Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Sistema de recomendação dos amigos na rede social online baseado em Máquinas de Vetores Suporte

Yang Liu

Dissertação apresentada como requisito parcial
para conclusão do Mestrado do Programa de
Pós-Graduação em Informática

Orientador
Prof. Dr. Li Weigang

Brasília
2014

Universidade de Brasília — UnB
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Pós-Graduação em Informática

Coordenadora: Prof.ª Dr.ª Alba Cristina Magalhaes Alves de Melo

Banca examinadora composta por:

Prof. Dr. Li Weigang (Orientador) — CIC/UnB
Prof. Dr. Camilo Chang Dórea — CIC/UnB
Prof. Dr. João Batista Camargo Júnior — POLI/USP

CIP — Catalogação Internacional na Publicação

Liu, Yang.

Sistema de recomendação dos amigos na rede social online baseado em Máquinas de Vetores Suporte / Yang Liu. Brasília : UnB, 2014.
90 p. : il. ; 29,5 cm.

Dissertação (Mestrado) — Universidade de Brasília, Brasília, 2014.

1. Microblog, 2. Redes Sociais Online, 3. Máquina de Vetores de Suporte, 4. Sistema de Recomendação, 5. Sistema de Recomendação Baseada em Conteúdo.

CDU 004

Endereço: Universidade de Brasília
Campus Universitário Darcy Ribeiro — Asa Norte
CEP 70910-900
Brasília–DF — Brasil

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Sistema de recomendação dos amigos na rede social online baseado em Máquinas de Vetores Suporte

Yang Liu

Dissertação apresentada como requisito parcial para
conclusão do Mestrado do Programa de Pós-Graduação em Informática

Prof. Dr. Li Weigang (Orientador)
CIC/UnB

Prof. Dr. Camilo Chang Dórea
CIC/UnB

Prof. Dr. João Batista Camargo Júnior
POLI/USP

Prof.ª Dr.ª Alba Cristina Magalhaes Alves de Melo
Coordenadora do Pós-Graduação em Informática

Brasília, 13 de março de 2014

Dedicatória

Dedico ao meu pai Tong Liu e à minha mãe Jiangling Yang ,que, mesmo estando longe, nunca deixaram de me apoiar.

À minha esposa Zhaodi Lei pela seus compreensão,amor e carinho.

Ao meu filho Xiao Leilei pelo me dê o poder.

Agradecimentos

Quero agradecer a todas as pessoas que me apoiaram desde o primeiro momento que tomei a decisão de começar o mestrado, quando cheguei nesse novo país, e no decorrer dos estudos até a culminação do mestrado. Sem o apoio deles nada do que consegui ate hoje houvesse sido possível. Ao meu orientador Prof.Li Weigang, pelas aulas, pelas sugestões, pelos conselhos e dicas de pesquisa, pelo material emprestado, pela paciência que teve comigo, pela participação e pela ajuda incondicional, quem com seus conhecimentos e experiência soube me encaminhar no mestrado. Torço pra que esta parceria continue por muitos anos. Aos meus amigos Jianya Zheng, Déborah Mendes, Tiago Araújo, Helard Becerra, as suas amizades foram muito importantes nesta etapa da minha vida.

A decisão de realizar os estudos fora do país de origem não é simples, por isso quero agradecer minha família pelo seu apoio incondicional, aos meus pais Tong e Jiangling e minha esposa Zhaodi.

A todos os professores do Departamento de Ciência da Computação, que foram tão importantes na minha vida acadêmica e no desenvolvimento deste trabalho, Aos professor Camilo Chang Dórea e o professor João Batista Camargo Júnior que são parte da minha banca, pela sua presença, suas sugestões e contribuições para com meu trabalho. Aos professores Alba, Mauricio Ayala; Maria Emilia; Marcelo Ladeira, os quais tive o prazer de assistir suas aulas.

Agradeço também aos meus colegas do mestrado com quem compartilhei muito durante meus estudos, de maneira especial a todos que me ajudaram nos momentos de duvidas e dificuldades, muito obrigado Ruben Cruz, Harley Vera, Toni Serrano, Stephanie Alvarez, Henrique Freitas, Ariane Alvez,Lucas Araujo, Jonathan Alis. Agradeço também aos meus colegas e amigos do (TransLab).

Resumo

O rápido desenvolvimento da tecnologia da Internet trouxe-nos para a era da explosão de informações, enquanto a massa de informações por um lado, torna difícil selecionar as mais interessantes. Por outro lado, também muitas delas são perdidas na rede de informação, pois existem "informações secretas", não permitindo o acesso aos usuários em geral. O Sistema de Recomendação(RS) é atualmente um esquema mais eficiente para resolver o problema recente de sobrecarga de informações. A recomendação é amplamente utilizada em Redes Sociais Online(como Twitter, Weibo e outros Microblogs), neste trabalho é utilizado o método de Máquina de Vetores de Suporte(SVM) para aplicar recomendação de amigos.

A dissertação propõe uma idéia que combina a teoria e os atributos de Microblog SVM para realizar a recomendação de amigos. Além disso implementá-lo como um sistema recomendado para aumentar a aceitação do usuário no microblog.

Os experimentos mostraram que o modelo SVM proposto apresenta um desempenho eficiente e boa exatidão na recomendação de amigos nas redes sociais. O resultado do SVM é 72% melhor que os métodos usados para comparação: os algoritmos Naïve Bayes e Random Forest, tendo sido considerados diferentes tamanhos de amostras para testar a eficiência e o desempenho destes modelos. O resultado mostrou que o algoritmo SVM é melhor para amostras de diversos tamanhos.

Palavras-chave: Microblog, Redes Sociais Online, Máquina de Vetores de Suporte, Sistema de Recomendação, Sistema de Recomendação Baseada em Conteúdo.

Abstract

The rapid growth of internet technology brought us to the era of the rapid diffusion of information. Nevertheless, the large quantity of information makes it difficult to find interesting information and therefore, much of it is lost in the information network due to "secret information", not permitting the access to the general public. Recommendation systems (RS) are nowadays the most efficient tools to solve the recent problem of information overload. RS is already widely used in Online Social Networks(such as Twitter,Weibo and other Microblogs). In this research, Support Vector Machines (SVM) method is applied in the recommendation of friends.

The dissertation proposes an idea which combining the SVM theory and attributes of Microblog to realize the recommendation of friends. Furthemore implement it as a recommended system to increase the acceptance of user no microblog.

The experiments showed that the proposed SVM model presents an efficient performance and good accuracy on the recommendation of friends in social networks. The result of the SVM is 72% better than the methods used for comparison: the algorithms Random Forest and Naive Bayes. Different sample sizes were considered separately to test the efficiency and performance of these models. These results showed that the SVM algorithm is better for samples of different sizes.

Keywords: Content-Based Recommendation Systems, Online Social Network, Recommendation Systems, Microblog, Support Vector Machines.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de abreviaturas

ACM-*Association for Computing Machinery*

CF - *Collaborative Filtering*(Filtragem Colaborativa)

CR - *Communication Reciprocal*

LD - *Lagrange Duality*(Dualidade de Lagrange)

LM - *Lagrande Multiplier* (Multiplicador de Lagrande)

KKT - *Karush-Kuhn-Tucker*

KDD - *Knowledge Discovery and Data Mining*

MMH - *Maximum-Margin Hyperplane*(Hiperplano Máximo Marginal)

MSN - *Multidimensional Social Networks*(Redes Sociais Multidimensionais)

MSS - *Multimedia Sharing System*(Sistema de Compartilhamento em Multimídia)

NB - *Naive Bayes*

OSN - *Oline Social Network*(Rede Social Online)

PARC- *Palo Alto Research Center*

QP - *Quadratic Programming*(Programação Quadrática)

RBF - *Radial Basis Function*(Função Rede Neural)

RH - *Hybrid Becommendation*(Recomendação Híbrida)

RS - *Recomendation System*(Sistema de Recomendação)

RSCB - *Recomendation System Content-Based*(Sistema de Recomendação Baseado em Conteúdo)

RSKB - *Recomendation System Knowledge-Based*(Sistema de Recomendação Baseado em Conhecimento)

SVM - *Support Vector Machine*(Máquina de Vetores de Suporte)

SNA - *Social Network Analysis*(Análise de Redes Sociais)

WCH - *Weight Categorial Hierarchical*(Ategorização Hierarquizada Baseada em Pesos)

Capítulo 1

Introdução

Com o rápido desenvolvimento da Internet, o número de acessos à *World Wide Web* apresentou uma tendência de crescimento exponencial. O avanço da tecnologia da Internet fez com que muitas informações tornassem facilmente acessíveis, por exemplo, o *Netflix*[2] possui milhares de filmes, a *Amazon*[3] milhares de livros, o *Del.icio.us*[4] coleta mais de um bilhão de páginas na internet, com tantas informações e impossível fazer com que o usuário tenha uma percepção apurada de tudo, e é ainda difícil fazer com que se encontre que seja do interesses deles.

O problema chave é a busca das informações úteis, pois o algoritmo tradicional de busca, o que é mais utilizado atualmente, só pode apresentar o mesmo conteúdo para todos os usuários e não pode fornecer um serviço individualizado para diferentes interesses. O surgimento do Sistema de Recomendação(RS) mudou esta situação.

A explosão de informações fez com que a utilização das mesmas decaísse, este fenômeno é chamado de sobrecarga de informações. Para resolver este problema, foi sugerido um esquema de recomendação personalizado através da interface da Internet. Dentro desta recomendação, inclui-se a busca personalizada, sendo esta considerada uma das ferramentas mais eficazes para resolver o atual problema de sobrecarga de informações. Pesquisas feitas pelos Sistemas de Recomendação são capazes de substituir os usuários, avaliando os produtos que os mesmos não conheceram ainda[5][6]. Esses produtos incluem livros, filmes, CDs, páginas da web, e até mesmo catálogos de hotéis, música, pintura, entre outros; sendo esses produtos antes desconhecidos pelos usuários, se tornam conhecidos para os usuários que possuam interesse em tais produtos.

Os Sistemas de Recomendação surgiram no final da década de 1990, sendo proposto como um conceito independente[7]. Um rápido desenvolvimento veio junto da Web 2.0. Na prática, quando recomendados em um sistema de recomendação, os produtos podem entrar na faixa dos milhões. Na *Amazon*[3], *eBay*[8] e no *Youtube*[9] entre outros, o número de usuários é enorme. Sistemas de Recomendação precisos e eficientes aumentam o potencial de consumo dos usuários, o RS os oferece mais serviços personalizados.

O RS não é apenas uma ferramenta de marketing comercial num ambiente cada vez mais competitivo, o mais importante é aproximar os usuários numa rede de contatos cada vez mais dinâmica, acarretando na fidelidade dos usuários nos sites que frequentam. Tais sistemas geram enormes interesses comerciais no domínio do comércio eletrônico. De fato, há muitos índices de avaliação que podem medir o desempenho do algoritmo de recomendação para melhorar o algoritmo em múltiplos ângulos. Naturalmente, indepen-

dentemente do ângulo em que as melhorias começarem, é a partir da arquitetura geral do RS que se terá compreensão completa.

Com a ascensão do *Facebook*[10], *Twitter*[11] e *Youtube*[9], as redes sociais se tornaram cada vez mais importantes para nossas vidas. O RS surge como uma aplicação da tecnologia, realizando um papel importante na recomendação de amigos. O Sistema de Recomendação pode ajudar os usuários a encontrar pessoas que possam os interessar, talvez a pessoa que ele encontra todo dia, mas não sabe o seu nome, talvez a sua pessoa favorita, talvez alguém que gostaria de conhecer sem motivo algum. O Sistema de Recomendação auxilia a aumentar a rede de pessoas, ao mesmo tempo que obtém informações úteis para os usuários. A recomendação de usuários afeta a lealdade dos seus usuários e indiretamente, o investimento em publicidade e renda. O método de recomendações predominante é o de Filtragem Colaborativa(CF) e de Sistema de Recomendação Baseado em Conteúdo (RSCB), Baseado em conhecimento(RSKB) e Recomendação Híbrida(RH). Esta pesquisa se aproveita da recente ascensão do método de apoio em Máquina de Vetores de Suporte(SVM) para aumentar a precisão da recomendação de amigos em Microblog.

1.1 Motivação

Suponha que um aplicativo de rede social de *Microblog* possua uma grande quantidade de informações (brincadeiras, piadas, conselhos, notícias, imagens, vídeo). Usuários de *Microblog* em aplicações sofrem por não saber por onde começar, e não recebem nenhuma ajuda de como encontrar amigos e informações. Se o usuário não encontrar nenhuma informação que o interesse, em breve perderá o interesse. Então um seleto sistema de recomendação é muito importante. Isso se difere da publicidade na internet, pois ele pode ajudá-lo a encontrar um amigo de longa data que o usuário não manteve contato. Ademais, esse sistema pode auxiliá-lo a encontrar informações relevantes na sua área de interesse e expandir os círculos sociais nas redes sociais.

Se o RS recomendar muitos amigos para um usuário, mas este usuário não conhecer ou não se interessar pela maioria deles, isso irá desempenhar um papel oposto. Melhorar a precisão de cada RS deve resolver o problema. Atualmente, o *Twitter* ou *Microblog* possui um RS muito simples, ele é capaz de recomendar apenas seguidores indiretos. Por exemplo, se muitos dos seguidores de um determinado usuário seguem uma pessoa, essa pessoa será recomendada para o usuário. É uma forma relevante de se descobrir amigos, porém esse sistema supre apenas uma parte ínfima do total de amigos recomendáveis. O apelo individual das pessoas é um campo de conhecimento complicado na vida real. Há muitos fatores que levam Microblogs a fazer eventuais previsões imprecisas, essencialmente deve-se resolver o problema de escolher os métodos adequados e algoritmos.

Com isto em mente, nosso grupo de pesquisa de redes sociais deseja propor o embasamento teórico a ser utilizado na implementação do sistema de recomendação baseado na Máquina de Vetores de Suporte. Uma importante motivação para que esta estratégia seja explorada é o fato de que ela não foi ainda atacada por quase nenhuma pesquisa sobre o sistema de recomendação de amigos baseado em SVM. A teoria da SVM foi utilizado em várias áreas incluindo o reconhecimento de padrões, processamento de imagens, sendo estes atributos amplamente reconhecidos pela comunidade científica. Dito isso, temos como objeto de estudo a eficiência do campo de estudos em sistemas de recomendação.

1.2 Objetivos

Este trabalho tem por objetivo estudar os comportamentos dos usuários nas redes sociais online como *Tencent Weibo* (site chinês de *Microblog*, similar ao *Twitter*) e os fatores de influência no sistema de recomendação. Com abordagem desenvolvida na utilização de Máquina de Vetores de Suporte (SVM) para melhorar a predição das atividades de recomendação dos amigos nas redes sociais online. Os objetivos específicos são:

- Coletar informações dos usuários e construir um banco de dados para estruturar e minerar os dados, e em seguida, criar um perfil para definir os usuários incluindo todas as informações detalhadas.
- Determinar os atributos de classificação dos usuários para minerar seus perfis, estes atributos serão a entrada para o SVM, aumentando a precisão do sistema de recomendação de amigos.
- Verificar a validade dos resultados pela simulação do modelo proposto e analisar os resultados através da comparação com diversos métodos.

1.3 Metodologia

Os conceitos teóricos sob os quais o trabalho é desenvolvido consistem em duas principais técnicas: o SVM e RS. Esta seção expõe uma descrição sucinta dos métodos de pesquisa e desenvolvimento utilizadas.

1.3.1 Teoria da SVM

Nesse estudo, utilizaremos o classificador de Máquina de Vetores de Suporte para resolver a recomendação de amigos que é um desafio de classificação e previsão. A máquina de vetores de suporte é um conceito da Ciência da Computação para um conjunto de métodos de aprendizado supervisionado que analisam os dados e reconhecem padrões usados para classificação e análise de regressão.

O SVM padrão toma como entrada um conjunto de dados e prediz, para cada entrada dada, qual de duas possíveis classes a entrada faz parte, fazendo do SVM um classificador linear binário não probabilístico. Dado um conjunto de exemplos de treinamento, cada um marcado como pertencente a uma de duas categorias, um algoritmo de treinamento do SVM constrói um modelo que atribui novos exemplos a uma categoria ou outra. Um modelo SVM é um método para representar os exemplos como pontos no espaço, mapeados de maneira em que os exemplos de cada categoria sejam divididos por um espaço claro que seja tão amplo quanto possível.

Nesta pesquisa, é proposto um classificador de SVM padrão para prever a aceitação e rejeição do usuário com um conjunto de exemplos de treinamento minerado pelo perfil dos usuários. Essa é uma classificação de duas categorias; este resultado de previsão se dá em dois valores, 1 ou -1, 1 representando se o usuário aceita a recomendação e -1 representando se o usuário rejeitar a recomendação.

O SVM pertencem à uma família de classificadores lineares generalizados e podem ser interpretados como uma extensão do *Therceptron*. Eles também podem ser considerados

um caso especial de regularização de *Tirkhonov*. Uma propriedade especial é que eles podem simultaneamente minimizar o erro empírico de classificação e maximizar a margem geométrica, doravante, eles são conhecidos como classificadores marginais máximos.

1.3.2 Metodologia do sistema de recomendação

O SVM [12]pode ser utilizado em um dos métodos de sistemas de recomendação como uma teoria independente. Neste caso, o SVM foi classificado como um ramo dos métodos baseados em conteúdo que consiste em um dos métodos gerais em sistemas de recomendação[13]. No seu cerne, a recomendação baseada em conteúdo é fundada na disponibilidade de descrições de itens e um perfil que atribui importância para essas características.

Se considerar o exemplo de uma livraria, as características possíveis dos livros incluirão o gênero, o tópico específico ou o autor da obra. Semelhante às descrições de itens, perfis de usuários também podem ser extraídos e "aprendidos" a partir dos *feedbacks* de análise comportamental do usuário ou por perguntas explicitas sobre preferências e interesses.

1.3.3 Metodologia de pesquisa

Na realização desta pesquisa, utilizamos as metodologias definidas a seguir:

- Levantamento dos atributos do *Microblog*: Na etapa inicial, é feita a elicitação dos atributos do *Microblog* que influenciam o usuário a aceitar uma recomendação.
- Levantamento de informações: Consolidação de dados, análise e criação de um formato de entrada de dados que engendra o algoritmo SVM. Utilizamos a *toolbox* do *Linux* como ferramenta de coleta de dados e *Eclipse* para desenvolvimento do sistema.
- Definição da arquitetura: Com o escopo de atuação bem definido, o próximo passo é estabelecer a arquitetura do protótipo: são definidas a linguagem de implementação, as classes e estruturas de dados utilizadas.
- Implementação: A implementação do protótipo segue as definições anteriores. A metodologia estudada será aplicada no tratamento algorítmico do problema apresentado.
- Simulação: São definidos os casos de teste da simulação. A simulação é executada de acordo com um conjunto de dados reais, os dados são coletados e armazenados.
- Análise dos resultados: Após as simulações, os resultados obtidos são compilados e analisados minuciosamente, a fim de que se tenha uma estimativa da eficiência do modelo proposto. São levantados os êxitos e deficiências do modelo e é provida a avaliação final do protótipo.

1.4 Organização do Trabalho

Este trabalho está organizado da seguinte maneira.

No Capítulo 2 é exposta a fundamentação teórica dos conceitos explorados, fornece um breve estudo da Teoria da Máquina de Vetores de Suporte(SVM), um dos pilares teóricos deste trabalho e apresenta as etapas da SVM.

O Capítulo 3 elucida o contexto e estado da pesquisa atual relacionados à sistemas de recomendação e introduz os quatro métodos correntes principais.

O Capítulo 4 apresenta o estado da arte, pesquisas que utilizam sistemas de recomendação(RS) e SVM. Assinala as vantagens e desvantagens de cada trabalho e propõe soluções para resolver este problema.

O Capítulo 5 introduze os dados selecionados. Analisa os dados para decidir os atributos deste modelo e também mostra a estrutura do módulo dos atributos.

O Capítulo 6 estabelece o modelagem do sistema de recomendação, descreve detalhadamente como a modelagem funciona incluindo os módulos dos attributos e o módulo da SVM.

O Capítulo 7 realiza a modelagem com as ferramentos, apresenta o pré-processamento realizado no início da modelagem, além disso, exibe os algoritmos elaborados e seus pseudo-códigos.

A descrição dos procedimentos de teste por dois casos estão no Capítulo 8. Ainda neste capítulo, são averiguados os resultados obtidos e é elaborada a análise dos mesmos.

Por fim, o Capítulo 9 aborda os resultados frente aos objetivos propostos, com as considerações finais da implementação e modelagem. São introduzidas também algumas propostas de trabalhos futuros e eventuais melhorias ao modelo apresentado.

Capítulo 2

Fundamentação Teórica

Neste capítulo, introduzimos a teoria fundamental de classificação, em princípio, a Máquina de Vetores de Suporte (SVM). A tecnologia SVM é complexa e de difícil compreensão, portanto, o entendimento macroscópico dessa teoria é crucial. Primeiramente, expõe-se conceitos fundamentais acerca das funções que visa explicar rapidamente o princípio do SVM. Depois dessa breve descrição, iniciaremos uma exposição detalhada das técnicas empregadas na SVM.

2.1 Conceito Básico da Máquina de Vetores de Suporte

SVM usa um mapeamento não linear para transformar os dados de treinamento originais para uma dimensão superior[12]. Dentro desta nova dimensão, ele procura o hiperplano separador ideal e linear, isto é, uma "fronteira de decisão"separando as tuplas de uma classe da outra. Com o mapeamento não linear apropriado para uma dimensão suficientemente elevada, os dados de duas classes podem sempre ser separados pelo hiperplano. O SVM encontra este hiperplano usando vetores de suporte e as margens.

SVM têm atraído muita atenção ultimamente, o primeiro trabalho em máquinas de vetores de suporte foi apresentado em 1992 por Vladimir Vapnik [14] e colegas Bernhard Boser e Isabelle Guyon, porém as bases para SVMs tem sido estudadas desde os anos 1960. Embora o tempo de formação de mesmo as mais rápidas SVM pode ser extremamente lento, elas são altamente precisos, devido à sua capacidade para modelar complexos limites de decisão não-lineares. Elas são muito menos propensas a sobreajustes que outros métodos. SVM podem ser usadas para a previsão, bem como a classificação. Elas foram aplicadas a uma série de áreas, incluindo o reconhecimento manuscrito de dígitos, reconhecimento de objetos e identificação do locutor, bem como testes de previsão de séries temporais de referência.

2.1.1 Dados linearmente separáveis

Para explicar o mistério de SVM, vamos, em primeiro lugar olhar o caso mais simples, o problema de duas classes, onde as classes são linearmente separáveis. Seja D o conjunto de dados dado como $(X_1, y_1), (X_2, y_2), \cdots, (X_{|D|}, y_{|D|})$, onde X_i é o conjunto de treinamento de tuplas com a variável y_i de classe associada, Cada y_i pode tomar um de dois valores,

+1 ou -1,ou seja, $y_i \in \{+1, -1\}$. Para auxiliar na visualização, consideramos um exemplo com base em dois atributos de entrada, eixo A_1 e A_2 tal como mostrado na Figura 2.1.

