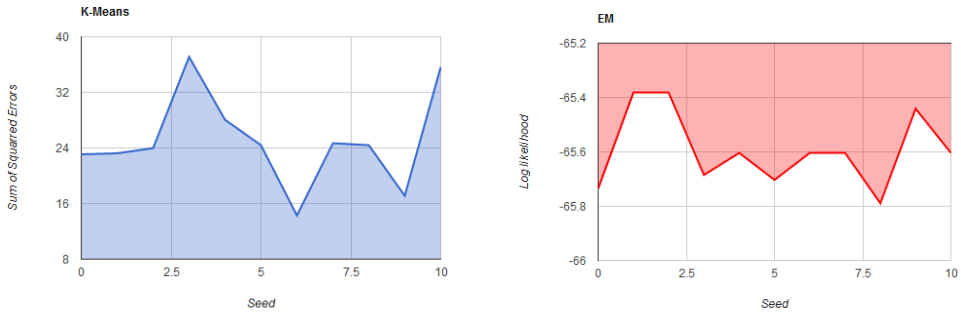
Número de *clusters* (k)	Performance		
	K-means	EM	Hierárquico
8	63	62	65
16	71	69	74
32	84	84	87
64	89	89	92

Tabela 4.1: Relação entre o número de *clusters* e a performance dos algoritmos

Relativamente ao conjunto de dados, o conjunto considerado grande consistiu em 600 linhas e 60 colunas enquanto que o pequeno em 200 linhas e 20 colunas. A qualidade dos algoritmos EM e *K-means* revelou-se superior quando usados conjuntos de dados grandes (ver tabela 4.2).

K = 32			
Tamanho	K-means	EM	Hierárquico
36000	910	898	850
4000	95	93	91

Tabela 4.2: O efeito do tamanho dos dados nos algoritmos



(a) Algoritmo *K-means* (b) Algoritmo EM

Figura 4.2: Comparação do desempenho dos algoritmos variando o parâmetro *seed*

Visto o número de estudantes inscritos na unidade curricular Projeto FEUP rondar os 1000 estudantes, o algoritmo hierárquico foi excluído por ser mais eficiente em conjuntos de dados mais reduzidos.

Posteriormente, foram efetuados testes à variação dos parâmetros dos algoritmos *K-means* e EM. Em ambos os algoritmos há um parâmetro comum, "*seed*". Esta variável consiste no número de instâncias que serão escolhidas aleatoriamente para formar os *clusters* inicialmente.

Variando este parâmetro, obtiveram-se os resultados ilustrados nos gráficos 4.2.3.1. Após todas estas avaliações, o algoritmo escolhido foi o *K-means* com o valor 6 (seis) para o parâmetro *seed*.

Outro parâmetro requerido pelo algoritmo, como foi referido anteriormente, é o número de grupos que se pretendem formar, que neste caso, foi escolhido o número 6 (seis) visto ser o número médio de alunos de cada grupo na unidade curricular Projeto FEUP.

4.2.3.2 Seleção dos dados

Para extrair dados do *Facebook* é necessário estar conectado a uma conta. Neste caso, foi usada a conta da autora (perfil *https://www.facebook.com/caterinecarvalho*), sendo que foram selecionados os 100 primeiros dados referentes a estudantes que atualmente frequentam a Faculdade de Engenharia da Universidade do Porto.