A partir do gráfico, vemos que os dados 2D são linearmente separáveis, porque uma linha reta pode ser traçada para separar todas as tuplas de classe +1 e todas as tuplas de classe -1. Existe um número infinito de linhas de separação que podem ser utilizadas. Queremos encontrar a melhor possível, isto é, aquela que esperamos ter o menor erro de classificação em tuplas inéditas. Note que, se os nossos dados fossem 3D, precisaríamos encontrar o melhor plano separado. Generalizando para n dimensões, queremos encontrar o melhor hiperplano. O termo hiperplano será usado para se referir à fronteira de decisão que estamos buscando, independentemente do número de atributos de entrada.

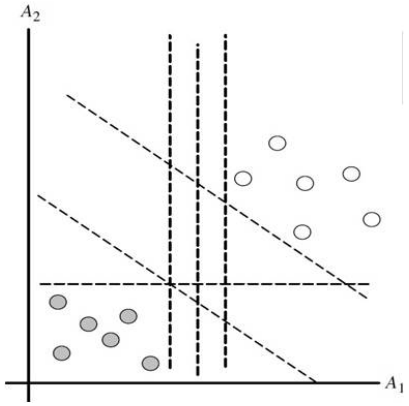

Figura 2.1: Exemplo com dois atributos.

O SVM aborda esse problema, procurando o hiperplano máximo marginal (MMH). Considere a Figura 2.2, que mostra dois hiperplanos separados e suas possíveis margens associadas. Ambos os hiperplanos podem classificar corretamente todos as tuplas de dados. Intuitivamente esperamos que o hiperplano com a margem maior seja mais preciso em classificar futuras tuplas de dados do que o hiperplano com margem menor. É por isso que, as pesquisas para o hiperplano do SVM procuram encontrar a maior margem, isto é, o hiperplano máximo marginal.

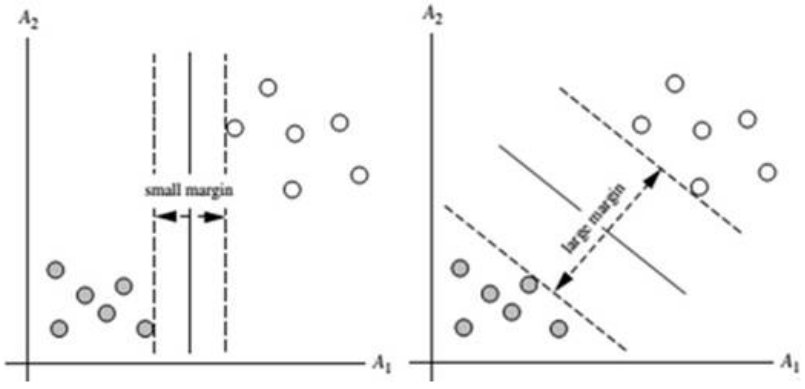

Figura 2.2: (a) Dados com uma pequena margem. (b) Dados com uma grande margem.

Chegando a uma definição informal de margem, podemos dizer que a menor distância a partir de um hiperplano a um lado da sua margem é igual à distância mais curta entre o hiperplano para o outro lado da sua margem, onde os "lados"da margem são paralelos ao hiperplano. A distância do MMH é mais curta entre o hiperplano e a tupla mais próxima na formação de ambas as classes. Um hiperplano separador pode ser escrito como na Equação 2.1:

$$W \cdot X + b = 0 \tag{2.1}$$

em que W é um vector de peso, ou seja, $W = \{w_1, w_2, \cdots, w_n\}$; n é o número de atributos, e b é um escalar, muitas vezes referido como um viés. Para ajudar na visualização, vamos considerar dois atributos de entrada,A_1 *e* A_2, como na Figura 2.2. Tuplas de treinamento são 2D, por exemplo, $X = (x_1, x_2)$, onde x_1 e x_2 são os valores de atributos A_1 *e* A_2 respectivamente, para X. Se pensarmos b como um peso adicional, w_0, podemos reescrever o hiperplano separador acima como na Equação 2.2:

$$w_0 + w_1 x_1 + w_2 x_2 = 0 \tag{2.2}$$

Assim, qualquer ponto que se encontra acima do hiperplano separador satisfaz a Equação 2.3:

$$w_0 + w_1 x_1 + w_2 x_2 > 0 \tag{2.3}$$

Da mesma forma, qualquer ponto que se situa abaixo do hiperplano separador satisfaz a Equação 2.4:

$$w_0 + w_1 x_1 + w_2 x_2 < 0 \tag{2.4}$$

Os pesos podem ser ajustados de modo que os hiperplanos definam os "lados"da margem, que pode ser escrita como a Equação 2.5:

$$\begin{aligned} H_1 &: w_0 + w_1 x_1 + w_2 x_2 \geq 1 \quad para \quad y_i = +1 \\ H_2 &: w_0 + w_1 x_1 + w_2 x_2 \leq -1 \quad para \quad y_i = -1 \end{aligned} \tag{2.5}$$

Isto é, qualquer tupla que cai sobre ou acima de H_1 pertence à classe 1, e qualquer tupla situada em ou abaixo de H_2 pertence à classe -1. Combinando as duas desigualdades da Equação 2.6, obtemos:

$$y_i(w_0 + w_1 x_1 + w_2 x_2) \geq 1, \forall i \tag{2.6}$$

Quaisquer formação de tuplas que caem sobre hiperplanos H_1 ou H_2 (isto é, as "partes"que definem a margem) e satisfazem a Equação 2.6 são chamados de vetores de suporte. Ou seja, eles estão igualmente perto do MMH. Na Figura 2.3 os vetores de suporte são mostrados com uma margem mais espessa. Hiperplano do SVM encontra o máximo de separação, isto é, um com a distância máxima entre as tuplas de formação mais próximas. Os vetores de suporte são mostrados com uma margem mais espessa.

Usando alguns artifícios de matemática, podemos reescrever a Equação 2.6, de modo que torna-se o que é conhecido como um restrito (convexo) problema de otimização quadrática, os artifícios envolvem reescrever a Equação 2.6, utilizando uma formulação de Lagrange e resolvendo a solução usando as condições Karush-Kuhn-Tucker (KKT).

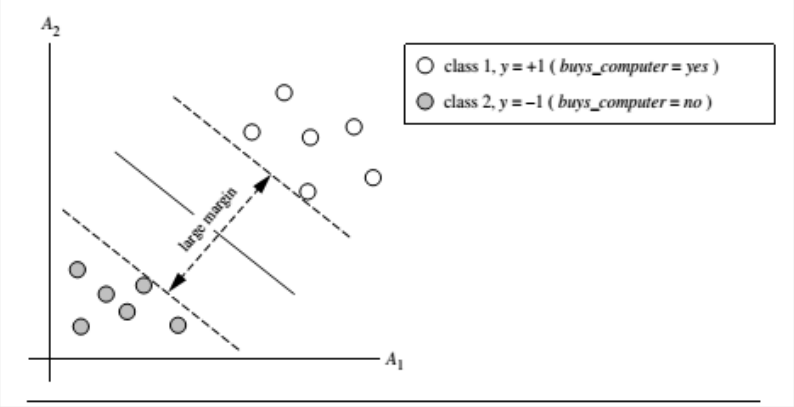

Figura 2.3: Vetores de Suporte.

Uma vez que se tem uma máquina de vetor de suporte treinada, com base na formulação de Lagrange mencionada acima, o MMH pode ser reescrita como a fronteira de decisão.

$$d(X^T) = \sum_{i=1}^{l} y_i \alpha_i X_i X^T + b_0 \tag{2.7}$$

onde y_i é o rótulo de classe do vetor de suporte X_i; X^T é uma tupla de teste; α_i e b_0 são parâmetros numéricos que foram determinados automaticamente pela otimização ou algoritmo SVM acima, e l é o número de vetores de suporte.

Dada uma tupla teste, X^T, substitui-a na Equação 2.7 e, em seguida, verifica-se o sinal do resultado. Isso nos diz de que lado do hiperplano a tupla teste cai. Se o sinal for positivo, então X^T cai acima ou no MMH, e assim o SVM prevê que X^T pertence à classe $+1$. Se o sinal é negativo, então X^T situa-se em ou abaixo da MMH e a previsão é classe -1.

A complexidade do classificador aprendido é caracterizada pelo número de vetores de suporte, em vez da dimensão dos dados. Assim, SVM tende a ser menos propensa a sobreajuste que alguns outros métodos. Se todas as outras tuplas de treinamento forem removidas e treinamento for repetido, o mesmo hiperplano separador seria encontrado.

2.1.2 Caso em que os dados são linearmente inseparáveis

Compreendemos sobre SVM lineares para a classificação de dados linearmente separáveis, mas existem casos onde os dados não são linearmente separáveis, como na Figura 2.4. Em tais casos, nenhuma linha capaz de separar as classes pode ser encontrada. As SVM lineares que estudamos não seriam capazes de encontrar uma solução viável neste caso. A Figura 2.1 mostrou um caso 2D simples, mostrando dados linearmente inseparáveis. Ao contrário dos dados separáveis lineares da Figura 2.1, neste caso, não é possível traçar uma linha reta para separar as classes. Em vez disso, a fronteira de decisão é não linear.

A abordagem descrita para SVM não lineares pode ser estendida para criar SVM não lineares para a classificação de dados linearmente inseparáveis. Tais SVM são capazes de detectar limites de decisão lineares no espaço de entrada.

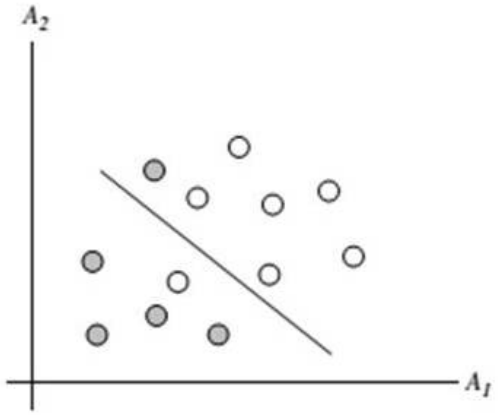

Figura 2.4: O caso de dados linearmente inseparáveis.

Obtemos uma SVM não linear, estendendo a abordagem para SVM lineares como se segue. Existem duas etapas principais: no primeiro passo, se transformam os dados de entrada originais para um espaço dimensional superior usando um mapeamento não linear. Vários mapeamentos não lineares comuns podem ser utilizados neste passo. Uma vez que os dados foram transformados para o novo espaço superior, obtém-se um hiperplano separador linear no novo espaço. Obtém-se um problema de otimização quadrática que pode ser resolvido usando a formulação SVM linear. O hiperplano máximo marginal encontrado no novo espaço corresponde a uma separação de hipersuperfícies não-lineares no espaço original.

Até agora, descrevemos SVM lineares e não lineares para classificação binária (ou seja, duas classes). Classificadores SVM podem ser combinados para o caso com multiclasses. Uma abordagem simples e eficaz, dado m classes e m treinos classificadores, um para cada classe (onde o classificador j aprende a retornar um valor positivo para a classe j e um valor negativo para o resto). Uma tupla de teste é atribuída à classe correspondente a maior distância positiva.

Além da classificação, SVM também pode ser utilizado para a regressão linear e não linear. Neste caso, ao invés vez de aprender a prever rótulos de classe discretos ($y_i \in \{+1, -1\}$ como o acima), SVM para tentativa de regressão para saber a relação de entrada-saída entre tuplas de entrada de treinamento, X_i, e as suas saídas contínuas com valores correspondentes,$y_i \in R$. Uma abordagem semelhante às SVM para a classificação é seguida. Parâmetros especificados pelo usuário adicionais são necessários. Um dos principais objetivos de pesquisa sobre SVM é melhorar a velocidade de treinamento e de testes para que SVM possam tornar-se uma opção mais viável para grandes conjuntos de dados (por exemplo, de milhões de vetores de suporte). Outras questões incluem a determinação do melhor kernel para um determinado conjunto de dados e encontrar métodos mais eficientes para o caso multiclasse.

2.2 Modelo de Processo do SVM

Descrevemos brevemente o SVM na Seção 2.1. Agora uma descrição detalhada com cada passo do processo do SVM será apresentada.

A Figura 2.5 apresenta a modelagem do SVM, exibindo um diagrama sobre os princípios e processos da classificação SVM, tem-se como objetivo o entendimento intuitivo.Este gráfico descreve passo a passo como obter o Hiperplano usa pela forma de raciocínio inverso.A subseção baixo vai explicar cada passo.

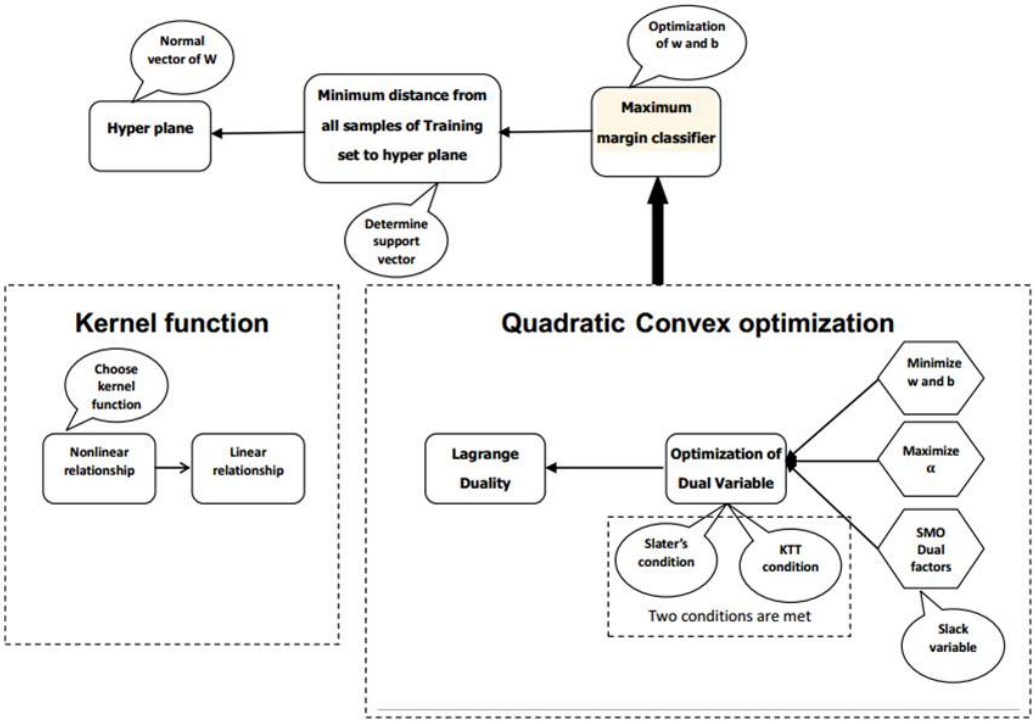

Figura 2.5: Diagrama sobre os princípios e processos da classificação SVM.

Para determinar o hiperplano, primeiramente, é necessário determinar os dois parâmetros w e b em $f(x) = w \cdot x + b$ (w, x representam o produto interior de w e x). w é o vetor normal, b é o intersecção; depois, a mudança na função de classificação se dá no problema de w, b. Após isso, o problema é traduzido na função de classificação dentro da otimização de w, b.

2.2.1 Margem geométrica

Considerando a Figura 2.6, a projeção vertical do ponto x no hiperplano é x_0, pois o w é o vetor perpendicular do hiperplano, γ é a distância do ponto até o hiperplano, dito isso, temos a Equação 2.8:

$$x = x_0 + \gamma \frac{w}{\| w \|} \tag{2.8}$$

Também, por x_0 ser um ponto do hiperplano que satisfaz $f(x_0) = 0$, ao substituí-lo na função do hiperplano temos a Equação 2.9:

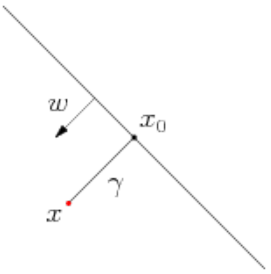

Figura 2.6: Distância do ponto à linha.

$$\gamma = \frac{w^T x + b}{\| w \|} = \frac{f(x)}{\| w \|} \tag{2.9}$$

2.2.2 Definição de classificador marginal máximo

De acordo com a nossa análise prévia, classificamos os pontos de dados e quanto maior é a margem, maior é a confiança na classificação. Pois, para fazer classificações de alta confiança, nós esperamos que os hiperplanos selecionados possam maximizar o valor marginal.

Então, a função objetiva do classificador marginal máximo pode ser definido como a Equação 2.10:

$$Max \quad \gamma y \tag{2.10}$$

Também precisamos satisfazer algumas condições, de acordo com a definição de margem, nós temos a 2.11:

$$y_i(w^T x_i + b) = \widehat{\gamma}_i \geq \widehat{\gamma} \tag{2.11}$$

Onde $\widehat{\gamma} = \gamma \| w \|$, para facilitar a derivação e a otimização, podemos fazer $\widehat{\gamma}$. A função objetiva acima γ pode ser convertida na Equação 2.12:

$$Max \frac{1}{\| w \|}, s.t.(w^T x_i + b) \geq 1, i = 1, 2, \cdots, n \tag{2.12}$$

Podemos achar um classificador marginal máximo através da solução do problema, como mostrado na Figura 2.7, no meio da linha vermelha está o hiperplano ótimo, as outras duas linhas são equidistantes à linha vermelha.

2.3 A Solução do Problema Original ao Problema Dual

Pelo valor máximo de $\frac{1}{\|w\|}$ equivaler o valor mínimo de $\frac{1}{2}\| w \|^2$, temos, no problema acima, a equivalência da Equação 2.12 à Equação 2.13:

$$\begin{aligned} &Min \tfrac{1}{2}\| w \|^2, \\ &s.t.(w^T x_i + b) \geq 1, i = 1, 2, \cdots, n \end{aligned} \tag{2.13}$$

Depois da demonstração supracitada, pode-se enxergar claramente que é um problema de otimização quadrática – a função objetiva é quadrática, mas as condições de restrição são

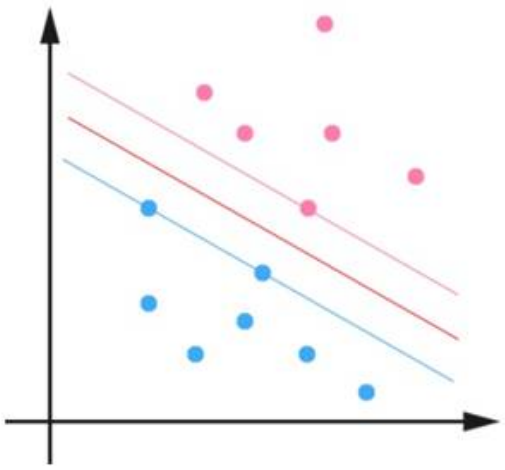

Figura 2.7: Vetores de suporte.

lineares. Apesar desse ser um problema de programação quadrática padronizada (QP), ele também possui uma estrutura especial, podemos achar uma forma mais eficiente para resolver o problema de minimização depois da sua transformação para variáveis duplas de problemas de otimização pela Dualidade de Lagrange (LD). A equação transformada é muito mais eficiente do que a utilização do QP diretamente para resolver o problema de otimização de variáveis duais.

Em suma, a Dualidade de Lagrange é um método que transforma cada condição de restrição em um Multiplicador de Lagrange (LM) e que os funde à função objetiva. Dito isso, temos a seguinte Equação 2.14.

$$L(w, b, \alpha) = \frac{1}{2} \| w \|^2 - \sum_{i=1}^{n} \alpha_i (y_i (w^T x_i + b) - 1) \tag{2.14}$$

Quando as condições de restrição são satisfeitas, $\alpha = 0$, $min \frac{1}{2} \| w \|^2 = \frac{1}{2} \| w \|^2$. Quando as condições não são satisfeitas, temos $\alpha = \infty$, $L(w, b, \alpha) = \infty$. Descrevemos a Equação 2.15 especificamente:

$$Min_{w,b} max_{\alpha_i} \geq 0 L(w, b, \alpha) = p^* \tag{2.15}$$

O p^* representa o valor ótimo do problema, a questão é equivalente à questão original. Mas agora mudamos a posição do máximo e do mínimo, conforme apresentado na Equação 2.16:

$$Max_{\alpha_i} \geq 0 min_{w,b} L(w, b, \alpha) = d^* \tag{2.16}$$

Esse problema, chamado de problema dual, é representado por d^* e $d^* \leq p^*$. Quando a fórmula acima supre as condições KTT, essa formula d é expressa como o dual de p. O melhor modelo matemático de otimização pode ser expresso na sua forma padronizada Equação(2.17):

$$
\begin{aligned}
&minf(x) \\
&s.t.h_j(x) = 0, j = 1, \cdots, p \\
&g_k(x) \leq 0, k = 1, \cdots, q \\
&x \in X \subset R^n
\end{aligned}
\tag{2.17}
$$

onde, $f(x)$ é a função a ser minimizada, $h(x)$ são as restrições de igualdade, $g(x)$ são as restrições de não igualdade, p e q são o número de condições de igualdade e não igualdade. As condições Karush-Kuhn-Tucker de otimização, que referem-se ao tipo do ponto mínimo de $x*$, que precisam ser dadas nas seguintes condições da Equação 2.18:

$$
\begin{aligned}
&h_j(x_*) = 0, j = 1, \cdots, p, g_k(x_*) \leq 0, k = 1, \cdots, q \\
&\bigtriangledown f(X_*) + \sum_{j=1}^{p} \lambda_j \bigtriangledown h_j(x_*) + \sum_{k=1}^{q} \mu_k \bigtriangledown g_k(x_*) = 0 \\
&\lambda_j \neq 0, \mu_k \geq 0, \mu_k g_k(x_*) = 0
\end{aligned}
\tag{2.18}
$$

Após o nosso argumento, a questão satisfaz as condições KTT, em seguida, resolvemos o segundo problema da Equação 2.16 em três passos: (1) minimizar b e w, (2) maximizar α.

1. Minimizar w e b, para que possamos obter a derivação parcial de w e b, sendo estes $\partial L/\partial w$ e $\partial L/\partial b$ igual à 0 nas Equações 2.19 e 2.10:

$$
\frac{\partial L}{\partial w} = 0 \rightarrow w = \sum_{i=1}^{n} \alpha_i y_i x_i \tag{2.19}
$$

$$
\frac{\partial L}{\partial b} = 0 \rightarrow w = \sum_{i=1}^{n} \alpha_i y_i \tag{2.20}
$$

   O resultado acima é substituído na Equação 2.14, portanto, temos a Equação 2.21:

$$
L(w, b, \alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j {x_i}^T x_j \tag{2.21}
$$

   O resultado expressa que a Dualidade Lagrange contém somente uma variável, a de α_i. Se obtermos o valor α_i, ao mesmo tempo, podemos extrair o valor de w e b.

2. Maximizar α, sobre a otimização da variável dual α conforme a Equação 2.22:

$$
\begin{aligned}
&Max_\alpha \sum_{i=1}^{n} \alpha_i - \tfrac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j {x_i}^T x_j \\
&s.t.\alpha_i \geq 0, i = 1, \cdots, n \\
&\sum_{i=1}^{n} \alpha_i y_i = 0
\end{aligned}
\tag{2.22}
$$

   A equação que mostramos acima é utilizada para otimizar a variável dual α. O próximo passo a ser introduzido é um algoritmo que é o princípio do Otimização Sequencial Mínima(SMO), que é utilizado para obter a variável dual otimizável α com mais eficiência.

2.4 Otimização Sequencial Mínima

De fato, dizemos frequentemente que o algoritmo Otimização Sequencial Mínima(SMO) é o processo de resolução de α, conforme a Equação 2.23:

$$\begin{aligned} &Max_{\alpha} W_{(}\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j {x_i}^T x_j \\ &s.t. 0 \leq \alpha_i \leq C, i = 1, \cdots, m \\ &\sum_{i=1}^{n} \alpha_i y_i = 0 \end{aligned} \tag{2.23}$$

Essa fórmula é utilizada para maximizar o valor W no parâmetro $\{\alpha_1, \alpha_2, \cdots, \alpha_n\}$, onde $x^{(i)}$ and $x^{(j)}$ são números dados, onde C é um parâmetro utilizado para controlar o peso do hiperplano máximo e o desvio mínimo dos pontos de dados. Ao comparar com a afirmação acima, percebe-se que a única diferença da variável α é que tem uma parte adicional, o teto C.

2.5 Função de Núcleo

Em um caso não-linear, a abordagem SVM seleciona uma função de núcleo $\kappa(,)$ para resolver os problemas no espaço original através do mapeamento de dados em um espaço de alta dimensão. A expansão não-linear não é muito mais complicada do que o cálculo original, devido a excelente qualidade da função de núcleo. Resumidamente: no caso de lineares inseparáveis, o classificador SVM mapeou essas instâncias no espaço de atributos de alta dimensão através de um método pré-selecionado de mapa não-linear. Em seguida, foi construído um hiperplano que é o novo espaço, conforme é apresentado na Figura 2.8[**?**].

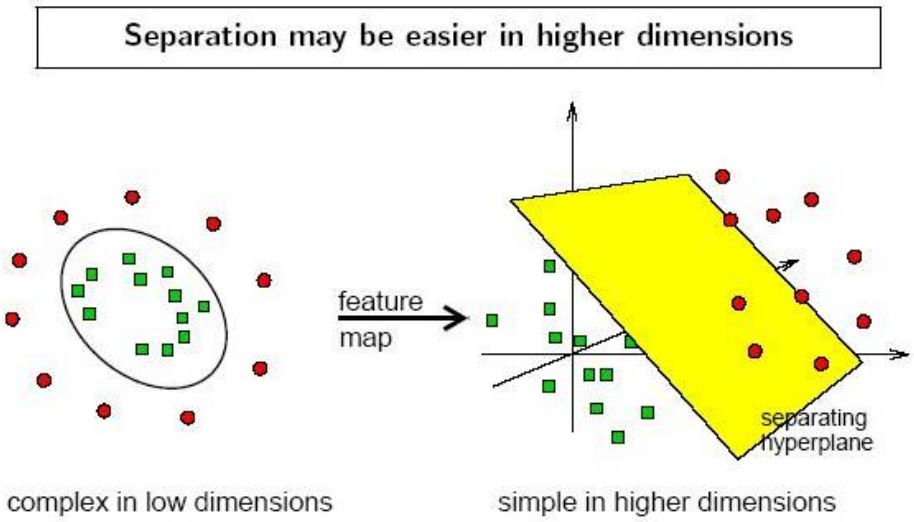

Figura 2.8: Espaço bidimensional é mapeado para o espaço tridimensional.

Isso significa que o seguinte processo de duas fases precisa ocorrer quando construímos o espaço de atributos não-linear:

(a) Primeiramente, mapeamos os dados originais no espaço de atributos.

(b) Depois, utilizamos uma máquina de aprendizagem linear para classificá-lo no espaço de atributos (produto pontual), isto é exibido na Equação 8.4.

$$f(x) = \sum_{i=1}^{N} w_i \phi_i(x) + b \tag{2.24}$$

onde $\phi : x \to F$ é o mapeamento do espaço de entrada para o espaço de atributos.

Fora mencionado o problema dual na Seção 2.2.1. Essa forma dual é uma propriedade importante para a linearidade das máquinas de aprendizagem, significando que os pontos de treinamento podem ser expressos como uma combinação linear, para que possamos utilizar o produto pontual de teste e pontos de treinamento para representar o classificador, conforme a Equação 2.25.

$$f(x) = \sum_{i=1}^{l} \alpha_i \gamma_i < \phi(x_i) \cdot \phi(x) > +b \tag{2.25}$$

Se existir uma função que calcula diretamente o produto pontual $< \phi(x_i) \cdot \phi(x) >$ no espaço de atributos, juntamos os dois passos para criar uma máquina de aprendizado não-linear. Ao definirmos a situação temos: a função nuclear é κ, para todo x, $z \in X$, $\kappa(x, z) = \phi(x) \cdot \phi(z)$, onde φ mapeia o X para o produto pontual no espaço de atributos.

Os dados explicitados acima foram compostos por dois círculos com raios diferentes. Portanto, uma demarcação ideal deverá ser um "círculo" em vez de uma linha (hiperplano). Se representarmos essa demarcação com x_1 e x_2, saberemos que a equação cônica pode ser escrita da seguinte forma Equação 2.26:

$$a_1 X_1 + a_2 (X_1)^2 + a_3 (X_2) + a_4 (X_2)^2 + a_5 X_1 X_2 + a_6 = 0 \tag{2.26}$$

Note que na forma acima, se construirmos outro espaço de cinco dimensões, onde o valor das cinco coordenadas são respectivamente, $Z_1 = X_1, Z_2 = (X_1)^2, Z_3 = X_2, Z_4 = (X_2)^2, Z_5 = X_1 X_2$,podemos reescrever a equação utilizando essas cinco dimensões Equação 2.27:

$$\sum_{i=1}^{5} a_i Z_i + a_6 = 0 \tag{2.27}$$

Sobre a nova coordenada Z, que é o nosso hiperplano. Isso significa que, se nós efetuarmos um mapeamento $\phi : R^2 \to R^5$(dois dimenções para cinco dimenções), e mapear X de acordo com as regras Z, no novo espaço, os dados originais serão linearmente separáveis antes de derivarmos utilizando o algoritmo de classificação linear no processamento.

Agora voltemos para o caso do SVM, assumindo que os dados brutos são não lineares, nós o mapeamos para um espaço de alta dimensão através de $\phi(,)$. Os dados se tornam inseparáveis. No momento, poderemos utilizar a derivação original para calcular, mas toda a derivação é processada no novo espaço no lugar do espaço original.

Claramente, o processo de derivação não é analogicamente direto, por exemplo, precisamos achar um vetor normal w de um hiperplano, mas se as dimensões do novo espaço depois do mapeamento possuir dimensões infinitas é mais complexo descrever um vetor dimensional infinito. Portanto, ignoremos esses detalhes de antemão, e analisamos diretamente as conclusões finais. Dito isso, temos a função final de classificação na forma da

Equação 2.28:

$$f(x) = \sum_{i=0}^{n} \alpha_i y_i < x_i, x > +b \tag{2.28}$$

agora, esse é o espaço depois do mapeamento(Equação 2.29):

$$f(x) = \sum_{i=0}^{n} \alpha_i y_i < \varphi x_i, \varphi x > +b \tag{2.29}$$

Mas há alguns problemas. Em primeiro lugar, como vamos escolher o mapeamento não-linear em um espaço dimensional superior? Em segundo lugar, o cálculo envolvido será dispendioso, remetem à Equação 2.8 para a classificação de uma tupla teste, X^T. Dado a tupla de teste, temos que calcular o seu produto escalar com cada um dos vetores de suporte. No treinamento, temos que calcular um produto escalar semelhante várias vezes, a fim de encontrar o MMH. Isto é particularmente caro. Assim, o produto de ponto de cálculo exigido é muito pesado e dispendioso.

Felizmente, podemos usar um truque de matemática. Acontece que em resolver o quadrático problema de otimização do SVM linear (ou seja, na busca de uma SVM linear no novo espaço dimensional superior), as tuplas de treinamento só aparecem sob a forma de produtos de ponto $\phi(X_i) \cdot \phi(X_j)$ onde $\phi(X)$ é simplesmente a função de mapeamento não linear aplicada a transformar as tuplas de treinamento.

Em vez de calcular o produto escalar nas tuplas de dados transformados, verifica-se que é matematicamente equivalente a aplicar uma função kernel,$\kappa(X_i, X_j)$, para os dados de entrada originais. Temos a seguinte relação lógica para explicar a função núcleo(Equação 2.30).

$$\begin{aligned} &\kappa(x_1, x_2) = (< x_1, x_2 > +1)^2 = 2\gamma_1\varepsilon_1 + {\gamma_1}^2{\varepsilon_1}^2 + 2\gamma_2\varepsilon_2 + (\gamma_2)^2(\varepsilon_2)^2 + 2\gamma_1\gamma_2\varepsilon_1\varepsilon_2 \\ &< \varphi(x_1), \varphi(x_2) > \gamma_1\varepsilon_1 + {\gamma_1}^2{\varepsilon_1}^2 + \gamma_2\varepsilon_2 + \gamma_1\gamma_2\varepsilon_1\varepsilon_2 \end{aligned} \tag{2.30}$$

Em outras palavras, toda a parte que$\phi(X_i) \cdot \phi(X_j)$ aparece no algoritmo de treinamento, podemos substituí-lo com $\kappa(X_i, X_j)$. Desta forma, todos os cálculos são feitos no espaço original de entrada, que é, potencialmente, de muito menor dimensionalidade.

2.6 Variável Slack

No início desse capítulo, assumimos que os dados são linearmente separáveis. Também discutimos a utilização dos métodos de núcleo para generalizar o SVM linear original, mas algumas situações ainda são difíceis de serem manuseadas. Por exemplo, alguns conjuntos de dados foram separados não linearmente devido aos ruídos de dados. Esses tipos de pontos de desvio (Figura 2.9) são chamados de *outliers*. Em um modelo SVM, reconhecemos que *outliers* exercem uma grande influência sobre o hiperplano. O hiperplano é composto por poucos vetores de suporte, se o outlier for um deles, exercerá uma grande influência sobre o hiperplano Figura 2.9.

O ponto azul circundado por um círculo preto é o outlier. A aparência do *outlier* apontou para um mal resultado, onde o hiperplano foi comprimido à uma margem diferente mostrado pela linha preta seccionada. Para lidar com essa situação, o SVM utiliza

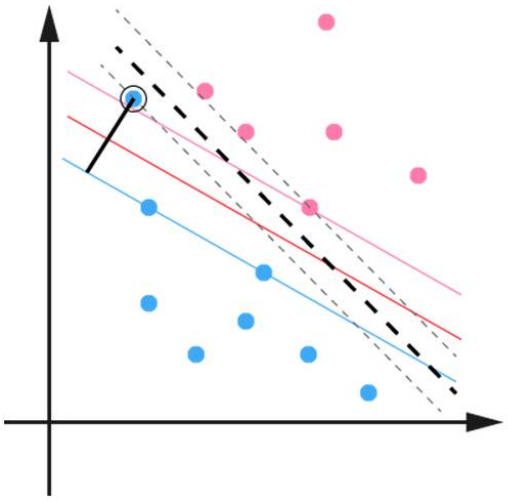

Figura 2.9: Variável *Slack*.

a técnica de variável *slack* que permite com que o *outlier* se locomova de volta para o lado original.

Em outras palavras, os pontos do *outlier* se tornaram parte do vetor de suporte. Portanto, a restrição original se torna a Equação 2.31:

$$y_i(w^T x_i + b) \geq 1 - \xi_i, i = 1, \cdots, n \tag{2.31}$$

Onde $\xi_i \geq 0$ é chamado de variável *slack*, correspondendo ao alcance permitido do ponto x_i. Se ξ_i é arbitrariamente grande, então qualquer hiperplano se encaixa nas condições pré-estabelecidas. Portanto, adicionamos um $C\sum_{i=1}^{n}\xi_i$ na função objetiva original(Equação 2.32):

$$Margem = Min\frac{1}{2}\| w \|^2 + C\sum_{i=1}^{n}\xi_i \tag{2.32}$$

Onde o C é um parâmetro, que é utilizado para controlar a margem máxima e o ponto de desvio de dados mínimo. Note que o ξ precisa ser otimizado, sendo o C uma constante pré-determinada.

Ao utilizar métodos anteriores para adicionar restrições ou limites à função objetiva, obtemos uma nova função lagrangiana descrita abaixo:

$$L(x, \xi, b, \alpha, r) = \frac{1}{2}\| w \|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i(y_i(w^T x_i + b) - 1 + \xi_i) - \sum_{i=1}^{n} r_i\xi_i \tag{2.33}$$

Ao utilizar a técnica analítica anterior, minimizamos o ξ.

$$\frac{\partial L}{\partial \xi_i} = 0 \rightarrow C - \alpha_i - r_i = 0, i = 1, \cdots, n \tag{2.34}$$

Entretanto, como obtemos o $C - \alpha_i - r_i = 0$ (como uma condição de multiplicador de lagrange), e o $r_i \geq 0$, então temos $\alpha_i \leq C$. Portanto, o problema dual na sua totalidade

pode ser escrito como na Equação 2.35:

$$\begin{aligned} &Max_{\alpha}W_{(}\alpha) = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j < x^{(}i), x^{(}j) \\ &s.t. 0 \leq \alpha_i \leq C, i = 1, \cdots, m \\ &\sum_{i=1}^{n}\alpha_i y_i = 0 \end{aligned} \tag{2.35}$$

Percebe-se que a única diferença é o limite superior C na variável dual α. Similarmente, no caso da função de núcleo, substituímos o $< x^{(i)}, x^{(j)} >$ por $\kappa < x^{(i)}, x^{(j)} >$.

Capítulo 3

Sistema de Recomendação

Sistemas de Recomendação são ferramentas de software e técnicas que fornecem sugestões de itens para um usuário [15] [16]. As sugestões referem-se a vários processos de tomada de decisão, tais como, qual item comprar, qual musica ouvir, ou quais notícias on-line ler. O "Item"é o termo geral usado para designar o que o sistema recomenda aos usuários. Um RS normalmente se concentra em um tipo específico de item (por exemplo, CDs, ou notícias) e, consequentemente, seu *design*, sua interface gráfica do usuário, e a técnica de recomendação central utilizada para gerar as recomendações são todas personalizadas para proporcionar sugestões úteis e eficazes para um tipo especifico de produto.

Sistemas de recomendação provaram ser mecanismos valiosos para que os usuários online consigam lidar com a sobrecarga de informações e têm se tornado uma das ferramentas mais populares no *e-commerce*, redes sociais, sites de filmes e vídeos, rádios online, etc. Sendo assim, várias técnicas de geração de recomendações foram propostas.

RS são direcionados principalmente para pessoas que não têm experiência suficiente para avaliar o número potencialmente enorme de itens que um site, por exemplo, pode oferecer [17]. Um caso em questão é um sistema de recomendação de livros que auxilia os usuários a escolher um livro para ler. No site *Amazon.com*, existe um RS para personalizar a loja on-line de cada cliente[18].

Uma ideia interessante surgiu nos anos 1990 – a primeira concepção de sistema de recomendação foi proposta pelo *Palo Alto Research Center*(PARC) Tapestry system que introduziu a ideia de filtragem colaborativa. Menos de dois anos depois, o sistema GroupLens [7] mostrou que a filtragem colaborativa pode ser distribuída entre uma rede e ser automatizada. O sistema de recomendações foi considerado como um grande problema em Berkeley em março de 1996. Neste ano, a edição especial de Communications da *Association for Computing Machinery*(ACM)[17], agregou todos os trabalhos sobre sistemas de recomendação

No final dos anos 90, os sistemas de recomendação acompanharam a rápida expansão do *e-commerce* e sua comercialização foi quase imediata. Desde a comercialização dos sistemas de recomendação, nós enfrentamos problemas desconhecidos nas pesquisas. Neste período, o principal objetivo das pesquisas era resolver os problemas que envolviam o lado comercial.

3.1 Recomendação de Filtragem Colaborativa

Algoritmos de recomendação de filtragem colaborativa(CF) são um sistema de primeira geração que é amplamente utilizado no ramo de *e-commerce* e também em redes sociais. Esse sistema foi introduzido não somente devido ao seu valor histórico, mas também a maioria dos recomendadores on-line dependem dessas técnicas. Visto de uma perspectiva prática, percebe-se que filtragem colaborativa baseada em item, como dito por[19] e utilizado pelo site *Amazon*.com pode ser escalado para lidar com grandes bases de dados de avaliações, e que portanto, produzem recomendações de qualidade razoável. O trabalho de Joachim[20] interpreta que o *Google News* usa a abordagem de recomendação colaborativa baseado no histórico de clique do usuário, ou seja, um clique em um artigo é interpretado como uma avaliação positiva. Konstas [21] criou um sistema de recomendação colaborativa que adapta as necessidades informacionais de cada usuário. Eles adotaram um *framework* genérico de caminhadas aleatórias com *restarts* para prover um modo mais eficiente de representar redes sociais. Ye [22] desenvolveu uma abordagem de filtragem colaborativa baseada em amigos para recomendação de local baseado nas avaliações de lugares feitas por amigos nas redes sociais.

Recomendação baseada em usuários de vizinho mais próximo: A ideia principal é a seguinte: dado um banco de dados de avaliações e o ID do usuário atual (ativo) como uma entrada, identificamos outros usuários (por vezes referido como usuários pares ou vizinhos mais próximos) que tenham preferências semelhantes aos do usuário ativo no passado[23] Figura 3.1. Então, para cada produto que o usuário ativo ainda não tenha visto, uma previsão é calculada com base na classificação para o produto feita pelos usuários pares. Os pressupostos subjacentes de tais métodos são de que (a) se os usuários tinham gostos semelhantes no passado terão gostos semelhantes no futuro e (b) as preferências do usuário mantem-se estáveis e consistentes ao longo do tempo[24][25].

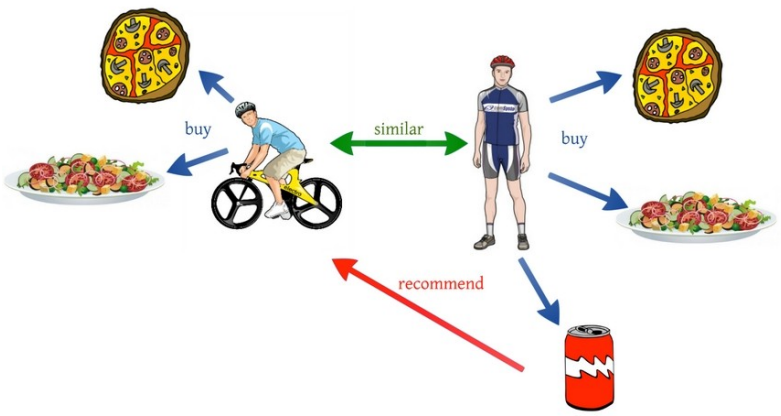

Figura 3.1: Cenário de operação do sistema de recomendação colaborativa

Mineração de regras de associação: É uma técnica comum usada para identificar padrões de relacionamento de regras em transações de vendas em larga escala. Uma aplicação típica desta técnica é a detecção de pares ou grupos de produtos em um supermercado que muitas vezes são comprados juntos. Uma regra típica poderia ser: "Se um cliente compra comida de bebê, então ele ou ela também compra fraldas em 70 por cento

dos casos."Quando essas relações são conhecidas, este conhecimento pode, por exemplo, ser explorado para fins promocionais e de vendas cruzadas ou para decisões de projeto sobre o *layout* da loja.

Abordagens de recomendação probabilística: Uma maneira de implementar filtragem colaborativa com um método probabilístico é ver o problema de previsão como um problema de classificação, o que geralmente pode ser descrito como a tarefa de "atribuir um objeto a uma das várias categorias predefinidas"[26]. Como um exemplo de um problema de classificação, considere a tarefa de classificar as mensagens de *e-mails* recebidos como *spam* ou não *spam*. Para automatizar esta tarefa, é necessário desenvolver uma função que define - com base, por exemplo, nas palavras que ocorrem no cabeçalho da mensagem ou conteúdo - se a mensagem é classificada como *spam* ou não. A tarefa de classificação pode, portanto, ser vista como o problema de aprender esta função de mapeamento a partir de exemplos de treinamento. Essa função também é informalmente chamada de modelo de classificação.

A popularidade dos sistemas de recomendação tem razões diferentes, uma delas é o fato de que os problemas de referência do mundo real e os dados que estão disponíveis a serem analisados para gerar recomendações têm uma estrutura muito simples: uma matriz de classificações de itens. Assim, a avaliação de se uma técnica de recomendação recém-desenvolvida, ou a aplicação de métodos existentes para o problema recomendação supera abordagens anteriores é simples, em particular porque as métricas de avaliação também são mais ou menos padronizadas. Pode-se facilmente imaginar que comparar diferentes algoritmos nem sempre é tão fácil como com filtragem colaborativa, em especial se mais conhecimento está disponível além da matriz de classificação simples. Pense, por exemplo, em aplicações de recomendação de conversação, no qual o usuário é interativamente perguntado sobre suas preferências e em que o conhecimento de domínio adicional é codificado.

No entanto, a disponibilidade de bases de dados de teste para a CF em domínios diferentes favoreceu o desenvolvimento das diversas e complexas técnicas de CF. Ainda assim, isso também estreitou a faixa de domínios em que as técnicas de CF são realmente aplicadas. Os conjuntos de dados mais populares são sobre filmes e livros, e muitos pesquisadores têm como objetivo melhorar a precisão de seus algoritmos apenas sobre esses conjuntos de dados. Se uma determinada técnica CF desempenha particularmente bem em um ou outro de domínio é, infelizmente, além do alcance de muitos esforços de investigação.

3.2 Recomendação Baseada em Conteúdo

Nada se sabe sobre os itens a serem recomendados para a aplicação de técnicas de filtragem colaborativa, exceto as avaliações do usuário. A principal vantagem é que a tarefa onerosa de fornecer uma descrição detalhada e atualizadas de item para o sistema é evitado. Por outro lado, com uma abordagem de filtragem colaborativa pura, uma forma muito intuitiva de selecionar produtos recomendáveis com base em suas características e as preferências específicas de um usuário não é possível: no mundo real, seria fácil recomendar o novo livro do *Harry Potter* para Alice, se sabemos que (a) este livro é um romance de fantasia e (b) Alice sempre gostou de romances de fantasia Figura 3.2 [**?**].

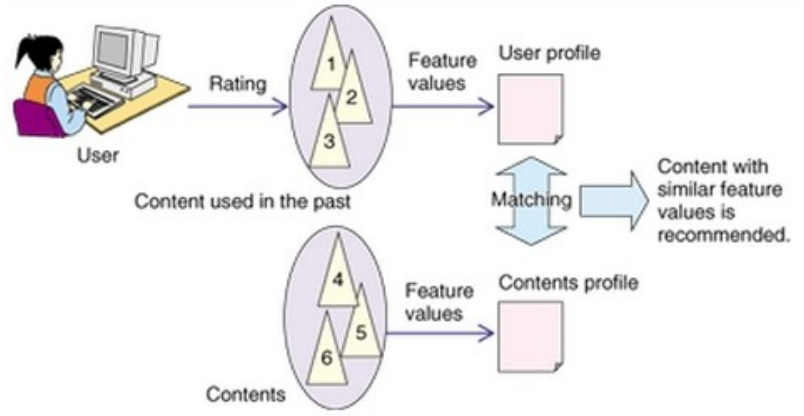

Figura 3.2: Cenário de operação do sistema de recomendação baseada em conteúdo

Um sistema de recomendação eletrônico pode realizar essa tarefa somente se duas informações estão disponíveis: a descrição das características de itens e um perfil de usuário que de alguma forma descreve os interesses(do passado) de um usuário, talvez em termos de características de itens preferenciais. Em seguida, a tarefa de recomendação consiste em determinar os itens que mais combinam com as preferências do usuário. Este processo é comumente chamado de recomendação baseada em conteúdo; tal abordagem deve contar com informações adicionais sobre os itens e as preferências do usuário, ele não exige a existência de uma grande comunidade de usuários ou uma história classificação - ou seja, listas de recomendação podem ser geradas mesmo se existe apenas um único utilizador.