4.2.3.3 Resultados obtidos

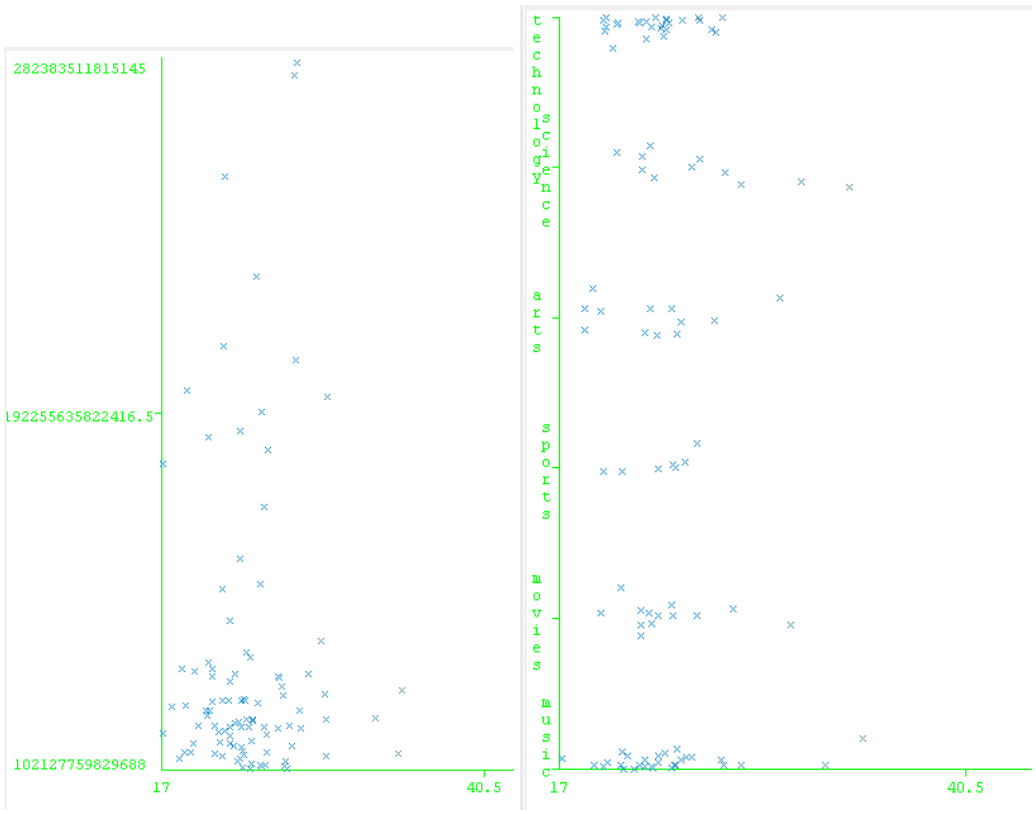
Os resultados obtidos são indicados em forma tabular, sendo que a primeira tabela indica os centroides escolhidos, isto é, o valor central de cada *cluster*, como se pode verificar na tabela 4.8, e o número de objetos em cada *cluster* segue a distribuição indicada na tabela 4.9.

```
1 Clustered Instances
2
3 0       14 ( 14%)
```

```

Cluster centroides:
                                                                                   Cluster#
Attribute       Full Data               0               1               2                 3               4               5
                    (100)            (14)            (21)             (7)              (14)            (30)            (14)
===========================================================================================================================
age                 23.89            22.5         22.5714         22.8571           26.2143         24.1667         24.8571
location            Porto           Trofa        Gondomar            Maia Vila Nova de Gaia      Matosinhos Povoa do Varzim
education 127792114553176 125412884869008 114518596991895 215555922307261   122514778258489 125274515613606 116871907867418
likes               music            arts      technology      technology            movies           music         science
```

Listing 4.8: Resultados obtidos para 6 grupos usando o algoritmo K-Means



(a) X: Idade, Y: Educação (b) X: Idade, Y: Interesses Pessoais

Figura 4.3: Gráficos bidimensionais de acordo com os eixos escolhidos

```
4  1        21 ( 21%)
5  2         7 (  7%)
6  3        14 ( 14%)
7  4        30 ( 30%)
8  5        14 ( 14%)
```

Listing 4.9: Distribuição dos objetos pelos *clusters*

Através da biblioteca Weka é também possível gerar gráficos bidimensionais com os resultados obtidos, como demonstrado na figura 4.3.

Se pretendido, através da interface Weka, é ainda possível guardar este modelo para aplicá-lo a outros conjuntos de dados.

4.2.4 Validação dos dados

Para uma validação correta e precisa dos resultados obtidos, seria necessário aplicar a metodologia aos estudantes acabados de ingressar na Faculdade de Engenharia da Universidade do Porto. Isto deve-se ao facto de a única forma de garantir que os agrupamentos estão a ser os corretos é avaliar o grau de satisfação de cada aluno, tendo em conta única e exclusivamente os fatores

usados, pois no decorrer da unidade curricular, inevitavelmente vão-se formando amizades entre os alunos que impedem que esta avaliação de satisfação seja precisa e imparcial.

4.3 Resumo

A combinação das tecnologias e frameworks usadas permitiu que o trabalho realizado cumprisse com os objetivos propostos. A metodologia implementada permite, através de informações extraidas de uma rede social, criar grupos de trabalho tentando equilibrar estes grupos a nível das caraterísticas dos estudantes tornando uma componente da avaliação dos trabalhos mais justa.

Apesar de os grupos serem formados usando características como interesses pessoais e idade, isto não basta para garantir que os grupos são equilibrados porque há fatores como a personalidade, atitudes e inclusivé o meio em que o estudante se insere que influenciam a dinâmica de um grupo, nunca sendo possível afirmar que há uma metodologia que tem sucesso absoluto garantindo o equilíbrio de todos os grupos.