Em cenários práticos , as descrições técnicas dos recursos e características de um item - tais como o gênero de um livro ou a lista de atores em um filme - são mais frequentemente disponíveis em formato eletrônico, como são parcialmente fornecidas pelos fornecedores ou fabricantes das mercadorias. O que continua sendo um desafio, no entanto, é a aquisição de características subjetivas e qualitativas . Nos domínios de qualidade e sabor, por exemplo, os motivos para que alguém goste de algo não são sempre relacionados com certas características do produto e podem ser baseados numa impressão subjetiva da estrutura exterior do produto. Um esforço notável e excepcional, nesse contexto, é o *"Music Genome Project"*, cujos dados são utilizados pelo recomendador de música no rádio pela Internet para descoberta de músicas no site popular *Pandora.com*. Nesse projeto, as músicas são catalogadas manualmente por músicos de acordo com diversas características da música, tais como a instrumentação, influências, ou instrumentos. Tal processo de aquisição Manual – catalogar uma música leva cerca de vinte a trinta minutos, como afirmado pelos prestadores de serviços - é, no entanto, muitas vezes inacessível.

Não há fronteira exata entre sistemas baseados em conhecimento e baseados em conteúdo na literatura, alguns autores chegam a ver abordagens baseadas no conteúdo como um subconjunto de abordagens baseadas no conhecimento. Neste estudo, nós seguimos o esquema de classificação tradicional, em que os sistemas baseados em conteúdo são caracterizados por seu foco em explorar as informações nas descrições de itens, enquanto que em sistemas baseados em conhecimento normalmente o usuário fornece uma informação adicional, tal como a qualidade desejada para um produto, para a produção de

recomendações. A maneira mais simples para descrever itens de catálogo é através de uma lista explícita de recursos para cada item (também muitas vezes chamado de atributos, características ou perfis do artigo). Para a recomendação de um livro, pode-se, por exemplo, usar o gênero, o nome do autor, o editor, ou qualquer outro atributo que descreve o item e armazenar essas informações em um sistema de banco de dados relacional. Quando as preferências do usuário são descritas em termos de seus interesses, usando exatamente esse conjunto de características, a tarefa de recomendação consiste em combinar as características dos pontos e as preferências do usuário.

A informação sobre o editor e o autor não são realmente conteúdos de um livro, mas sim conhecimento adicional sobre o assunto. Contudo, os sistemas baseados em conteúdo têm sido historicamente desenvolvido para filtrar e recomendar itens baseados em texto, como mensagens de *e-mail* ou notícias. A abordagem padrão na recomendação baseada em conteúdo é portanto, não para manter uma lista de recursos, uma "meta-informação", como no exemplo anterior, mas para usar uma lista de palavras-chave relevantes que aparecem no documento. A ideia principal, é que essa lista pode ser gerada automaticamente a partir do próprio conteúdo do documento ou de uma descrição em texto livre.

Entre os métodos de classificação, também temos os métodos probabilísticos e os métodos lineares:

Métodos probabilísticos: Os métodos de classificação mais importantes desenvolvidos em sistemas de classificação de texto primeiros são os probabilísticos. Essas abordagens são baseadas na independência condicional *Naive Bayes* (NB) (com relação a ocorrências prazo) e também foram implantados com sucesso em recomendadores baseados em conteúdo. O classificador de Bayes mostrou levar a resultados surpreendentemente bons e é amplamente utilizado para a classificação de texto. Uma análise das razões para esta prova um pouco contra-intuitiva pode ser encontrada nos trabalhos de Shen [26], e pesquisas recentes têm mostrado bons resultados. Manning [27] resolve o problema de rotular um documento como relevante ou irrelevante em um cenário de recomendação de documentos, que pode ser visto como um caso especial da classificação mais ampla e mais velha de texto que consiste em atribuir um documento para um conjunto de classes pré-definidas. As aplicações desses métodos podem ser encontrados em recuperação de informação para a resolução de problemas, tais como *e-mail* pessoal de triagem, detecção de páginas de *spam*, ou detecção de sentimento.

Classificadores lineares: Ao visualizar o problema de recomendação baseada em conteúdo como um problema de classificação, várias outras técnicas de aprendizado de máquina podem ser empregadas. A um nível mais abstrato, a maioria dos métodos de aprendizagem visam encontrar coeficientes de um modelo linear para discriminar entre documentos relevantes e não relevantes. Muitos algoritmos de classificação de texto são classificadores lineares, e isto pode ser facilmente demonstrado; tanto o classificador de Bayes e o método Rocchio se enquadram nesta categoria [27]. Outros métodos para o aprendizado de classificadores lineares são, por exemplo, o algoritmo de Widrow-Hoff [28] ou máquinas de vetores de suporte [29]. Um outro desafio quando se utiliza um classificador linear é lidar com ruído nos dados. Pode haver características ruidosas que induzem o classificador ao erro se forem incluídos na representação do documento. Além do mais, também podem existir documentos ruidosos por quaisquer motivos, entretanto, a identificação desses ruídos não é difícil. Uma avaliação comparativa das diferentes técnicas de formação para classificadores de texto pode ser encontrada em Lewis[30] e em Yang[31].

Apesar de que, nesses experimentos alguns algoritmos e, em especial os baseados em SVM, possuem um desempenho melhor do que os outros, não existe nenhuma orientação estrita sobre qual técnica tem o melhor desempenho em todas as situações. Além disso, nem sempre é claro se a utilização de um classificador linear é a escolha certa em todas as ocasiões, pois existem muitas configurações em que as fronteiras de classificação não podem ser razoavelmente aproximadas por uma linha ou por um hiperplano.

3.3 Recomendação Baseada em Conhecimento

Os algoritmos colaborativos e baseados em conteúdo possuem pontos fortes e fracos, entretanto, existem diversas situações para que essas abordagens não são as mais apropriadas. Normalmente, não compramos um carro, casa ou computador com muita frequência. Nesse cenário, um sistema puro de CF não possuirá um bom desempenho devido à baixa disponibilidade de avaliações [32].

Além disso, os intervalos de tempo desempenham um papel importante. Por exemplo, avaliações de cinco anos atrás para computadores podem ser inadequadas para recomendações baseados em conteúdo. O mesmo vale para itens como carros ou casas. Como as preferências dos usuários mudam ao longo do tempo, por exemplo, mudanças no estilo de vida ou situações familiares, os domínios de produtos mais complexos, tais como carros, os clientes muitas vezes querem definir suas exigências explicitamente - por exemplo, "o preço máximo do carro é x e a cor deve ser preta". A formulação de tais requisitos não é típica para os quadros de recomendação colaborativos baseados em conteúdo puros.

Os sistemas de recomendação baseados em conteúdo nos ajudam a enfrentar os desafios supracitados. Algumas vantagens desse sistema incluem a inexistência de problemas *ramp-up* pois os seus cálculos não necessitam de dados de avaliações. As recomendações são feitas independentemente de avaliações individuais de usuários: ou na forma de similaridades entre as necessidades dos usuários e itens ou baseados em regras de recomendação explícitos. Interpretações tradicionais do que deveria ser um sistema de recomendação focam no aspecto de filtragem de informação [33] [34], no qual os itens que provavelmente serão de interesse para um consumidor especifico são retirados através da filtragem. Em contraste, os aplicativos baseados em recomendação baseadas em conhecimento são altamente interativos, uma propriedade fundacional que é razão para a sua classificação como um sistema de conversação [32]. Esse aspecto de interatividade suscitou um distanciamento da interpretação como um sistema de filtragem e mais para uma interpretação onde os recomendadores são definidos como sistemas que guiam um usuário de forma personalizada para objetos interessantes ou úteis em um amplo espaço de opções possíveis [32]. Recomendadores que dependem de fontes de informação não exploradas por abordagens colaborativas ou de conteúdo são automaticamente definidos como recomendadores baseados em conteúdo por [32] e [35].

Dois tipos básicos de sistemas de recomendação baseados em conhecimento são sistemas baseados em restrição [35][36][37] e sistemas baseados em casos [32]. Ambas as abordagens são semelhantes em termos do processo de recomendação: o usuário deve especificar os requisitos e o sistema tenta identificar uma solução. Se não for encontrada uma solução, o usuário deve alterar os requisitos. O sistema também pode fornecer explicações para os itens recomendados. Estes recomendadores, no entanto, diferem na forma como eles usam o conhecimento proporcionado: recomendadores baseados em caso se con-

centram na recuperação de itens semelhantes, com base em diferentes tipos de medidas de similaridade, enquanto recomendadores baseados em restrição contam com um conjunto explicitamente definido de regras de recomendação.

Baseados em restrição: O fluxo geral de interação de um recomendador de conversação baseado em conhecimento pode ser descrito abaixo:

- O usuário especifica suas preferências iniciais - por exemplo, usando um formulário on-line. Tais formas podem ser idênticas para todos os usuários ou personalizadas à situação específica do usuário atual. Alguns sistemas utilizam um processo de elicitação de preferência com perguntas e respostas, em que as perguntas podem ser feitas tanto de uma só vez ou de forma incremental em um assistente de estilo, de diálogo interativo, como descrito por Felfernig [38].
- Quando informações suficientes sobre as necessidades e preferências do usuário forem coletadas, um conjunto de itens correspondentes são apresentados ao usuário. Opcionalmente, o usuário pode solicitar as razões pelas quais o item foi lhe recomendado.
- O usuário poderá alterar as suas necessidades, Por exemplo, para ver uma solução alternativa ou diminuir os itens correspondentes.

Embora este regime geral de interação do usuário pareça ser bastante simples, a fim de implementar padrões de interação mais elaborados para dar suporte ao usuário final no processo de recomendação, aplicações mais práticas são necessárias. Considere, por exemplo, situações nas quais nenhum dos itens no catálogo satisfaz todas as necessidades dos utilizadores. Nessas situações um recomendador de conversação deve inteligentemente apoiar o usuário final a resolver o seu problema, por exemplo, propor de forma proativa alternativas de ação.

Baseados em caso: semelhante aos recomendadores baseados em compressão, versões iniciais de recomendadores baseados em casos seguem uma abordagem pura baseada em perguntas, onde um usuário deve especificar as suas necessidades até um que item-alvo (um item que se encaixe nas necessidades do usuário) seja identificado [39]. Para amadores, esse tipo de pré-requisito pode levar a sessões de recomendação tediosas, visto que as propriedades interdependentes necessitam um conhecimento amplo de domínio para obter os melhores resultados [39]. Essa desvantagem de abordagens puras baseadas em perguntas motivaram o desenvolvimento de abordagens baseadas em navegação para a retirada de itens em que usuários – muitas vezes não sabendo o que procuram – navegam no espaço de item com o objetivo de achar alternativas úteis. A avaliação é um método efetivo de apoiar as navegações e também consistem em um ponto crucial no conceito de recomendação baseado em caso. Através das avaliações, os usuários conseguem especificar os seus pedidos de mudança no formato de objetivos que não foram alcançados pelo item sob consideração [32][40]. Por exemplo, se o preço da câmera digital em questão for muito alto, uma avaliação para que se tenha um produto mais barato é ativada; se o usuário deseja ter uma câmera com maior resolução (mpix), uma avaliação correspondente para mais mpix pode ser ativada. Reilly [41] propôs um método baseado em caso que explora dinâmicas de avaliação, que consiste em descrições genéricas das diferenças entre o item recomendado e os itens candidatos. Esses padrões são utilizados para avaliação

de derivação de compostos. As críticas são chamadas de dinâmicas pois são engendradas imediatamente em cada ciclo de avaliação.

Há dois exemplos famosos de sistemas de recomendação baseada em conhecimento, um é um aplicativo de recomendação baseada em restrições desenvolvido por um fornecedor de serviços financeiros húngaro e o outro é um ambiente de recomendação baseada em casos desenvolvido para recomendar restaurantes localizados em Chicago. O RS baseado em conhecimento também é amplamente utilizado em serviços financeiros. A aplicação de recomendação de serviços financeiros VITA foi construído para a associação de empréstimo Fundamental na Hungria [42], a fim de apoiar os representantes de vendas em diálogos comerciais com os clientes. Ele foi desenvolvido a partir do ambiente recomendador CWAdvisor apresentado por Felfernig [38]. Um método conhecido de um aplicativo comercial baseado em avaliações é o Entree – um sistema desenvolvido para a recomendação de restaurantes em Chicago [43]. O objetivo inicial era orientar os participantes da Convenção Nacional do Partido Democrata em 1996, realizado em Chicago, mas o seu sucesso prolongou a sua utilização por vários anos.

Outro exemplo foi utilizado em diversas áreas para demonstrar a aplicação de tecnologias de recomendação baseados em críticas em ambientes móveis [44]. Outras contribuições científicas importantes no campo de aplicativos recomendadores baseados em críticas podem ser encontradas em Lorenzi[45]. Jiang[46] introduziu uma abordagem para recomendadores de câmeras digitais de multimídia, no qual mudanças nas necessidades dos consumidores não resultam somente em uma mudança no conjunto de recomendações, mas essas mudanças também acontecem em tempo real. Por exemplo, uma mudança no objetivo pessoal de fotos de perfis para fotos esportivas resultaria em uma mudança de lente de uma lente padrão para uma lente rápida especialmente feita para movimentos de alta velocidade tipicamente encontradas em cenas esportivas.

3.4 Abordagens de Recomendação Híbridas

As três abordagens de recomendação mais proeminentes discutidas nas seções anteriores exploram diferentes fontes de informação e seguem diferentes paradigmas para efetuar as recomendações. Mesmo que elas produzam resultados que são considerados personalizados baseados em interesses assumidos dos seus recipientes, elas possuem um desempenho com diferentes graus de sucesso em domínios de aplicação diferentes. A Figura 3.3 esboça um sistema de recomendação como uma caixa preta que transforma dados de entrada em uma lista ordenada de itens como saída. Modelos de usuários e informações de contexto, dados da comunidade e de produtos e modelos de conhecimento constituem os potenciais tipos de entrada de recomendação. No entanto, nenhuma das abordagens básicas é capaz de explorar totalmente todos os itens. Consequentemente, os sistemas de construção híbridos que combinam as vantagens de diferentes algoritmos e modelos para superar algumas das deficiências e problemas acima mencionados tornaram-se alvo de pesquisas recentes. De um ponto de vista linguístico, o termo híbrido deriva do latim *nounhybrida* (de origem mista) e denota um objeto feito pela combinação de dois elementos diferentes. Analogamente, sistemas de recomendação híbridos são abordagens técnicas que combinam várias implementações de algoritmos ou componentes de recomendação.

Embora muitas aplicações de recomendação são realmente híbridas, pouco trabalho teórico tem-se centrado sobre a forma de hibridizar algoritmos e em que situações se

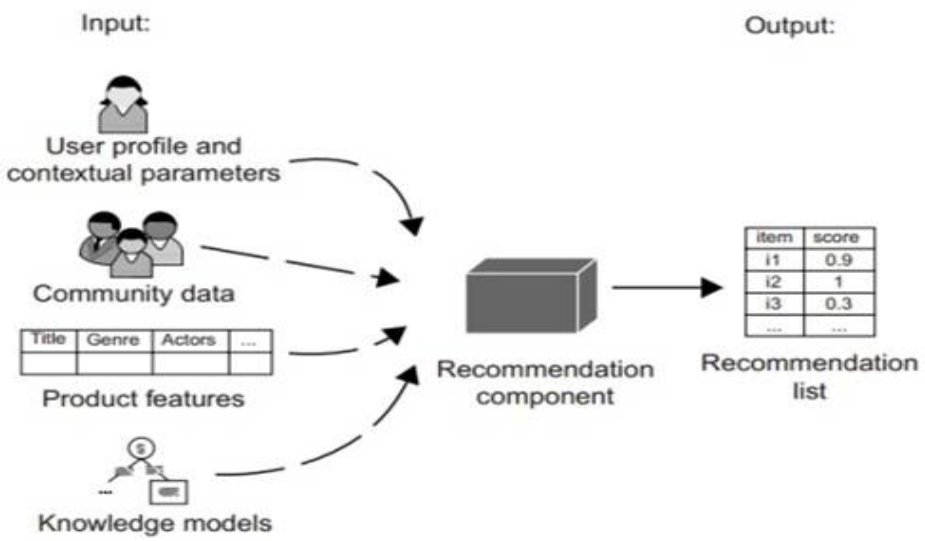

Figura 3.3: Recomendação Híbridos

pode esperar benefícios da hibridização. Um excelente exemplo para a combinação de diferentes variantes do algoritmo de recomendação é a competição Prêmio *Netflix*, em que centenas de estudantes e pesquisadores se uniram para melhorar o recomendador de filme colaborativo hibridando centenas de diferentes técnicas de filtragem colaborativa e abordagens para melhorar a precisão geral. O artigo de Robin Burke, "Sistemas de Recomendação híbridos: Levantamento e Experiências"[47] é uma pesquisa bem conhecida do espaço de *design* de diferentes algoritmos de recomendação híbrido. Ele propõe a taxonomia das diferentes classes de algoritmos de recomendação.

Capítulo 4

Estado de Arte

Neste capitulo, serão apresentadas duas seções para introduzir os Sistemas de Recomendação em Redes Sociais e SVM nas Redes Sociais. Apresentamos os últimos resultados da descoberta, nós apontamos alguns dos problemas que foram resolvidos, porém, ainda há vários campos a serem melhorados. Algumas das soluções serão propostas nesse trabalho e alguns desses desafios serão estudados no futuro.

4.1 Sistema de Recomendação nas Redes Sociais

Apesar dos sistemas de recomendação terem sido intensamente analisados na última década, o surgimento do *Facebook* e do *Twitter* – juntamente com o seu estabelecimento como formas de comunicação amplamente difundidas nos últimos anos – impulsionou os estudos em sistemas de recomendação baseados em redes sociais.

Recomendação de Amigos

Sites de redes sociais permitem os usuários articular a sua vida social através da adição de outros usuários para a sua lista de amigos. Pesquisas mostram que usuários se conectam não somente aos amigos previamente conhecidos na vida real, mas também se articulam socialmente com novos amigos conhecidos através dos sites de interação social. Chen[48] mostrou que a recomendação de pessoas em redes social é válida ser estudada pois ela é diferente da recomendação tradicional de livros, filmes, restaurantes, etc. Ademais, a lista de amigos de um usuário pode ser visualizada no seu perfil, isso torna flagrante a percepção de um amigo por outros no mesmo site. Essas dinâmicas sociais podem ser obstáculos na aceitação de recomendações, mesmo quando elas são relevantes e desejáveis.

Chen também encontrou e avaliou quatro algoritmos de recomendação efetivos em um site de rede social. Algoritmos baseados em informações extraídas de redes sociais foram capazes de produzir recomendações com uma taxa de aceitação melhor e encontrar mais contatos para usuários, enquanto algoritmos utilizando a similaridade de conteúdos criados por usuários se mostraram mais eficientes na busca de novos amigos para usuários.

Esse estudo prova que o algoritmo baseado em relações é melhor do que os demais algoritmos. Porém, falta estudar a fundo a efetividade de diferentes atributos na recomendação de pessoas. Ma [49] resolveu esse problema detalhadamente. Eles elaboraram sobre como informações sobre redes sociais podem beneficiar sistemas de recomendação

e utilizaram regularização social para representar os limites sociais em sistemas de recomendação, dito isso, sistematicamente ilustramos como elaborar uma função objetiva de fatoração de matriz como Regularização Social.

Redes Sociais Multidimensionais(MSN)

Todos os sistemas online de compartilhamento angariam dados que refletem o comportamento coletivo dos usuários e as suas atividades em comum. Esses dados podem ser utilizados para extrair diferentes tipos de relações. Kazienko[50] propôs que esses diferentes tipos de relacionamentos podem ser agrupados em níveis e que são os componentes básicos das redes sociais multidimensionais (MSN). Isso significa que essas relações podem ser facilmente extraídas a partir dos dados sobre atividades de usuários. Os usuários do sistema de compartilhamento em multimídia(MSS), juntamente com relações interpessoais diretas e indiretas, podem ser tratados como uma rede social heterogênea multidimensional (MSN).

Kazienko analisou perfis de 9 diferentes níveis de relação dentro da MSN extraídos dos dados disponíveis em um sistema de compartilhamento de fotos. Esses níveis podem refletir tanto inspirações sociais quanto semânticas de atividades de usuários. Esse MSN é utilizado em sistemas de recomendação para a sugestão de um usuário para outro, e em consequência, expandir a comunidade humana online.

A pesquisa revelou que relações baseadas em marcações dominam cada vez mais o MSN dentro do sistema de publicação online. A grande quantidade de cálculos resultam em problemas de eficácia como todo o processo é feito online. Para solucionar esse problema, algumas tarefas podem ser feitas offline e repetidas periodicamente.

Dados de *Networking* Geo-Social Baseado em Localidade

A popularidade de redes sociais baseadas em localidade nos dá com uma nova plataforma para entender as preferências do usuário baseando-se no histórico de localidade, pois este atributo consiste em um componente crucial no contexto do usuário e o seu comportamento, provendo-nos com oportunidades para melhor entender usuários em uma estrutura social não somente de acordo com o seu comportamento online, mas também a mobilidade e as atividades no mundo físico.

Neste artigo [51], é apresentado um sistema de recomendação baseado em localidade e sensível a preferências que oferece um usuário com um conjunto de localidades (como restaurantes) dentro de um limite geoespacial que considera: (1) preferências de usuários, que são extraídas do histórico de localidades do usuário, e (2) opiniões sociais, que são mineradas dos históricos de localidades de especialistas locais. Esse sistema de recomendação facilita os deslocamentos dos usuários não somente no local onde habitam, mas também em cidades desconhecidas. Como um usuário pode visitar um número limitado de localidades, a matriz de localidades de usuários é muito esparsa, consistindo em um grande desafio para sistemas de recomendação tradicionais baseadas em filtragem colaborativa.

Eles propuseram um novo sistema de recomendação de localidades, que consiste em duas partes principais: modelagem offline e recomendação online. A parte de modelagem offline engendra as preferências individuas dos usuários com uma categorização hierarqui-

zada baseada em pesos (WCH) e infere o conhecimento de cada usuário sobre uma cidade em despeito a diferentes categorias de localidades de acordo com o seu histórico utilizando um modelo interativo de aprendizagem. A parte de recomendação online seleciona candidatos a especialistas locais em um alcance geoespacial que combina com as preferências do usuário utilizando um algoritmo de seleção de candidatos sensível a preferências, após isso, infere-se uma pontuação aos candidatos baseados nas opiniões de especialistas locais.

O seu benefício consiste na solução do problema de disparidade de dados na matriz original de localidade de usuário, porém, usuários reais possuem interesses com escalas de tempo distintas (por exemplo, interesses de curto prazo relacionados ao planejamento de viagens e interesses de longo prazo relacionados ao local de habitação e preferências politicas), esses sistemas de recomendação negligenciam as datas e horários das avaliações.

Avaliações Baseadas em Círculos

Para melhor servir as atividades dos usuários em domínios diferentes, muitas redes sociais suportam novos atributos de "Círculos de Amizades"[52], que refinam o conceito de " Amizade". Os RS também devem se beneficiar dos "Círculos de Confiança" com domínio específico. Um usuário pode confiar em diferentes subcategorias de amigos relacionados à domínios diferentes. Infelizmente, na maioria dos conjuntos de dados existentes de avaliação, as conexões sociais de todas as categorias são misturadas. Dito isso, propuseram um conjunto de algoritmos para inferir círculos de amigos específicos em categoria para inferir o valor da confiança em cada link baseado em avaliações de atividades de usuários em cada categoria.

Existe um fenômeno real onde usuários confiam em subconjuntos diferentes de amigos em diferentes domínios públicos. Todos possuem a sua área de especialidade para fazer uma recomendação para outra pessoa. Nesse artigo, o autor explora o seu próprio algoritmo somente com uma categoria top 10. Isso significa que eles não articulam a adaptabilidade do algoritmo devido à redução artificial da categoria. Caso contrário, o problema da disparidade de dados aparecerá. Ao mesmo tempo, esse algoritmo não recomenda um item regional (e.g. restaurante, cinema) próximos à vida real do usuário.

Importância de Microblog para Recomendação Comercial Online

Recentemente, existe um fenômeno que merece a nossa atenção total. Quando um usuário faz login em um portal conhecido (por exemplo, *TripAdvisor*, *Groupon*, *Yahoo*), aparecerá na página um local para efetuar o login do usuário utilizando uma conta do *Facebook* ou Google, porque fizeram isso? Por que os microblogs são na atualidade uma das plataformas de redes sociais mais difundidas. Como no marketing por microblog, o marketing viral é uma das estratégias mais utilizadas. Entretanto, o marketing objetivo é outra opção a ser utilizada se a empresa reconhece as características e círculos sociais dos usuários.

Ting[53] propôs a arquitetura de um sistema de recomendação baseado nos dados extraídos de microblogs. A estrutura social de recomendação é conduzida de acordo com as mensagens e estruturas sociais de usuários alvo. A similaridade dos atributos conhecidos dos usuários e produtos são calculados como a essência de uma máquina de recomendação. A partir da análise dos resultados, podemos encontrar a diferença da

medição de análise de redes sociais (SNA) entre diferentes produtos. Portanto, isso mostra que o sistema de recomendação pode ser utilizado para recomendar diferentes produtos para consumidores-alvo.

É uma tendência combinar redes sociais com diferentes sites utilizando o sistema de recomendação. Até o momento, ainda não fomos capazes de coletar dados sobre e-commerce baseados na rede social para provar a influência dos conteúdos em redes sociais.

4.2 SVM na Rede Social

A tecnologia de aplicação da SVM tem sido desenvolvida por 15 anos, sendo amplamente utilizada em categorização de textos, reconhecimento de imagens, reconhecimento de dígitos escritos a mão [54]e bioinformática [55]. A maioria dessas tecnologias já estão disponíveis, entretanto, a aplicação de SVM para recomendação em redes sociais ainda estão no seu estágio inicial de pesquisa. Dito isso, é necessário explorar esse novo campo com urgência. Nessa seção, introduziremos algumas pesquisas na parte de recomendação baseado na tecnologia SVM.

Recomendador de Notícias

Digg é um site norte-americano que reúne links para notícias, podcasts e vídeos enviados pelos próprios usuários e avaliados por eles. Combina *social bookmarks*, *blog* e *feed*. O *Digging Digg* é um site de mineração de comentários, predição de popularidade e análise de redes sociais. Jamali [56] propôs que a utilização de informações de comentários disponíveis no *Digg* poderá definir uma rede de co-participação entre usuários através das informações de comentários disponíveis no Digg. Eles focaram na análise dessa rede implícita e estudaram as características comportamentais dos usuários. Eles também utilizaram atributos derivados de dados de comentários e redes sociais para predizer a popularidade de conteúdos atrelados ao Digg utilizando um framework de classificação e regressão.

O método de classificação que eles mencionaram é o classificador SVM. Os resultados de classificação mostraram uma alta eficiência do classificador SVM comparado ao DT e ao 9-NN(dois algoritmos). Não obstante, um par de usuários comentando na mesma história poderão apresentar diferentes opiniões e até visões opostas. Esse estudo não estava preocupado com essa questão, portanto, é crucial que eles refinem as definições de relacionamentos entre os usuários baseados na polaridade dos comentários.

Predição de Aplicativos Móveis

Wei Pan[57] propôs um problema sem precedentes de predizer instalações de aplicativos móveis utilizando redes sociais. Eles desenvolveram um método computacional simples que analisa as informações de redes sociais coletadas por sensores embutidos, e publicaram os resultados no mercado de aplicativos como a AppStore do iPhone para predizer a instalação de apps futuros. Eles utilizaram o classificador SVM como um método de comparação do seu método de abordagem. Embora os resultados do classificador SVM foram piores do que o método de Wei Pan, a abordagem do SVM foi o primeiro método de predição no domínio de aplicativos móveis.

Detector de Padrão

Nos últimos anos, o classificador SVM tem sido utilizado em detecção de padrões. Por exemplo, o detector de contas *Sybil* [58], detector de spam de correio eletrônico [59], detector de *Social Bookmarking Site* [60] e o detector de força de sentimento, Thelwall [61] possuem características em comum – a performance nos seus modelos possuem alta capacidade de detecção.

As contas Sybil são identidades falsas criadas para injustamente aumentar o poder de recursos de um único usuário. Yang descreve o seu esforço de detectar, caracterizar e entender as atividades de contas Sybil na rede social Renren (OSN). Eles utilizaram dados *ground-truth* sobre o comportamento de Sybils nas rede sociais para criar um detector Sybil em tempo real baseado em medidas. O resultado mostra que o classificador SVM é eficiente na absorção de 99% dos Sybils, com baixas taxas negativas e positivas. Porém, existe um problema incerto de que se os resultados obtidos poderão ser generalizados à todos os OSN.

A acumulação de spams sempre foi um grave problema na comunicação via e-mail. Filtros tradicionais de spam visam analisar o conteúdo de e-mail para caracterizar os atributos comuns aos spams. Entretanto, é claramente observado que a presença de estratégias de cercear esses filtros são constantes devido aos benefícios econômicos trazidos pelo envio de spams. Devido à essa situação, existe um amplo gama de pesquisa no campo de detecção de spams baseados na reputação dos remetentes, porém há poucos estudos na área de bloqueio de spams baseados em conteúdo. As pesquisas anteriores a esse artigo padecem de dois problemas principais: (1) o sistema não é totalmente funcional em meios diferentes, e (2) nenhum esquema de atualização é dado para perceber as mudanças de atributos envolvendo redes. Nesse sentido, o autor propõe um modelo de máquina de suporte vetorial (SVM) incremental para detecção de spam em uma rede dinâmica de correio eletrônico. Esse sistema foi engendrado para melhorar a adaptação em redes diversificadas.

Numerosos atributos de cada usuário em rede são extraídos para treinar um modelo de SVM, ademais, para dominar a natureza evolutiva da comunicação de correio eletrônico, apresenta-se um esquema de atualização incremental para eficientemente treinar um modelo de SVM. Contudo, existem dois problemas nos atributos extraídos: (1) eles ignoram o conteúdo do correio eletrônico. Na verdade, se não utilizarmos o conteúdo do e-mail como um método básico de sistema de detecção de correio eletrônico, é como se tivéssemos a receita, porém sem ingredientes para concluí-la. (2) Eles utilizam reciprocidade de comunicação (CR) como um atributo, eles apontaram que quando um usuário recebe um correio eletrônico, ele/ela normalmente irá responder ao remetente. Por outro lado, poucos usuários respondem spams. Eles ignoram o fato de que muitos usuários utilizam os seus correios eletrônicos para receber informações subscritas (por exemplo, receber informações de atualizações de capítulos em uma biblioteca online ou informações acerca de um vídeo favorito no youtube).

Confiança nas Predições de Usuários

A confiança entre um par de usuários é um valioso item de informação para usuários em uma comunidade *online* (como em sítios de comércio eletrônico e sítios de criticas de produtos) onde usuários podem fazer uso de informações de confiança para tomar decisões

e predizer se um usuário confia no outro. O método antigo consiste em inferir avaliações de confiança desconhecidos através de avaliações previamente conhecidas. A efetividade dessa abordagem depende na conectividade da rede conhecida de confiança e pode ser relativamente franca quando a conectividade é muito esparsa – que normalmente é o caso em comunidades online. Dito isso, Liu [62] propuseram uma abordagem de classificação para solucionar o problema de predição de confiança. Desenvolveram uma taxonomia para obter um set de atributos relevantes adquiridos de atributos e interações de usuários em uma comunidade online. Os resultados do experimento mostram que os classificadores SVM e Naive Bayes que utilizam a interação podem desempenhar melhor do que os sistemas de classificação que utilizam somente atributos. Se melhorarem esse classificador para se adaptar 'as evoluções na confiança – que por sua vez mudam dinamicamente através do tempo – para que o estudo se torne mais compreensivo.

Correspondência de Redes Sociais

Com o crescente número de pessoas com presença na web, mais máquinas de pesquisa provem resultados de pesquisa no nível objetivo, e.g. através da exposição de noticias correlatas, imagens, produtos e pessoas que frequentemente aparecem nos noticiários. Na vida real, nós sempre nos deparamos com nomes desconhecidos ou nomes familiares que queremos conhecer e procurar no *Twitter*. Mas o problema é que o feedback que procuramos disponibiliza vários resultados, devido a dupla aplicação do nome ou sobrenome. Então precisamos achar um método existente para extrair as informações.

Ferramentas existentes para esse fim são construídas a partir de uma máquina textual primitiva, e inevitavelmente, sofre de baixa precisão, devido aos falsos positivos e falsos negativos. Para superar essas limitações, You [63] alavancou evidências "relacionais" extraídas do corpus do *Web*. Em particular, como um exemplo, eles adotaram co-ocorrências de documentos de *Web*, que podem ser interpretados como um homólogo "implícito" da relação de seguidores no Twitter. Utilizando atributos relacionais e textuais, eles utilizaram o SVM para aprender uma função de nivelamento agregando esses atributos para uma ordenação acurada de correspondência de candidatos. Eles treinaram um classificador SVM utilizando os seis atributos. Entretanto, esse estudo não soluciona o problema da desambiguação de nomes, no qual delegamos para um trabalho futuro para investigar como atributos relacionais podem ser utilizados para fins de desambiguação.

Algoritmos de Recomendação em E-commerce

A maioria dos algoritmos de recomendação empregados no e-commerce recomendam todas as mercadorias, enquanto a maioria do mercado é tomado por uma parte pequena de produtos altamente populares.Li [64] apresentou um algoritmo de recomendação para mercadorias populares. Esse algoritmo constrói um modelo de atributos grupais de mercadorias populares para os consumidores utilizando os seus atributos pessoais e características comportamentais, depois, o relacionamento entre o modelo e a forma atual é minerada utilizando um algoritmo de regressão de máquinas de vetores de suporte. Comparado com algoritmos convencionais de filtragem colaborativa, esse novo algoritmo pode melhorar a exatidão da recomendação com menor erro médio absoluto.

Nesse artigo, os autores citados expuseram os passos para se chegar no algoritmo de recomendação em e-commerce. Existem três problemas que foram sumarizados. (1) Eles estabeleceram um modelo de regressão para cada mercadoria popular, porém esse cálculo é demasiadamente custoso para ser suportado. (2) Eles coletaram as informações das mercadorias populares para construir o seu modelo, porém, ignoraram o fato na função das mercadorias populares que a sua popularidade declina ao longo do tempo, portanto, não faz sentido recomendar mercadorias obsoletas. (3) Os pesquisadores supracitados selecionaram três atributos que são simples demais para construir um modelo de regressão.

Análise de Caráteres Utilizando SVM em Microblog

O Microblogging fornece uma nova plataforma para comunicar e compartilhar informações entre os usuários da Web. Usuários podem expressar opiniões e recordar acontecimentos diários através dos microblogs. O trabalho de Tian[65] frisa na utilização do Sina Weibo, a plataforma de microblog mais popular da China, para a análise de caráteres. Eles definiram quatro categorias de atributos através da análise de microblogs, e mostraram como coletar corpus com rótulos como dados de treinamento. Utilizando o corpus e via SVM, eles construíram um classificador de caráteres, que é capaz de determinar extroversão ou introversão para um microblog. As avaliações experimentais mostram que o seu método pode identificar o caráter do usuário de forma eficiente.

Os conjuntos de dados que eles utilizaram para mineração contém 200 id's de usuários e entre esses, 100 usuários são de extroversão e os demais são de introversão. O volume de dados não é suficiente para apoiar o experimento como um todo, pois é impossível controlar a distribuição na vida real com uma amostragem de apenas 200 usuários, sendo necessário pelo menos 10000. Eles também ignoram a definição de introversão e extroversão. Finalmente, os pesquisadores aqui mencionados deveriam coletar mais dados de referência além dos quatro atributos para se obter a predição.

4.3 Problemas Encontrados nas Abordagens Existentes e Solução

Baseados estudos nesta pesquisa, identificamos que SVM não foi utilizado na recomendação de redes sociais. Já apresentou-se um entendimento sobre o estado da arte de recomendações da SVM. Combinando-o com o estudo neste trabalho, encontramos uma solução para a disparidade de dados, *cold start* e volume de dados.

- Disparidade de dados: como o conjunto de itens disponíveis é extremamente grande (livrarias virtuais oferecem milhões de livros, por exemplo), a sobreposição entre dois usuários é quase inexistente. Ademais, até quando o número médio de avaliações de um usuário ou item é alto, eles são distribuídos de maneira desigual e normalmente são distribuídos de acordo com a lei do poder [66]. Levando em conta que a maioria dos usuários/itens receberam uma quantidade pequena de avaliações, um algoritmo de recomendação compreensivo deve considerar a disparidade de dados ao elaborar um sistema de recomendação [67].

- *Cold start*: Quando novos usuário entram no sistema, normalmente, não existem informações suficientes para produzir recomendações. As soluções mais utilizadas para esse tipo de problema se baseiam em técnicas de recomendação híbridas que combina conteúdo com dados colaborativos [68] [69] e as vezes são acompanhados por dados pessoais dos usuários. Um outro método consiste em identificar usuários individuais em diferentes serviços da web. Por exemplo, o Baifendian[70] desenvolveu uma técnica que pode acompanhar as atividades de e-commerce de usuários individuais, para que um usuário de cold start (arranque frio) em um site A possa obter recomendações de acordo com os seus dados nos sites B, C, D, etc.

Alguns destes problemas não podem ser resolvidos devido ao limite de dados, por exemplo, não será possível resolver o problema relacionado com as mudanças de preferências dos usuários que ocorrem com o passar do tempo. O conteúdo da seção 4.1 e da seção 4.2 será utilizado neste estudo.

Capítulo 5

Descrição dos Dados e Seleção de Atributos

Nesse Capítulo, introduziremos os dados selecionados para análise. Section 5.1 explícita porque foram escolhidos os dados do Tencent Weibo. Então, na Seção 5.2, são detalhados os dados originais fornecidos pela o congresso que chamado *Knowledge Discovery and Data Mining* (2012 KDD Cup)Figura (5.1), na seção 5.3 a estrutura e definição dos dados são apresentadas e por última, na seção 5.4, os atributos selecionados são exibidos.

5.1 Motivação para Coletar os Dados de Tencent Weibo

Os dados do experimento foram coletados do maior portal de microblog chinês, o Tencent Weibo[71]. Desde 2012, o site apresentava mais de duzentos milhões de usuários, produzindo cerca de quarenta milhões de itens de informação diariamente. Estes dados são fornecidos pela KDD Cup (Figura 5.1), que é uma competição anual de Mineração de dados e de descoberta de conhecimento, organizada pelo grupo ACM Special Intereset on Knowledge Discovery and Data Mining, a principal organização profissional em mineração de dados[1]. Dito isso, esse formato de dados foi escolhido pelas seguintes motivos:

Figura 5.1: Tencent 2012 KDD Cup[1]

1. O Microblog é a forma mais difundida de networking social que pode refletir de forma objetiva o valor da disseminação informacional devido ao seu peso numérico de usuários, trazendo consigo uma quantidade formidável de informação.

2. Mesmo Tencent Weibo não sendo a plataforma de microblog (como *QQ* ou *Twitter*) mais difundida do mundo, a quantidade maciça de usuários do Twitter dificulta com que o site seja utilizado como um padrão uniforme para mensurar diferenças entre usuários de países distintos. Ademais, esses dados não são claramente organizados de acordo com as diferenças linguísticas. Isso culminará em dispersão de dados, claramente prejudicando a precisão da mineração de dados. Na plataforma Weibo, somente dados em chinês são utilizados, provendo a homogeneidade cultural e linguística necessários para conduzir os processos mais acurados de mineração de dados.

3. Seu tamanho comparado a outros conjuntos de dados publicamente disponíveis, junto com uma ampla gama de informações acessíveis em diversos domínios como perfis de usuários, gráficos sociais e categorias de itens permite um alto grau de sofisticação no que tange a metodologia de análise de dados. Os usuários nesse conjunto de dados – na casa dos milhões – são providos de informações (informações demográficas, palavras-chave, histórico de seguidores; etc.) para gerar um bom modelo de predição.

4. O formato de recomendação: Ao utilizar seis mil personalidades de alta visibilidade ou grupos com ampla permeabilidade na sociedade, a recomendação aos usuários é maximizada para que dados de alta qualidade sejam extraídos, diminuindo a dispersão dos mesmos, e que portanto, reflete a superioridade do algoritmo SVM.

5.2 Terminologias do *Microblog*

Os dados de Tencent Weibo possuem cinco conceitos, que precisam ser elucidados para uma maior compreensão dos algoritmos, citados no item, tweet, retweet, comentário, seguido/seguidor :

ITEM: um item é um usuário específico na plataforma Tencent Weibo, podendo consistir em uma pessoa, organização ou grupo. Estes são selecionados e recomendados a outros usuários, usualmente celebridades ou organizações estabelecidas. O tamanho do conjunto de dados é de seis mil itens.

TWEET: um tweet é a ação de postar uma mensagem no sistema do microblog que também é utilizado para denominar a mensagem em si. No momento em que o usuário posta o tweet, seus seguidores o enxergarão na plataforma do Twitter.

RETWEET: Um usuário pode repostar um tweet e agregar comentários, com a finalidade de compartilhar o tweet com mais pessoas.

COMENTÁRIO: um usuário pode adicionar comentários a um tweet. O conteúdo dos comentários não será imediatamente compartilhado aos seguidores como um "tweet" ou "retweet", mas aparecerá no histórico de comentários atrelado ao tweet.

SEGUIDO/SEGUIDOR: Se um usuário B é seguido pelo usuário A então B é um seguido de A, e A é um seguidor de B.

5.3 Formato dos Dados

Os dados experimentais são adquiridos a partir de itens, dados acerca do comportamento de usuário, dados sobre relacionamento de clientes e dados de pessoas-chave. Os sets de treinamento registram o histórico de recomendações que sugere itens aos usuários. Há setenta e três milhões de conjuntos de treinamento, que por sua vez possuem 2.3 milhões de usuários nos dados de perfil do usuário. Arquivos de dados de item são divididos em duas partes: palavras-chave e categoria do item. O arquivo user action contém as estatísticas sobre as ações "at" (@) entre usuários em um certo numero de dias transcorridos. O arquivo user sns contém o histórico de seguidos de cada respectivo usuário (p.s., o histórico de usuários que foram seguidos por cada respectivo usuário), com cinquenta milhões de registros. A seguir está a formatação dos dados:

1. Formato de dados para treinamento

Formato de dados Para treinamento	(UserID)\t(ItemID)\t(Result)\t(Unix Timestamp)

Resultado: os valores são 1 ou -1, onde 1 é quando o usuário "UserID"aceita a recomendação do "ItemID"e -1 quando o usuário rejeita o item recomendado.

2. Formato de dados de perfil dos usuários

Formato de dados de Profile dos usuários	(UserID)\t(Ano de nascimento)\t (Sexo)\t(Numero de post)\t(Tag-Ids)

O ano de nascimento é selecionado pelo usuário no momento do seu registro no microblog. O gênero possui um valor integral de 0, 1, ou 2, que representam respectivamente; "indefinido", "masculino"ou "feminino". O número de tweets é uma integral que representa o montante de tweets que o usuário efetuou. Marcações são selecionadas por usuários para representar os seus interesses. Se um usuário tem como gosto alpinismo e natação, ele ou ela poderá selecionar "alpinismo"ou "natação"como sua marcação, também existe a possibilidade de não selecionar nenhuma marcação. A linguagem original das marcações não são utilizadas para fins do experimento, sendo estas substituídas por integrais codificadas.

3. Formato de dados de categoria de item

Formato de dados de Categoria de item	(ItemID)\t (Item-Categoria) \t (Item-Keyword)

Item-Category é um ordenamento "a.b.c.d", onde as categorias são hierarquicamente definidas pelo caractere "."e estes são listados de cima para baixo (p.s., a categoria 'd' é uma categoria dentro de 'c' que por sua vez é uma categoria dentro de 'b' e assim por diante). Os itens de palavras-chave contém as palavras-chave extraídas do perfil do usuário correspondente na plataforma Weibo, podendo este ser uma pessoa, organização ou grupo.

4. Formato de dados de ação dos usuários

Formato de dados de Ação dos usuários	(UserID)\t (Ação-Destino-UserID)\t(Numero de ação de at) \t(Numero de repost)\t(Numero de comentário)

Se o usuário A quiser notificar outro usuários sobre o seu tweet, retweet ou comentário, ele/ela deve usar o sinal (@) antes do nome do perfil do usuário que deseja ser contactado, por exemplo '@tiger' (aqui o usuário a ser notificado é o 'tiger'). Se um usuário A "retweetar"o usuário B 5 vezes, marcar '@' 3 vezes e comentar 6 vezes, então a expressão se resumiria em uma linha "A B 3 5 6"em user_action.txt.

5. Formato de dados de relacionamento dos usuários

Formato de dados de Relacionamento dos usuários	(Follower -UserID)\t(Item-Palavrachave)

O arquivo user_sns.txt contém o histórico de seguidos de cada usuário. Note que a relação supramencionada pode ser recíproca.

6. Formato de dados de palavra chave dos usuários

Formato de dados de palavra chave dos usuários	(UserID)\t(Palavrachave)

Palavras-chave são apresentadas na forma de "kw1:weight1;kw2:weight2;··· kw3:weight3". Essas palavras são extraídas dos tweets, retweets e comentários de um usuário, e podem ser utilizadas como atributos para melhor representar o usuário no modelo de predição. Quanto maior a significância de uma palavra-chave, maior será a sua atratividade em respeito ao usuário.

5.4 Atributos Selecionados

Na seção 5.2, foram especificadas as descrições dos dados fornecidos. De acordo come estes dados, escolhemos 9 atributos que podem afetar o resultado das recomendações. Estes 9 atributos são 9 vectores do modelo SVM. Sendo assim, nosso classificador possui 9 dimensões no espaço. A Tabela 5.1 mostra a razão de escolha de cada atributo.