Capítulo 5

Conclusões e Trabalho Futuro

No capítulo 2 aprenderam-se os conceitos essenciais sobre formação automática de grupos, EDM e DM e de que forma estes problemas têm sido estudados ao longo dos últimos anos. Este capítulo foi dividido nas três áreas principais: EDM, Facebook (como rede social usada) e os Algoritmos de Clustering. O capítulo 3 aprofundou o conhecimento do problema a resolver. O problema foi dividido em sub-problemas de modo a explicar as dificuldades e detalhes de cada um destes. Por último, no capítulo 4 foram especificados todos os detalhes da implementação, desde as tecnologias às frameworks usadas, bem como exemplos de código para resolução de problemas.

5.1 Satisfação dos Objectivos

Dos objetivos definidos (secção 1.2), conclui-se o seguinte:

A formação automática de grupos para projetos estudantis é um problema bastante estudado. De acordo com a literatura revista no capítulo 2, identifica-se que este problema é bastante estudado. São vários os métodos possíveis para obter uma divisão de alunos em grupos sendo que estas dependem dos critérios e padrões que se pretendem aplicar.

Identificação dos dados relevantes para a formação de grupos no Projeto FEUP. A unidade curricular Projeto FEUP não foca proincipalmente nas competências técnicas dos estudantes mas sim na sua integração com os outros estudantes, com a nova faculdade e novos métodos de trabalho. Deste modo, foi conseguido com sucesso a escolha dos fatores principais que influenciam a dinâmica de trabalho do grupo no projeto.

Extração dos dados do Facebook e formação dos grupos. Apesar de os dados provenientes da Faculdade de Engenharia da Universidade do Porto não terem sido usados, a extração destes dados foi feita pelo Facebook. Embora o Facebook seja uma vasta rede de dados, usando a sua linguagem e métodos próprios, estes podem ser obtidos e analisados. Após os dados serem recolhidos, foi necessário efetuar vários testes de modo a escolher o algoritmo a ser

utilizado e os seus parâmetros configuráveis de modo a obter a precisão máxima na formação dos grupos.

Concluindo, podemos afirmar que é possível formar de forma automática grupos de estudantes para projetos estudantis obtendo os dados através de bases de dados não convencionais.

5.2 Trabalho Futuro

Existem várias possibilidades para extender o trabalho desenvolvido, sendo que duas delas passam pela criação de um novo algoritmo e a implementação da metodologia desenvolvida na unidade curricular Projeto FEUP.

O novo algoritmo deveria tomar em consideração na computação da função distância o tipo de parâmetros usados, isto é, um melhoramento da função distância entre dois objetos, tornando o agrupamento mais preciso.

A nível da implementação na unidade curricular Projeto FEUP, esta seria feita de modo a avaliar se a formação atual dos grupos satisfaz a condição de equilibrar os grupos tornando uma avaliação mais justa. Esta implementação deveria ser acompanhada de questionários aos alunos no fim do projeto avaliando o seu grau de satisfação relativo ao grupo em que estava inserido, bem como o seu grau de satisfação da comparação dos níveis dos grupos (se considera o seu grupo equilibrado em relação aos restantes).

Referências

[Abb08] OA Abbas. Comparisons Between Data Clustering Algorithms. *The International Arab Journal of Information*, 2008.

[BDC09] José Borges, Teresa Galvão Dias e João Falcão E Cunha. A new group-formation method for student projects. *European Journal of Engineering Education*, 34(6):573–585, December 2009.

[Bel] Mary Bellis. Who Invented Facebook? http://inventors.about.com/od/fstartinventions/a/Facebook.htm.

[BSA01] Donald R Bacon, Kim A Stewart e Elizabeth Scott Anderson. Methods of Assigning Players to Teams: A Review and Novel Approach. *Simulation & Gaming*, 32(1):6–17, 2001.

[BY09] Ryan S J D Baker e Kalina Yacef. The State of Educational Data Mining in 2009 : A Review and Future Visions. *Review Literature And Arts Of The Americas*, 2009.

[CP07] Christos E. Christodoulopoulos e Kyparisia a. Papanikolaou. A Group Formation Tool in an E-Learning Context. *19th IEEE International Conference on Tools with Artificial Intelligence(ICTAI 2007)*, pages 117–123, October 2007.

[CVNM07] Félix Castro, Alfredo Vellido, Àngela Nebot e Francisco Mugica. Applying Data Mining Techniques to e-Learning Problems. *Evolution of Teaching and Learning Paradigms in Intelligent Environment*, 62:183–221, 2007.