Tabela 5.1: Atributos e razão escolhada

Gênero	Existem algumas regras no atributo de sexo. Por exemplo, a recomendação do sexo oposto é mais fácil de ser aceita.
Idade	Pessoas em diferentes faixas etárias possuirão diferentes idades mentais, seus interesses também diferem, Pessoas mais jovens preferem cantores famosos no lugar de pessoas de idade.
Atividade	Determina se o usuário está frequentemente online, as pessoas que estão online frequentemente deixarão um registro de histórico. Por exemplo, as páginas pessoais que visitaram. Podemos minerar informações significantes através disso.
Network relativo	De acordo com os mesmos amigos ou amigos de amigos, poderá se saber quem é conhecido na vida real ou possuem interesses mútuos.
Aceitação de usuário	Reflete as opiniões do usuário acerca do sistema de recomendação. Por exemplo, se um usuário não aceita uma recomendação, claramente percebe-se de que ele não aprecia o conteúdo do sistema de recomendação.
Similaridade de palavras	Nós utilizamos palavras-chave que foram mineradas dos posts do usuário e cada palavra-chave possui um peso individual para representar as suas preferências.
Categoria de item	De acordo com o princípio de classificação, classificamos o item em 300 categorias. Através disso, consideramos que os usuários possuem preferências específicas.
Aceitação de item	Reflete a popularidade do item, maior a popularidade o item possuir, maior a probabilidade de aceitação por parte do usuário.
Número de fãs do item	O numero de fãs de um item indica a sua influência social.

Em seguida, a formulação original foi classificada pela seleção de atributos na seção anterior Equação(5.1).

$$x_1 = ([x]_1 \cdots [x]_n)^T \tag{5.1}$$

Onde n é o número do atributo de característica em um objeto em classificação. Nesse estudo, o gênero, idade, etc, são considerados atributos de característica Equação(5.2).

$$X = (gender, age, activity, followeeac, followerac, category, fansnumber, relation, similarity)^T \tag{5.2}$$

Considere os seguintes objetos de classificação que são resultados de classificações conhecidas Equação(5.3):

$$T = (x_1, y_1), \cdots, (x_l, y_l) \tag{5.3}$$

Onde $x_i = ([x_i]_1, \cdots, [x_i]_n)^T$ e $y_i \in \gamma = -1, 1$ correspondem, respectivamente, a entrada e saída do ponto de treinamento. O formato do conjunto de treinamento para fins do experimento é o seguinte:

Para recomendar o item ao usuário, é preciso construir um modelo de características de acordo com a formato supracitado ($T = (x_1, y_1), \cdots, (x_l, y_l)$).

Após a classificação sistemática do problema, focaremos no modo de construir a estrutura de características. Os quatro gráficos estruturais abaixo mostram o formato das características Figura(5.2).

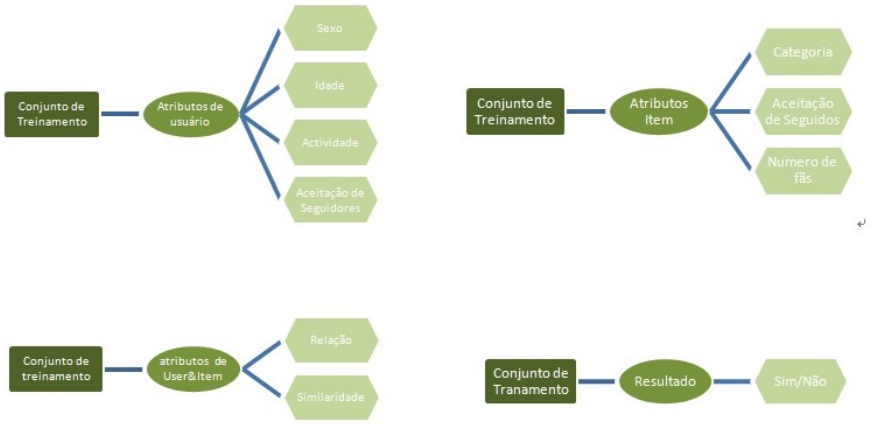

Figura 5.2: Estrutura de atributos

Esses quatro gráficos representam os atributos extraídos dos usuários, dos itens, dos usuários $\wedge$ itens e dos resultados, eles possuem três níveis. Se encontram no terceiro nível os atributos selecionados como idade, gênero, atividade, aceitação de seguidores, aceitação de seguidos, categoria, número de fãs, relação, similaridade.

Capítulo 6

Modelagem em Sistema de Recomendação pela SVM

Nesse capitulo, introduziremos como construir um modelo de um sistema de recomendação SVM. Na seção 6.1, nós introduzimos a estrutura integral do modelo. Na seção 6.2 nós expandimos o modelo SVM em dois módulos, e apresentamos os módulos de atributo. Na seção 6.3 apresentamos o módulo do algoritmo SVM.

6.1 Arquitetura do Sistema

Geralmente, existem duas etapas importantes que devem ser executadas em qualquer construção de um modelo de máquina de aprendizagem. O primeiro passo a ser considerado é determinar os atributos. Alguns atributos razoáveis foram selecionados pois podem afetar a aceitação da recomendação aos usuários. Uma seleção razoável afetará diretamente os resultados da classificação. O segundo passo é programar um algoritmo para realizar a classificação SVM. A Figura 6.1 mostra o modelo geral apresentado nesse estudo.

Os dados dos relacionamentos são "limpados" pelo processamento de dados e os dados relacionais são transformados para os dados de entrada do algoritmo SVM pelo módulo de atributo. O módulo SVM irá usar o seu algoritmo para resolver o problema de predição e classificação.

6.2 Modelagem de Atributos de Obtenção

Em seguida, tomando como base o conteúdo citado na seção 5.2, é necessário processar os dados, transformando-os em objetos de classificação padronizados a luz dos dados apresentados no KDDcup. Devido a grande quantidade de dados (setenta milhões), todos os atributos devem ser organizados de acordo com a ordem estabelecida no set de treinamento.

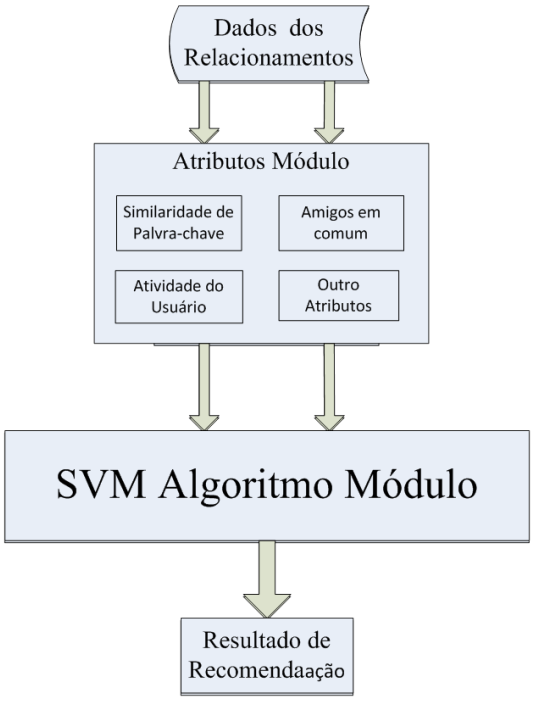

Figura 6.1: Dois módulos no sistema de recomendação: módulo de atributo e módulo SVM.

6.2.1 Gênero

Os dados originais são apresentados no formato numérico. É fácil perceber que existem apenas dois sexos no aparelhamento de dados. Se considerarmos o atributo "gênero" na sua forma numérica no conjunto de treinamento, a exatidão do experimento será afetada, então outro formato de dados atributivos é utilizado – o da enumeração (homem, mulher).

6.2.2 Idade

Assim como o gênero, o limite da idade no eixo vertical é de 0-100 anos. Podemos o dividir em cinco intervalos 0-14, 15-24, 25-34, 35-44, acima de 45.

6.2.3 Similaridade de palavras-chave

Similaridade: Os dados são atribuídos as palavras-chave dos usuários e também aos seus respectivos pesos, porém, é impossível atribuir dados diretamente a cada uma das seiscentas mil palavras-chave. Dito isso, o algoritmo de similaridade de cosseno de palavras-chave dado pela Equação 6.1 é utilizado para calcular a similaridade entre as palavras-chave dos usuários [72], cada instância corresponde a sua similaridade numérica.

$$Sim(a,b) = \frac{\sum_{p \in P} w_{ap} \cdot w_{bp}}{\sqrt{\sum_{p \in P} (w_{ap})^2} \sqrt{\sum_{p \in P} (w_{bp})^2}} \tag{6.1}$$

Onde o P representa o número de palavras-chave que o usuário possui, p representa uma variável contável, w_{ap} representa a palavra-chave pth do item recomendado.

Após os testes, encontramos muitas palavras-chave causadas pela disparidade de dados. Assim, a similaridade das palavras-chave entre usuários foi inferior a 1%, tornando-o incapaz de ser utilizado como um atributo. É importante notar que usuários possuem diversos itens nos dados de histórico. Relações indiretas podem ser obtidas dos itens do usuário e dos itens recomendados. Percebemos que há uma grande quantidade de palavras chaves nos itens recomendados. É possível calcular a similaridade entre os itens após inspeção e certificação, obtendo 10% de similaridade de dados. Dito isso, construímos um modelo mostrado na Figura 6.2.

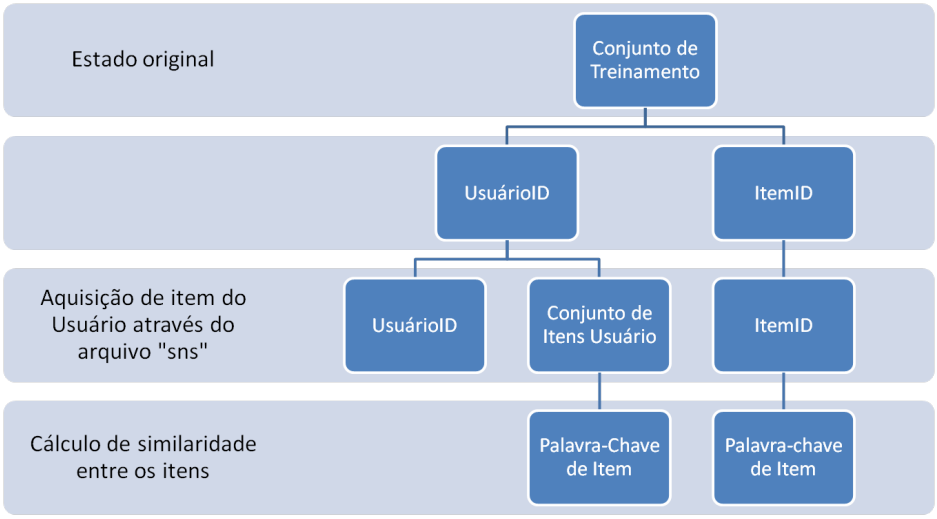

Figura 6.2: Fluxograma da obtenção de similaridade de palavras-chave

Existem quatro passos para se obter as palavras chaves. Primeiramente, obtém se a primeira linha do Conjunto de treinamento. Segundo, obtemos o userid e o itemid. Terceiro, examinamos o arquivo "sns" para achar os itens correspondentes. Quarto, obtemos as palavras chaves em comum utilizando a Equação 6.1.

6.2.4 Amigos em comum

Ao estudar os relacionamentos entre amigos no microblog, podemos utilizar o modelo de fãs para expressar os relacionamentos aqui analisados[73]. Primeiramente, será explanada a definição do modelo de fãs: definimos as concepções de "seguidor" e "seguido", designados como A e B, respectivamente. Se eu e A são fãs de B, então podemos utilizar gráficos direcionados baseados em teoria do grafo para explicar o relacionamento entre usuários e itens.

Definimos $G = (V, E)$, onde V representa os nódulos do usuário e $E : V \cdot V$ representa a conexão direcionada dos dois vértices f_{in},f_{out}, que expressam respectivamente, a função de follower(seguidor) e a função de followee(seguido). A função $f_{out}(A) = B|(A, B) \in E$ denota o conjunto de pessoas que foram seguidas pelo usuário A e a função $f_{in}(B) = A|(A, B) \in E$ denota o conjunto de seguidores de B. Podemos utilizar essas duas funções para expressar três relações de recomendação dos usuários A_1 e A_2; (1)

seguido em comum de dois usuários, a função $f_{out}(A_1) = B_1|(A,B) \in E \bigcap f_{out}(A_2) = B_2|(A,B) \in E$ representam os seguidos em comum de A_1 e A_2. (2) a função $f_{in}(A_1) = B_1|(A,B) \in E \bigcap f_{in}(A_2) = B_2|(A,B) \in E$ representa seguidores em comum de A_1 e A_2. (3) $f_{out}(A_1) = B_1|(A,B) \in E \bigcap f_{in}(A_2) = B_2|(A,B) \in E$ representa a intersecção dos seguidos do usuário A_1 e seguidores do usuário A_2. Como os itens do conjunto são compostos por celebridades e grupos já bem conhecidos, a relação (1) reflete apenas a similaridade de interesses entre A_1 e A_2. Na relação (2) o usuário não necessariamente conhece os seus fãs, portanto não há um relacionamento direto entre o seguidor e o seguido. Na relação (3), os seguidos dos seguidos de um usuário provavelmente dividem interesses com ele, portanto, a relação (3) é a mais apropriada para os dados desse experimento. Figura (6.3).

$$f_{out}(A_1) = B_1|(A,B) \in E \quad \bigcap \quad f_{out}(A_2) = B_2|(A,B) \in E \tag{6.2}$$

$$f_{in}(A_1) = B_1|(A,B) \in E \quad \bigcap \quad f_{in}(A_2) = B_2|(A,B) \in E \tag{6.3}$$

$$f_{out}(A_1) = B_1|(A,B) \in E \quad \bigcap \quad f_{in}(A_2) = B_2|(A,B) \in E \tag{6.4}$$

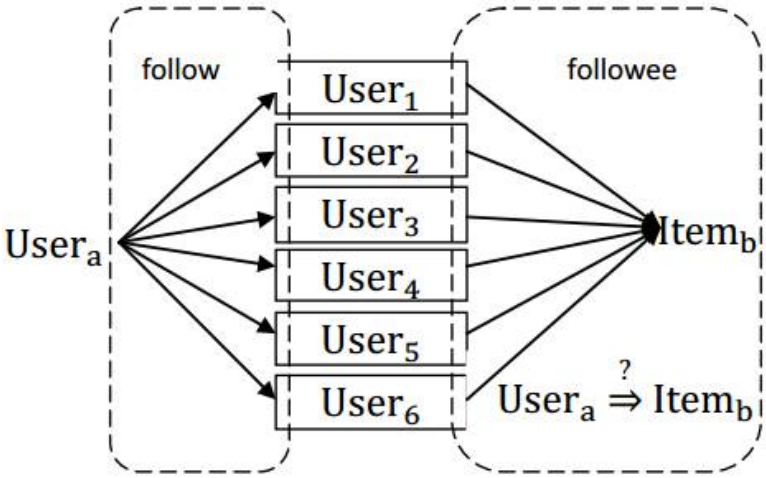

Figura 6.3: Amigos em comum

O $usurio_a$ da esquerda e $item_b$ da direita são dados através do conjunto de treinamento. Esse modelo tenta achar o usuário em comum entre o seguidor do $usurio_a$ e o seguido do $item_b$

6.2.5 Atividade

O nível de atividade de um usuário pode ser mensurado de acordo com o número de seus microblogs, número de "retweets" e assiduidade de comentários. Geralmente, a soma desses três fatores determinam o grau de atividade de um usuário, sendo este também o método mais simples. Porém, a técnica supracitada possui uma série de falhas, como na sua inabilidade de apontar os indicadores de maior impacto na atividade do usuário, como

alguém que "retuita" mil vezes mas possui menos de uma dúzia de tweets e comentários individuais. Esses fãs são chamados de "fãs- zumbi", são controlados por computadores e programas, não possuem sentimentos individuais, tendo somente a capacidade de criar "spam". Portanto, o seu número total é grande, porém são inativos. Existem dois caminhos para se chegar a uma correta avaliação do peso das atividades de um usuário. O primeiro caminho é baseado na experiência, dependendo nos conselhos dados pelos "experts". Como é impossível obter conselhos dos "experts" e depender na visão individual para uma avaliação subjetiva, utilizamos um segundo método, o do "Objective Weighting Method". Esse método utiliza o coeficiente de variação(CV) [74], que consiste em uma relação de variação de medidas para alcançar um índice médio em um set de dados. Os fatores comumente utilizados são o "Fator de Alcance" e o "Desvio Médio". Para fins do experimento, foi utilizado o coeficiente de desvio padrão expressado pela seguinte Equação(6.3).

$$CV = \frac{\sigma}{\mu} = \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{n}(x_i - \mu)^2}}{(x_1 + x_2 + \cdots x_n - 1 + x_n)/N} \tag{6.5}$$

Onde σ representa o desvio padrão e μ representa o valor médio. O valor de dispersão é refletido na sua unidade de valor médio. Os dois valores médios comumente utilizados não possuem o mesmo valor.

6.2.6 Categoria

Transformamos os dados brutos em uma categoria numérica Equação(6.4):

$$Category = 10^{n_i} + (0.5 \cdot 10^{n_i})X_i + (0.5 \cdot 10^{n_i})X_i + (0.5 \cdot 10^{n_i})X_i \tag{6.6}$$

Onde o i representa o nível em que se encontrava o item e n_i representa o expoente do nível. Por exemplo, se $i = 1$, então $n_i = 3$. As categorias de nível 2,3,4 excedem 10, então é necessário dividir pela metade o coeficiente do nível.

6.3 Modelagem do Otimização Sequencial Mínima

Otimização Sequencial Mínima (SMO) é um algoritmo para resolver o problema de otimização que surge durante o treinamento das máquinas de suporte vetorial. Porém, diferentemente do conteúdo discutido no Capítulo 2, SMO combina a parte teórica com uma atualização da aplicação iterativa otimizada a partir de grandes quantidades de dados Existem muitos pontos dos conjuntos de treinamento que se encaixam no espaço de treinamento multidimensional em problemas de vida real. Precisamos colocar todos os pontos nesse espaço para encontrar os vetores de suporte, portanto, precisamos de um algoritmo que primeiramente coloca os pontos no espaço, e depois atualiza os vetores de suporte que não satisfizerem as condições KTT. De acordo com a explicação na secção 2.3, conseguimos compreender a forma dual da Equação fatorial (6.4)

$$\begin{aligned} &Min_\alpha \psi(\vec{\alpha}) = min_\alpha \tfrac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j (\vec{x_I} \cdot \vec{x_J})\alpha_i \alpha_j - \sum_{i=1}^{N}\alpha_i \\ &s.t. \alpha_i \geq 0, i = 1, \cdots, n \\ &\sum_{i=1}^{n} \alpha_i y_i = 0 \end{aligned} \tag{6.7}$$

A função Min iguala a função max que obtivemos no final da secção 2.3

O algoritmo SMO escolhe dois fatores α a cada vez. Se eles não satisfazerem a condição KTT, precisamos atualizar α, temos a Equação (6.5) que é a atualização do limite inferior L_S em uma interação.

$$\begin{aligned} &L_S = \max_{\alpha}\{(\alpha_1+\alpha_2)+\textstyle\sum_{i=3}^{n}\alpha_i - \frac{1}{2}[\alpha_1\gamma_1\phi(X_1)+\alpha_1\gamma_1\phi(X_1)+\textstyle\sum_{i=3}^{n}\alpha_i\gamma_i\phi(X_i)]^2\} \\ &s.t.\alpha_1\gamma_1+\alpha_2\gamma_1 = -\textstyle\sum_{i=3}^{n}\alpha_i\gamma_i, 0<\alpha_i<C, \forall i \end{aligned} \tag{6.8}$$

as funções de atualização α_1^{new} e α_2^{new} são as Equação (6.6) e (6.7) respectivamente:

$${\alpha_1}^{new} = {\alpha_1}^{old} + \gamma_1\gamma_2({\alpha_2}^{old} - \alpha^{temp}) \tag{6.9}$$

$${\alpha_2}^{new} = {\alpha_2}^{old} - \frac{\gamma_2(e_1-e_2)}{\eta} \tag{6.10}$$

Onde $e = \mu - \gamma$, $\eta = \gamma(x_1,x_1)+\gamma(x_2,x_2)-2\gamma(x_1,x_2)$. O algoritmo SMO elege somente dois fatores duais al α_i,α_j para ajustar cada interação – os outros pontos permanecem fixos. Depois de obtermos os resultados de α_i,α_j, nós o utilizamos para aprimorar os demais pontos. Apesar deles requererem mais interações comparado com o algoritmo de decomposição usual, eles necessitam menos cálculo computacional em cada interação. Por isso, o algoritmo exibe convergência rápida, ademais eles não precisam armazenar o núcleo da matriz e também não necessitam de operações matriciais.

Capítulo 7

Implementação

Este capítulo descreve o funcionamento e a aplicação do modelo SVM para o sistema de recomendação.

7.1 Ambiente de Desenvolvimento

O sistema foi desenvolvido com o uso da linguagem Java, considerada uma importante linguagem de programação orientada a objetos, fundamental para esta implementação. Para tanto, foi utilizada a IDE (Integrated Development Environment) Eclipse - versão de release para a tarefa de codificação e o compilador java para a geração do arquivo executável. E o ambiente de desenvolvimento utiliza as instalações do Laboratório de Pesquisa da Universidade de Brasília, ou simplesmente TransLab(Tabela 7.1)

Tabela 7.1: Instalação experimental

Modelo	HP Pavilion p7-1060br PC
CPU	Intel core i5-2300 2.80G
Memória	DDR3, 4GB
HD	1TB
Sistema	Win7 Home premium 64Bits

7.2 Descrição do Sistema

O modulo do Sistema de atributos e o algoritmo SVM foram apresentados no capítulo 6. Esta seção descreve o processo especifico realizado no Eclipse e como os resultados foram obtidos pelo Weka. A figura a seguir (Figura 7.1) mostra as etapas de desenvolvimento realizadas que é detalhe da figura (6.1) :

- Os dados fornecidos pela KDD Cup 2012 são processados utilizando o Eclipse;
- Os dados processados são inseridos como entrada para o módulo de atributos e gera 9 atributos de perfil de todos os usuários e itens como saída. Armazena estes dados como dados de background;

- Seleciona os parâmetros do SVM;
- Java é utilizado como linguagem de programação, Eclipse como o ambiente de desenvolvimento. Weka é utilizada como a plataforma de trabalho, implementando o algoritmo SVM;
- O resultado é exibido na interface do Weka;

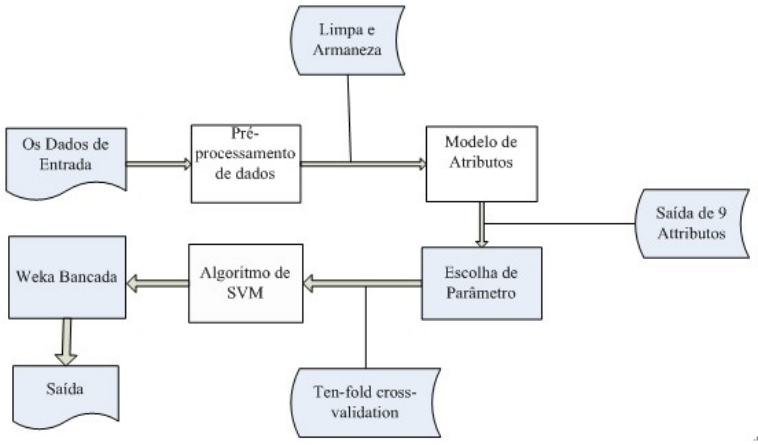

Figura 7.1: Cenário de operação do sistema SVM

7.2.1 Pré-processamento de dados

O pré-processamento de dados é uma parte importante deste trabalho. Métodos de processamentos de dados são fundamentais, como por exemplo: limpeza de dados, redução, remoção de dados incompletos, ruídos e registros redundantes. É importante enfatizar que a etapa de limpeza de dados é a fase central no processamento de dados. E possível observar os dados seguintes analisando os dados, exceto pelos dados que necessitam de alguma programação para adquirir alguns atributos, como, por exemplo, similaridade de palavra-chave, amigos em comum, etc.