[Del01] Nathan J Delson. Increasing Team Motivation in Engineering Design Courses. *International Journal of Engineering Education*, 17(4/5):359–366, 2001.

[Fac11a] Facebook. Authentication, 2011.

[Fac11b] Facebook. Graph API, 2011.

[Fac12] Facebook. Facebook Query Language (FQL), 2012.

[FGM+99] R Fielding, J Gettys, J Mogul, H Frystyk, L Masinter, P Leach e T Berners-Lee. Rfc2616: Hypertext transfer protocol – http/1.1. *RFC Editor United States*, 1999.

[HÏ0] Roland Hübscher. Assigning Students to Groups Using General and Context-Specific Criteria. *IEEE Transactions on Learning Technologies*, 3(3):178–189, 2010.

[Han04] Margo Hanna. Data mining in the e-learning domain. *Campus-Wide Information Systems*, 21(1):29–34, 2004.

REFERÊNCIAS

[HjHAk+09] Sharon Hardof-jaffe, Arnon Hershkovitz, Hama Abu-kishk, Ofer Bergman e Rafi Nachmias. How do Students Organize Personal Information Spaces ? *Communication*, pages 250–258, 2009.

[HLRH11a] E Hammer-Lahav, D Recordon e D Hardt. The oauth 2.0 authorization protocol. *IEEE Internet Computing*, 8(1):1–47, 2011.

[HLRH11b] E Hammer-Lahav, D Recordon e D Hardt. The OAuth 2.0 Authorization Protocol. *IEEE Internet Computing*, 8(1):1–47, 2011.

[JJS91] D W Johnson, R T Johnson e K A Smith. *Active learning: Cooperation in the College Classroom*. Interaction Book Company, 1991.

[KCSL08] Kenneth R Koedinger, Kyle Cunningham, Alida Skogsholm e Brett Leber. An open repository and analysis tools for fine-grained , longitudinal learner data. *Area*, pages 157–166, 2008.

[MB06] Jack Mostow e Joseph Beck. Some useful tactics to modify, map and mine data from intelligent tutors. *Natural Language Engineering*, 12(02):195–208, May 2006.

[Mer05] Agathe Merceron. Educational data mining: a case study. *Proceeding of the 2005 conference on Artificial*, pages 1–8, 2005.

[ODM] Asma Ounnas, Hugh C Davis e David E Millard. Semantic Modeling for Group Formation. *Networks*, pages 2–5.

[Rag05] NR Srinivasa Raghavan. Data mining in e-commerce: A survey. *Sadhana*, 30(June):275–289, 2005.

[Red01] Michael A Redmond. A Computer Program to Aid Assignment of Student Project Groups. *Symposium A Quarterly Journal In Modern Foreign Literatures*, pages 134–138, 2001.

[Res] RestFB. A simple Java interface for the Facebook Graph.

[RV04] C Romero e S Ventura. Knowledge discovery with genetic programming for providing feedback to courseware authors. *User Modeling and User-Adapted*, 14(5):425–464, 2004.

[RV07] C. Romero e S. Ventura. Educational data mining: A survey from 1995 to 2005. *Expert Systems with Applications*, 33(1):135–146, July 2007.

[RV10] Cristóbal Romero e Sebastián Ventura. Educational Data Mining : A Review of the State of the Art. 40(6):601–618, 2010.

[RVPB10] Cristóbal Romero, Sebastián Ventura, M Pechenizkiy e Ryan Baker. *Handbook of Educational Data Mining*. New York: Taylor & Francis, 2010.

[Smi89] K A Smith. The craft of teaching cooperative learning: an active learning strategy, 1989.

[TB11] Published Taylor e Paul Blowers. SELF-ASSESSMENTS BALANCED GROUP STUDENT SKILL TO GET FOR GROUPS PROJECTS individuals who will work suc. *College Teaching*, 51(3):106–110, 2011.

[TF07] Carlos Miguel Tobar e Ricardo Luís De Freitas. A Support Tool for Student Group Definition. *37th ASEE/IEEE Frontiers in Education Conference*, pages 7–8, 2007.

[Vis12] Visco. La mappa dei social network nel mondo – giugno 2012. http://vincos.it/2012/06/11/la-mappa-dei-social-network-nel-mondo-giugno-2012/, 2012.

[Wek] Weka. ARFF (book version).

[Wik12] Wikipedia. Lista de concelhos por NUTS, 2012.

[WP01] M. Wessner e H-R. Pfister. Group formation in computer-supported collaborative learning. *Group*, pages 24–31, 2001.