Primeiramente, os dados devem ser analisados a partir da situação geral, a Figura 7.2 é dividida em duas zonas, a zona vermelha e a zona azul, que mostram os usuários que rejeitaram ou ignoraram as recomendações e os usuários que aceitaram as recomendações, respectivamente, 93% dos usuários rejeitaram ou ignoraram e somente 7% aceitaram.

Existem várias razões para a ocorrência deste fenômeno, primeiramente, o algoritmo de recomendação não foi adaptado para os dados reais. Em segundo lugar, cada pessoa possui uma personalidade diferente da outra, algumas são extrovertidas e outras são introvertidas. As pessoas introvertidas não aceitam facilmente as recomendações. Em terceiro lugar, o mecanismo do Microblog Tencent mostra as recomendações no lado direito de sua página (Figura 7.3). Caso o usuário clique no botão "seguir", a recomendação foi bem-sucedida, se não clicar, significa que a recomendação foi malsucedida. O problema é que o usuário pode não notar a coluna de recomendação, mesmo que exista alguém

Figura 7.2: Distribuição dos resultados das recomendações no conjunto de treinamento.

Figura 7.3: Página principal do microblog

que o interesse. Este mecanismo aumenta os erros, afetando a aceitabilidade. Estes três aspectos da recomendação podem afetar os resultados, gerando um deslocamento.

Muitas vezes são encontradas distribuições de dois tipos de amostras que estão desequilibradas [Min Liu and chengde]. Caso não seja adotada nenhuma estratégia para lidar com isso, as regras de classificação geradas irão tender para uma das amostras, no caso extremo, todas as amostras que pertencem à classe preponderante serão julgadas como positivas e julgar que como negativas as amostras que pertencem a outras classes. Obviamente esta classificação não está correta.

As seguintes estratégias são utilizadas atualmente para este problema: A primeira é manter o equilíbrio entre os dois tipos de amostras no conjunto de treinamento de forma artificial, por exemplo, selecionando novamente a classe preponderante para que seu tamanho se aproxime do tamanho da outra classe [75]. A segunda é manter a situação original de desequilíbrio na amostra e ajustar os pesos de penalidade para os dois tipos de erros[76]. A terceira é ganhar um equilíbrio aproximado entre os dois tipos de amostras, aumentando o número de amostras "dummy" na classe de minoria[77].

Luo e Peng [78] apontaram que é necessário que estratégias que não aumentem o número de amostras "dummy" sejam adotadas em primeiro lugar. Se de fato for necessário adotar este tipo de estratégia, deverão ser consideradas estratégias que aumentem o número de amostras "dummy" do vetor de suporte a priori. Portanto, deverá se reduzir o

número de amostras negativas para alcançar uma razão de aproximadamente 1.

Além dos dados não serem balanceados, existe um outro problema com o conjunto de dados: ele é muito grande. Existem 73.209.277 instâncias para os dados de treinamento experimental. A capacidade de dados para processamento é de 4GB. Para a mineração de dados, a qualidade dos dados é mais importante que o tamanho dos dados. Um conjunto de dados de alta qualidade não contém ruídos e registros perdidos, ele pode ser agrupado e é escalável[78]. Quanto maior a quantidade de dados, maior será a quantidade de ruídos, portanto, é necessário remover os ruídos para obter melhores resultados.

No Microblog, aparecem três recomendações no layout da página assim que o usuario realiza o login no Microblog. Se o usuário realiza o login no Microblog todos os dias em um mês, o que significa que a pessoa é um usuário frequente, então seu conjunto de treinamento registra anotações no seu histórico de login 90 vezes. Olhando a Figura 7.4 é possível observar que neste conjunto de dados, o número de recomendações não ultrapassou 100 vezes.

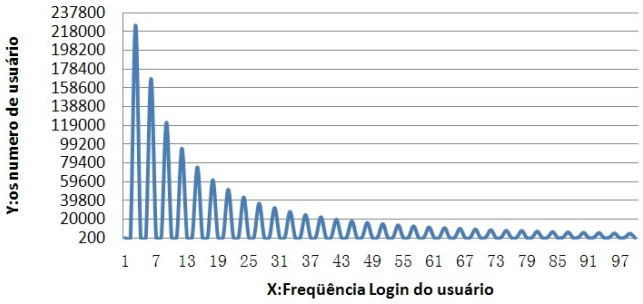

Figura 7.4: Distribuição de frequência de login

Então é necessário retirar todos os usuários cujo número de recomendações foi menor que 100 vezes, ou seja, retirar os usuários que não utilizar o microblog frequentemente. Isto irá diminuir bastante a quantidade de cálculo necessário para os dados e a interferência gerada pelos usuários infrequentes, também irá aumentar a precisão dos resultados de treinamento. Em seguida, um pré-processamento de dados passo a passo sugerido por Han[79] é adaptado para este estudo. Os passos são: diminuir os ruídos dos dados, resolver inconsistências, eliminar os atributos menos significativos, integrar os dados, transformar os dados, etc.

7.2.2 Weka

O weka foi utilizado como plataforma de trabalho para alcançar o SVM. O pacote de software Weka (Waikato Environment for Knowledge Analysis - Figura 7.5) é um difundido sistema de aprendizagem de máquina escrito sob a plataforma Java, desenvolvido na Universidade de Waikato na Nova Zelândia. Weka é um software gratuito disponível sob o GNU (General Public License). O workbench Weka contém uma coleção de ferramentas de visualização e algoritmos para análise de dados e modelagem preditiva, junto com interfaces gráficas de usuários para fácil acesso e funcionalidade, as vantagens do Weka incluem:

- Livre disponibilidade gratuita sob o GNU General Public License;

- Portabilidade, pois é totalmente integrado à plataforma Java podendo ser adaptado a qualquer plataforma de computação moderna;
- Uma coleção vasta de pré-processamento de dados e técnicas de modelagem;
- Fácil utilização devido à simplicidade da interface gráfica.

7.2.3 Formato de entrada

De acordo com o formato do entrada de dados ARFF, utiliza-se um algoritmo próprio para mineração de dados. Se existir a necessidade de escrever um algoritmo próprio, pode se consultar um interface de documentação para alcançar uma visualização mais intuitiva.

O Weka utiliza um arquivo de entrada no formato ARFF, composto por uma tabela bidimensional, cada linha horizontal da tabela representa uma instância, cada coluna representa um atributo. De acordo com a descrição dos dados de aquisição acima, transformamos todos os dados no formato da Figura 7.6.

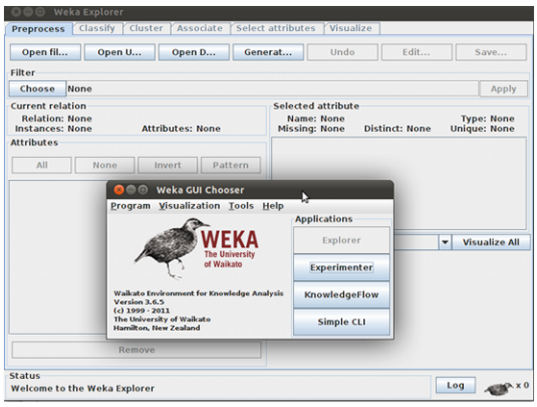

Figura 7.5: Formato de entrada de weka

7.2.4 Parâmetros do aprendizado

O classificador SVM pode ser afetado por vários parâmetros, dois deles são os fatores mais críticos: o parâmetro C de regularização. Ele funciona sobre um acordo entre amostra erroneamente classificada e complexidade do algoritmo. Regular a proporção do intervalo de confiança e risco empírico no aprendizado de máquina para melhorar a sua capacidade de generalização. Ao ser selecionado por um problema específico, o parâmetro C depende da quantidade de ruídos no processamento de dados para determinar o subespaço de características. Quanto menor for o valor de C, menor o coeficiente de erro empírico. Cada subespaço de característica possui pelo menos um C que permite a otimização da habilidade de generalização do SVM. Quando C excede um determinado valor, a complexidade do SVM alcança o valor máximo permitido pelo subespaço de característica. Nesse momento, o risco empírico e a capacidade de generalização são quase os mesmos. O formato da função do núcleo e os seus valores de parâmetro, juntamente com diferentes funções de núcleo afetam a performance da classificação, funções de núcleo

iguais com parâmetros diferentes também são afetados. Na função de núcleo polinomial, o parâmetro é d. Na função de núcleo *Radial basis function kernel* (RBF) o parâmetro é σ e na função de núcleo sigmoide o parâmetro é d. A escolha das funções de núcleo e parâmetros afetam diretamente a qualidade do classificador SVM.

O método de escolha dos parâmetros são essencialmente um problema de otimização. Atualmente, os métodos são: método de seleção por experiência, teste experimental, método de erro, método de gradiente descendente, método de validação cruzada e método bayesiano. Entretanto, com o sucesso dos algoritmo genéticos, da otimização de enxame de partículas e de outros métodos inteligentes de organização, alguns especialistas utilizam esses métodos como um meio de otimizar os parâmetros do SVM. Recentemente, com a ascensão da computação evolucionária, um novo algoritmo de otimização – consistindo na estimativa dos algoritmos de distribuição – rapidamente se tornou foco das pesquisas e um método eficiente para resolver questões de engenharia no campo da computação evolucionária. Outros modelos também foram introduzidos ao método SVM de seleção de parâmetros. A validação cruzada é uma delas e é um dos métodos mais utilizados por ser de fácil implementação, mas é matematicamente pesado, especialmente para problemas amostrais grandes. Ao utilizar parâmetros de suporte, as máquinas de vetores foram otimizadas a fim de suprir os problemas apresentados.

A correta escolha dos parâmetros de aprendizado consiste em um passo fundamental na aquisição de máquinas de suporte vetorial bem calibradas. Usualmente, as configurações desses parâmetros se baseiam na busca em redes [80]. Um algoritmo de busca em redes deve ser guiado por alguma contagem de desempenho, tipicamente mensurado por validação cruzada no conjunto de treinamento [81]. A biblioteca Libsvm pode otimizar automaticamente o C e o gamma.

7.3 Procedimentos de Módulo

Esta seção apresenta o pseudocódigo do módulo descrito no capítulo 6. Todos os pseudocódigos foram implementados utilizando o Eclipse. Existem quatro algoritmos: Amigos em comum, similaridade de palavras-chave, atividades e SMO.

7.3.1 Algoritmo de amigos em comum

O modelo de amigos em comum foi descrito na seção 6.1.2, de acordo com esta ideia, este algoritmo foi criado. O algoritmo consiste de três passos. Baseando-se no tamanho do conjunto de treinamento para executar o cálculo no loop:

- Ler a primeira linha no conjunto de treinamento, obtendo o userID e itemID;
- Encontrar o userID correspondente no arquivo "use's followers". Enquanto isso, encontrar o itemID correspondente no arquivo "item's followees";
- Computar os amigos em comum.

```
Data: conjunto de treinamento, sns de usurios, sns de item
Result: Amigos em comum
initialization;
while No nulo do
    ler a linha de arquivo "treinamento";
    obter userID;
    obter itemID;
    while No nulo do
        ler a linha de arquivo"sns de usuários";
        obter userID;
        obter follower de user;
        while No nulo do
            ler a linha de arquivo"sns de item";
            obter itemID;
            obter followee de item;
            calcula o Amigos em comum;
        end
    end
end
```

Algorithm 1: Algoritmo1 Amigos em comum

7.3.2 Algoritmo de similaridade de palavras-chave

Da mesma forma, o modelo mostrado na seção 6.1.1 deu origem a um algoritmo. O algoritmo possui três etapas:

- Ler a linha do conjunto de treinamento, obtendo o userdID e itemID;
- Encontrar tais itens no arquivo “user’s followers”;
- Calcular a similaridade de palavras-chave entre cada par de itens de usuário e itens de treinamento, somando todos os valores de similaridade.

Data: conjunto de treinamento, palavras-chave dos usuários, palavras-chave dos itens
Result: Similaridade de Palavras-chave
initialization;
while *Não nulo* **do**
  ler a linha de arquivo "conjunto de treinamento"obter o userID e itemID obter o set de item de userID
  while *Não nulo* **do**
    ler a linha de arquivo "palavras-chaves dos itens"**if** *itemID de conjunto de treinamento igular com itemID de arquivo "palavras-chave dos itens"* **then**
      calcular a similaridade de palavras-chave entre itemIDs ;
      similaridade de palavras-chave <-pointMulti(vector1, vector2)/sqrtMulti(vector1, vector2)
    else
      voltar para o início da seção atual
    end
  end
end

Algorithm 2: Algoritmo2: Similaridade de Palavras-chave

7.3.3 Algoritmo de atividade

O terceiro algoritmo calcula o grau de atividade. Ele acontece em três passos:

- Calcular o peso CV das postagens , comentário e retweet.
- Ler a linha do conjunto de treinamento, obtendo o userID;
- Calcular o grau de valor do usuário.

Data: conjunto de treinamento, ação de usuários
Result: atividade dos usuários
initialization;
while *Não nulo* **do**
    ler a linha de arquivo "ação de usuários";
    número de post de CV (npcv)<- sqr sum pow((xi-u)/xN)/média de número de post;
    número de comentário de CV(nccv) <- sqr sum pow((yi-u)/yN)/média de número de comentário;
    número de retweet de CV(nrcv) <- sqr sum pow((zi-u)/zN)/média de número de retweet;
    peso de post <- npcv/npcv+nccv+nrcv;
    peso de comentário <- nccv/npcv+nccv+nrcv;
    peso de retweet <-nrcv/npcv+nccv+nrcv;
    close ação de usuários
end
while *Não nulo* **do**
    ler a linha de arquivo"set de treinamento";
    obter userID;
    while *Não nulo* **do**
        ler a linha de arquivo"ação de usuários";
        actividade <- (post mul peso de post)+(comentário mul peso de comentário)+(retweet mul peso de retweet);
    end
end

Algorithm 3: Algoritmo1 Atividade

Capítulo 8

Estudo de Caso

O estudo de caso é baseado os treinamentos de dados feito utilizando o SVM para a recomendação de amigos. Os treinamentos são elaborados a fim de se detectar a precisão do SVM em comparação com outras duas técnicas (*Naive Bayes*, *Random Forest*) e os principais fatores de influência na Weibo para recomendação de amigos.

Este capítulo apresenta os resultados obtidos com a utilização do SVM para prever a aceitação de recomendação de amigos. A Seção 8.1 descreve o planejamento de uso dos dados e a motivação de experimento detalhado. A Seção 8.2 apresenta os resultados obtidos com os testes realizados no Caso 1, onde houve uma variação no tamanho da amostra e a seção 8.3 apresenta os resultados na aplicação do Caso 2, que ocorreu com a variação dos atributos escolhidos. Por fim, a Seção 8.4 encerra este capítulo com uma breve discussão dos resultados.

8.1 Planejamento do Estudo de Caso

Esta seção irá apresentar o planejamento dos experimentos para verificar a eficiência e sensibilidade da presente proposta. Primeiramente, são apresentados dois casos para estudo para que seja possível avaliar o desempenho do algoritmo SVM e os atributos selecionados. Em seguida, são apresentadas as quatro métricas utilizadas no experimento para avaliar esta proposta. Por último, são apresentados dois algoritmos utilizados para comparar a eficiência e desempenho do SVM, naive bayes e random forest, ambos são algoritmos clássicos na área de mineração de dados.

8.1.1 Casos avaliado

No capítulo anterior foi descrito o processamento de dados e 40 milhões de amostras foram selecionadas dos dados originais. Todos os exemplos deste capítulo foram realizados com extrações aleatórias de dados destas 40 milhões de instâncias reais. A avaliação de dois casos foi considerada:

1. Caso 1:Diferentes tamanhos de amostras

   Inicialmente, a recomendação de amigos foi testada utilizando a metodologia proposta – SVM- para uma rede social online (Weibo). Devido ao fato da base de dados possuir um tamanho consideravelmente grande, 40 milhões de amostras, é

necessário para que os testes possam ser realizados dentro do tempo necessário, que somente uma parte destes dados sejam considerados. Porém, é necessário selecionar estes dados de forma que as amostras escolhidas ainda sejam capazes de representar a totalidade dos dados. Para que isso seja possível, as amostras foram escolhidas de forma aleatória, e considerou-se a média da aplicação do algoritmo SVM para 3 amostras diferentes.

No primeiro experimento, para cada teste realizado, aleatoriamente foram escolhidos um tamanho de amostra diferente (1mil 5mil 10mil 20mil 30mil 40mil 50mil 60mil 70mil 80mil 90mil 100mil) para treinamento. Para cada um destes tamanhos, 3 amostras diferentes foram testadas para garantir a consistência dos resultados. Foram duas razões para selecionar esses doze conjuntos de dados.A primeira delas é o caráter fortuito deles, selecionamos doze conjuntos de dados para prevenir que os erros surjam do caráter fortuito deles. A segunda razão é que escolhendo diferentes conjuntos de dados de tamanho 1 mil até 100 mil é possível testar o desempenho do classificador SVM em diferentes tamanhos de conjuntos de dados.

Também é importante que amostras pequenas sejam selecionadas, devido ao fato de que em alguns casos, não se tem acesso à amostras muito grandes para que seja feita a recomendação de amigos, nesta proposta são testados valores pequenos de amostras para verificar se o modelo é capaz de ser eficiente e se possui um com resultado em casos em que não se possui acesso à uma quantidade significativa de amostras. Caso os resultados com a aplicação do algoritmo SVM para pequenas amostras sejam tão eficientes quanto para amostras maiores, então será possível utilizar o SVM diretamente em amostras menores, por questões de desempenho computacional e espaço de armazenamento.

Para verificar o desempenho do algoritmo SVM para a recomendação de amigos em relação às atuais técnicas de mineração de dados utilizadas, dois algoritmos foram utilizados e aplicados em cima da base de dados utilizada para que possa ser realizada uma comparação com os resultados obtidos com o SVM. Os algoritmos utilizados para comparação foram: Naive Bayes e Random Forest e eles foram aplicados nas mesmas amostras nas quais foram aplicados o algoritmo SVM.

2. Caso2:Identificaão dos atributos de influência

   Este estudo de caso foca em testar o desempenho do algoritmo SVM para diferentes atributos dos perfis do usuário. Alguns fatores podem ter maior influência na recomendação de amigos do que outros, por exemplo, se for considerada uma rede social cujos usuários são em suas maiorias brasileiras, o atributo "país" não faria uma diferença significativa no resultado das recomendações.

   A importância deste caso é identificar quais os atributos realmente exercem uma influência na decisão da aceitação ou rejeição de uma recomendação de amigos neste conjunto de dados. O ideal é selecionar a menor quantidade possível de atributos fundamentais para que o sistema apresente um bom desempenho utilizando a menor quantidade possível de dados.

   Para os testes realizados, foi considerada uma amostra de tamanho 50 mil, devido ao fato deste tamanho ser centralizado em relação aos tamanhos de amostras testadas anteriormente. Os atributos testados dos usuários foram: atividade, idade,

categorias, aceitação de seguidos, aceitação de seguidores, gênero, número de fãs, similaridade de palavras-chaves, amigos em comum.

Os testes foram realizados removendo cada um dos atributos anteriores dos experimentos para verificar a queda dos resultados, indicando que o respectivo atributo possui uma grande influência no resultado final. Em cada experimento um atributo é retirado e os cálculos são realizados na ausência daquele atributo.

Similarmente ao Caso 1, as mesmas amostras testadas para o algoritmo SVM, foram também testas com Naive Bayes e Random Forest para verificar a consistência dos atributos selecionados e também confirmar se os atributos se comportam da mesma maneira para os diferentes algoritmos, confirmando se a escolha dos atributos foi adequada.

8.1.2 Medidas e avaliação

Esta seção apresenta as medidas utilizadas para avaliar os algoritmos executados no Caso 1. São quatro medidas: Precisão, Sensitividade, F-measure e CCI.

Em uma classificação, a precisão para uma classe é o número de verdadeiros positivos (ou seja, o número de itens corretamente classificados como pertencentes à classe positiva) dividida pelo número total de elementos rotulados como positivos (ou seja, a soma dos verdadeiros positivos e falsos positivos, que são itens incorretamente rotulados como positivos). A precisão determina a fração de registros que realmente são pertencentes à classe positiva em um grupo que foi considerado positivo pelo classificador. A Equação 8.1 mostra como é computada a precisão.

$$Preciso = \frac{tp}{tp + fp} \tag{8.1}$$

A Sensitividade neste contexto é definida como o número de verdadeiros positivos dividido pelo número total de elementos que de fato pertencem à classe positiva (ou seja, a soma de verdadeiros positivos e falsos negativos, ou seja, itens que deveriam ser considerados como positivos, mas não foram). A sensitividade é a fração de instâncias classificadas corretamente como positivas dentre todas as que realmente são positivas. A Equação 8.2 mostra como é calculada a sensitividade.

$$Sensitividade = \frac{tp}{tp + fn} \tag{8.2}$$

Neste estudo, tp representa o número de recomendações corretamente rotuladas como positivas. Fp representa as recomendações incorretamente rotuladas como positivas. Fn representa as recomendações que não foram classificadas como pertencentes à classe positiva incorretamente.

A F-measure pode ser interpretada como uma média harmônica da Precisão e da Sensitividade, onde uma pontuação F-measure chega a seu melhor valor em 1 e pior valor em 0. F-measure será alto somente se os valores de Precisão e Sensitividade forem razoavelmente altos. A Equação do F-measure é apresentada a seguir (Equação 8.3):

$$F - measure = \frac{Preciso * Sensitividade * 2}{Preciso + Sensitividade} \tag{8.3}$$

A CCI(Corrected Correctly Classified Instances – Instâncias Corretamente Classificadas) ou exatidão é uma medida calculada pela razão entre a soma das instâncias verdadeiramente positivas e as verdadeiramente negativas, ou seja, todas aquelas classificadas corretamente e a o tamanho total da amostrado, dada pela soma de todas as instâncias classificadas corretamente e incorretamente. A Equação 8.4 apresenta o cálculo da CCI:

$$CCI = \frac{tp + tf}{tp + fn + tn + fp} \tag{8.4}$$

8.1.3 Métodos de comparação

O presente modelo foi comparado com dois algoritmos, apresentados a seguir:

- *Naïve Bayes*: O classificador Naïve Bayes é um classificador probabilístico simples baseado na aplicação do Teorema de Bayes com afirmações fortes e independentes. Um termo mais descritivo para o modelo probabilístico fundamental seria "modelo de aspectos independentes".
- *Random Forests*: São conjuntos de métodos de aprendizagem para classificação (e regressão) que operam através da construção de múltiplas árvores de decisão durante a etapa de treinamento e então apresentando como saída a classe utilizando o modo de saída de classes em árvores individuais.

Estes dois algoritmos foram escolhidos para medição e comparação com o algoritmo SVM devido ao fato de que ambos são algoritmos populares e utilizados como referência para diversos trabalhos, também são algoritmos bastante utilizados para análise comparativa. Além disso, estes dois algoritmos foram escolhidos pelas autoridades internacionais de organização acadêmica como dois dos 10 algoritmos mais influentes da área de mineração de dados.

8.2 Resultados do Caso 1

Esta seção apresenta os resultados obtidos com a aplicação do algoritmo SVM no Caso 1, variando o tamanho das amostras de 1 mil a 100 mil. A seção 8.2.1 apresenta o resultado final dos testes, a seção 8.2.2 mostra a comparação realizada com os outros algoritmos e a seção 8.2.3 apresenta uma análise dos resultados obtidos nas seções anteriores. A Tabela 8.1 mostra os parâmetros considerados para os testes realizados no Caso 1.

Tabela 8.1: Parâmetros de treinamento

Conjunto de dados	40 milhões de instâncias
Algoritmos para comparação	Classificador **Naive bayes, Random Forest**
Número de experimentos	72
Método de treinamento	10-*Fold Cross Validation*

8.2.1 Resultados do algoritmo SVM

Os testes foram realizados com a aplicação do algoritmo SVM nos dados da rede social Tencent Weibo. A Tabela 8.2 apresenta os resultados obtidos, na primeira coluna está o número de instâncias utilizado, ou seja, o tamanho da amostra, na segunda coluna está o resultado do cálculo da precisão da recomendação resultante, a terceira coluna apresenta o resultado do cálculo da Sensitividade. Na quarta coluna está o resultado da F-measure e, por fim, na última coluna é apresentado o resultado da CCI ou exatidão da recomendação de amigos.

Tabela 8.2: Resultado de SVM

Instâncias	**Precisão**	**Sensitividade**	**F-measure**	**CCI**
1mil	0,731	0,712	0,706	0,712
5mil	0,771	0,747	0,741	0,746
10mil	0,775	0,75	0,745	0,745
20mil	0,765	0,746	0,741	0,746
30mil	0,774	0,756	0,752	0,756
40mil	0,763	0,746	0,742	0,746
50mil	0,768	0,752	0,748	0,752
60mil	0,766	0,75	0,746	0,75
70mil	0,769	0,754	0,751	0,754
80mil	0,769	0,755	0,752	0,754
90mil	0,77	0,756	0,752	0,756
100mil	0,766	0,751	0,748	0,751

Pela Tabela 8.2 é possível observar que o algoritmo apresenta resultados semelhantes de Precisão, Recall, F-measure e CCI para diferentes tamanhos da amostra, ou seja, o tamanho da amostra não interferiu de forma extrema no desempenho do algoritmo.

A melhor precisão foi obtida para 10 mil instâncias, com um valor de 0,775, o maior valor de sensitividade foi obtido para uma amostra de tamanho 90 mil, no valor de 0,56. O maior valor de F-measure obtido foi no valor de 0,752 para um número de instâncias de 30, 80 e 90 mil. Por fim, o CCI com melhor resultado foi de 0,756 para amostras de tamanho 30 mil e 90 mil. Pela Tabela 8.2, pode se concluir que o algoritmo SVM apresenta um desempenho consistente para diversos tamanhos de amostras. Os valores de precisão vão de 0,731 a 0,775, os valores de Sensitividade vão de 0,712 a 0,756, os valores de F-measure vão de 0,706 a 0,752 e o CCI varia de 0,712 até 0,756, sendo assim, a maior variação é de 0,046, considerada uma variação muito pequena.

8.2.2 Comparação de algoritmos

As mesmas amostras aplicadas ao algoritmo SVM foram aplicadas também aos algoritmos Naive Bayes e Random Forest, conforme mostrado na Tabela 8.3. As mesmas medidas foram utilizadas para avaliar os algoritmos, indicadas pelas colunas 3, 4, 5 e 6 da Tabela 8.3.

Tabela 8.3: Resultados dos algoritmos naive bayes e random forest para o caso 1

Instância	**Método**	**Precisão**	**Recall**	**F-measure**	**CCI**
1mil	NaiveBayes	0,731	0,713	0,708	0,713
	RandomForest	0,7	0,7	0,7	0,704
5mil	NaiveBayes	0,75	0,73	0,72	0,729
	RandomForest	0,698	0,697	0,697	0,697
10mil	NaiveBayes	0,752	0,711	0,699	0,699
	RandomForest	0,703	0,703	0,703	0,703
20mil	NaiveBayes	0,709	0,702	0,704	0,703
	RandomForest	0,684	0,684	0,684	0,684
30mil	NaiveBayes	0,752	0,735	0,73	0,734
	RandomForest	0,698	0,698	0,698	0,697
40mil	NaiveBayes	0,726	0,72	0,718	0,719
	RandomForest	0,692	0,692	0,692	0,691
50mil	NaiveBayes	0,73	0,724	0,722	0,724
	RandomForest	0,696	0,696	0,696	0,695
60mil	NaiveBayes	0,747	0,716	0,707	0,715
	RandomForest	0,692	0,692	0,692	0,692
70mil	NaiveBayes	0,743	0,72	0,713	0,719
	RandomForest	0,696	0,696	0,696	0,696
80mil	NaiveBayes	0,739	0,727	0,723	0,726
	RandomForest	0,695	0,695	0,695	0,695
90mil	NaiveBayes	0,735	0,724	0,721	0,724
	RandomForest	0,694	0,694	0,694	0,694
100mil	NaiveBayes	0,745	0,711	0,7	0,71
	RandomForest	0,695	0,695	0,695	0,695

Os resultados de ambos algoritmos (*Naive Bayes* e *Random Forest*) também se apresentam de forma consistente para os diversos números de instâncias, sem que ocorra uma variação muito grande entre os diferentes tamanhos de amostras.

Os resultados dos algoritmos SVM, Naive Bayes e Random Forest foram todos comparados e os gráficos foram construídos para facilitar a visualização desta comparação. A Figura 8.1 mostra a comparação dos diferentes valores de Precisão, na Figura 8.2 está a comparação dos valores de Sensitividade, os valores de F-measure são comparados na Figura 8.3 e por último estão os valores de CCI na Figura 8.4. O eixo X indica o número de instâncias e o Eixo Y indica o valor obtido da medida utilizada.

Pelos gráficos podemos observar que para a maioria dos tamanhos de amostras o algoritmo SVM apresenta um resultado superior aos demais algoritmos, além de apresentar uma maior consistência para os diferentes tamanhos de amostra.

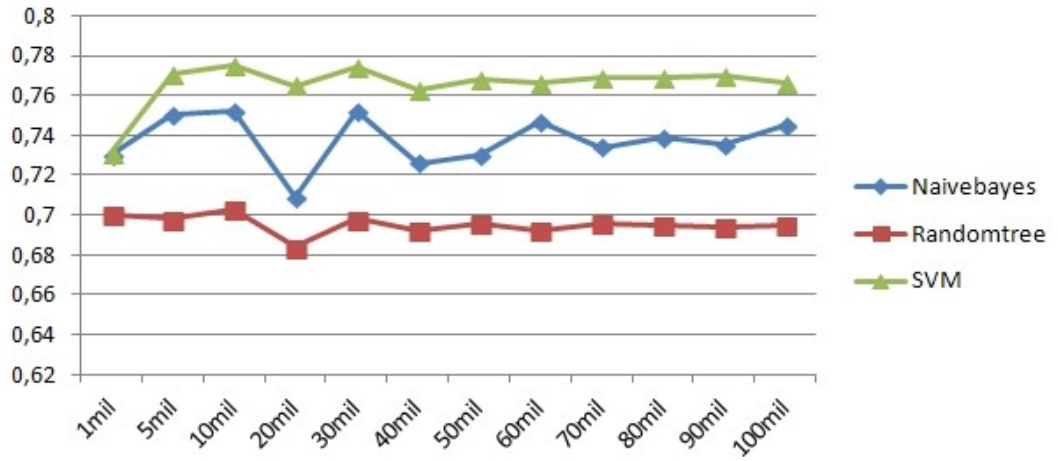

Figura 8.1: Comparação da precisão

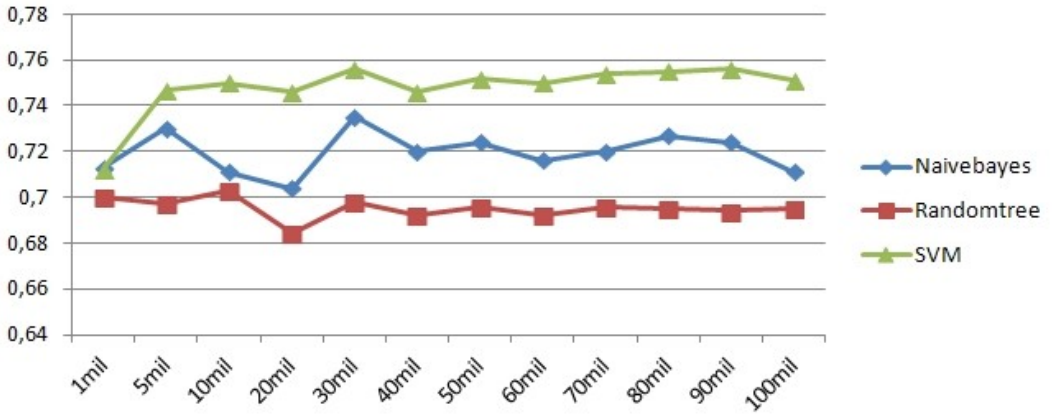

Figura 8.2: Comparação da sensitividade

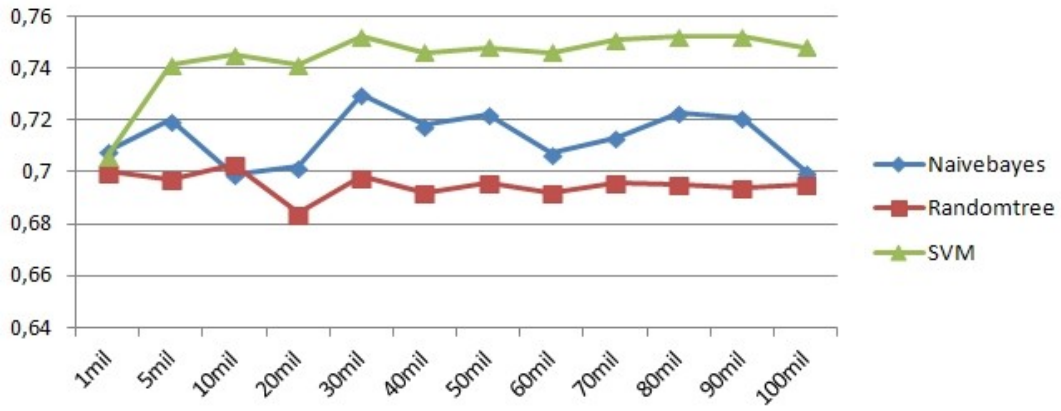

Figura 8.3: Comparação da f-measure

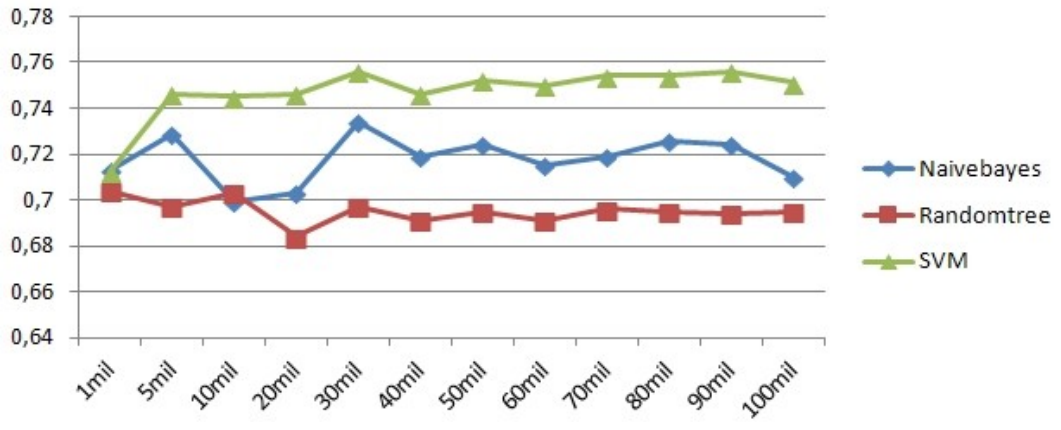

Figura 8.4: Comparação da CCI

Tabela 8.4: Resultado de SVM

Algoritmo	**Precisão**	**Sensitividade**	**F-measure**	**CCI**
Naive Bayes	↑2.71%	↑ 2.8%	↑2.7%	↑ 2.65%
Random Forest	↑ 7.1%	↑ 5.2%	↑ 6.0%	↑ 5.15%

A Tabela 8.4 apresenta que um resumo da taxa de melhoria que o algoritmo SVM apresenta em relação ao Naivebayes e Random Forest na Precisão, Sensitividade, F-measure e CCI.

8.2.3 Análise dos resultados de caso 1

Baseados os resultados do Caso 1, foram realizadas três análises:

1. É possível confirmar que os três algoritmos escolhidos possuem uma exatidão (CCI) significativa, podendo chegar a 68%. Isto indica que os usuários irão aceitar 68 dos 100 itens recomendados. A exatidão destes três algoritmos chega a uma média de 72%, com isto é possível fazer algumas suposições: primeiro, os três algoritmos possuem um efeito superior para recomendação de amigos em Microblogs, segundo, os atributos escolhidos tiveram um papel fundamental na taxa de exatidão.

2. Os quarto critérios de avaliação utilizados (precisão, Sensitividade, F-measure e CCI) se mostraram similares e regulares, nenhum valor anormal foi obtido. Então é possível chegar a conclusão que o experimento é estável e confiável devido à variação das medições realizadas.

3. Os gráficos mostraram testes iniciando de 5 mil instâncias nos três algoritmos, o desempenho das medidas do algoritmo SVM é superior às métricas dos dois algoritmos, e o resultado é uma gradual convergência. Observando os resultados da Naive Bayes e random decision trees, apesar do desempenho da random forest tree ter sido regular, sua exatidão é significativamente menor que os outros dois modelos. O Naive Bayes apresentou um bom desempenho, porém ao atingir uma ordem de 90.000 instâncias, este desempenho caiu repentinamente, indicando que os resultados não convergiram. Não sendo considerado um método adequado para recomendação de amigos nos Microblogs.

8.3 Resultados do Caso 2

Esta seção apresenta os resultados obtidos com a aplicação do algoritmo SVM nos testes do Caso 2, considerando um tamanho de amostra igual a 50 mol. A seção 8.3.1 apresenta os resultados do algoritmo SVM, a seção 8.3.2 exibe os resultados dos algoritmos de comparação e a seção 8.3.3 realiza uma análise dos resultados obtidos. A Tabela 8.5 mostra os parâmetros considerados para os testes realizados no Caso 2.

Tabela 8.5: Parâmetros para o caso2

Conjunto de dados	50 mil instâncias
Número de experimentos	81
Método de treinamento	10-*Fold Cross Validation*

8.3.1 Resultados do algoritmo SVM

A Tabela 8.6 apresenta os resultados do algoritmo SVM para o Caso 2, a primeira coluna apresenta o atributo que foi retirado para comparação, as outras colunas representam os valores de Precisão, Sensitividade e F-measure, respectivamente, calculados sem o atributo que foi retirado.

Tabela 8.6: Resultados do algoritmo SVM para o caso 2

Atributos	Precisão	Sensitividade	F-measure
Nenhum Retirado	0,768	0,752	0,748
Atividade	0,738	0,713	0,725
Idade	0,768	0,753	0,75
Categoria	0,768	0,753	0,749
Aceitação de Seguidos	0,762	0,738	0,732
Aceitação de Seguidores	0,61	0,603	0,596
Gênero	0,767	0,75	0,747
Número de fãs	0,776	0,752	0,748
Similaridade de palavras- chave	0,709	0,703	0,705
Amigo em comum	0.712	0,681	0,696

A Tabela 8.6 mostra como os atributos influenciam na recomendação de amigos, em negrito estão os valores que mais foram afetados com a retirada de um atributo, as maiores quedas de Precisão, Sensitividade e F-measure ocorreram no teste em que o atributo "Aceitação de Seguidores" foi retirado, indicando que este é um fator crucial para que possa ser feita uma recomendação adequada de amigos nas redes sociais.

8.3.2 Comparação de algoritmos

Para verificar se os atributos se comportam da mesma forma para diferentes algoritmos, foram realizados os mesmos testes para os algoritmos Naive Bayes e Random Forest. As Figuras 8.5, 8.6 e 8.7 mostram os gráficos que apresentam as comparações para as medidas de Precisão, Sensitividade e F-measure, respectivamente. O eixo X apresenta o atributo que foi retirado e o eixo Y apresenta o valor da medida considerada.

Em todos os algoritmos aplicados, é possível ver que ocorre uma queda nos resultados em que se retira o atributo "aceitação de seguidores", indicando que este atributo é realmente importante, independente do algoritmo utilizado. As redes sociais costumam basear suas recomendações nos amigos em comum dos usuários, porém, através desse experimento, vemos que outros fatores podem ser ainda mais importantes.

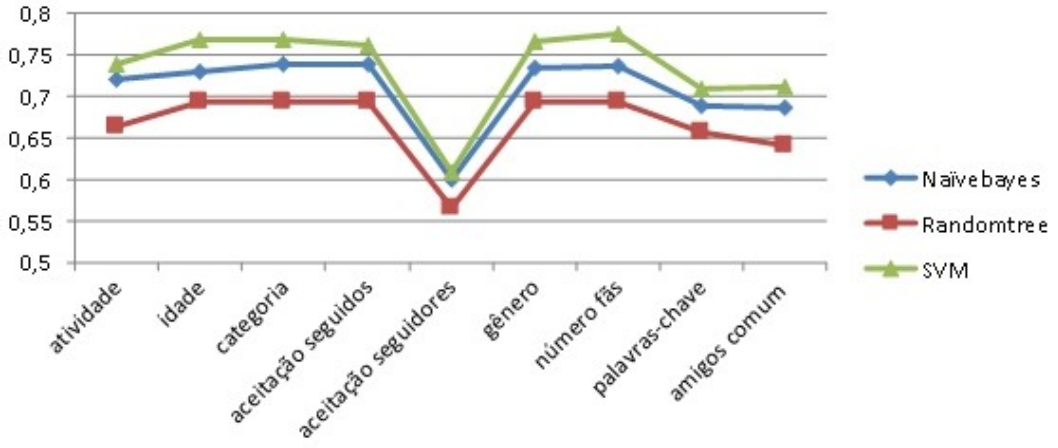

Figura 8.5: Comparação da Precisão

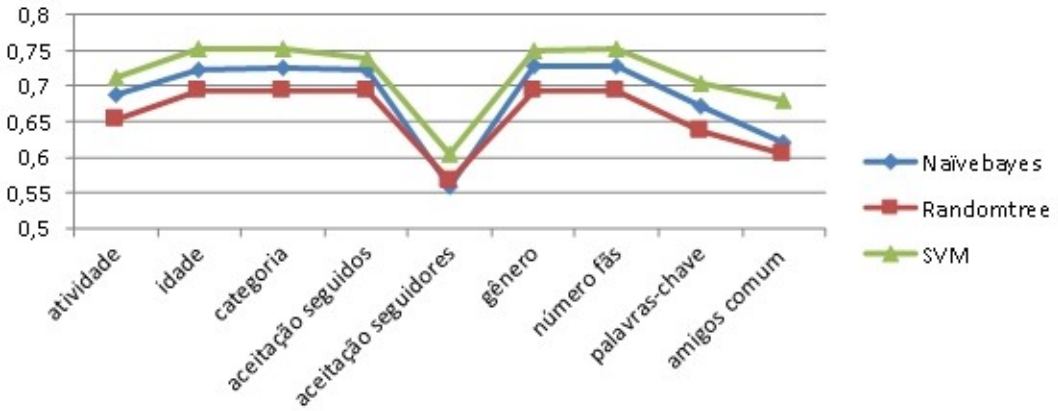

Figura 8.6: Comparação da Sensitividade

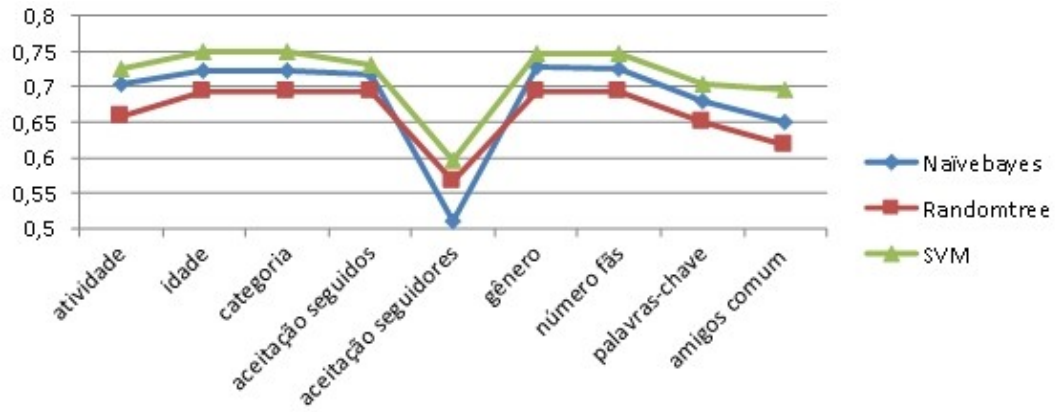

Figura 8.7: Comparação da F-measure

8.3.3 Análise dos resultados de caso2

Para este experimento, o resultado gerou as seguintes análises:

a) O modelo de treinamento é particularmente sensível à aceitação dos seguidores, no caso em que o atributo de aceitação de seguidores foi retirado, a exatidão caiu significativamente. Nos casos em que o atributo retirado foi "similaridade de palavras-chave", "amigos em comum", "atividade" ou "aceitação de seguidos" também houve uma influência no resultado obtido. Outros atributos como: gênero, idade, categoria e número de amigos apresentaram uma influência baixa. A ordem de influência dos atributos, considerando sua exatidão, é a seguinte:

   Aceitação dos seguidores>Similaridade de palavras-chave>Atividade> Aceitação de seguidos>Gênero>Amigos em comum>Categoria>Idade

b) O modelo SVM ainda mantém um desempenho excepcional. Removendo um atributo fundamental, a aceitação de seguidores, desta situação, o modelo ainda obtém uma exatidão de aproximadamente 60%. Isto não ocorre nos outros dois modelos.

8.4 Discussão dos Resultados

Através destes dois casos estudados é possível observar a superioridade da classificação do SVM. Tanto no Caso 1, quanto no Caso 2 ele superou os algoritmos comparados. O Caso 1 foi útil para determinar o comportamento do algoritmo SVM para diferentes tamanhos de amostras, ele provou ser capaz de lidar com diferentes tamanhos de amostras de forma eficiente e com bons resultados. Isso mostra que ele é capaz de atuar em redes sociais de diferentes dimensões. Os algoritmos Naive Bayes e Random Forest também apresentaram um bom desempenho e resultados, porém não foram capazes de superar o algoritmo SVM.

O Caso 2 permitiu que fosse possível compreender como o algoritmo SVM se comporta com diferentes atributos, foi possível confirmar que alguns fatores são mais importantes que outros para a correta recomendação de amigos nas redes sociais. Este fato foi confirmado com a aplicação de outros algoritmos, que também se comportaram de forma similar para os mesmos atributos.

Com isso, é possível confirmar que o algoritmo SVM apresenta bons resultados para diferentes tamanhos de amostras e que é possível potencializar uma melhoria na recomendação de amigos com a seleção apropriada de atributos.

Capítulo 9

Considerações Finais

Um fenômeno fundamental nas redes sociais é a formação as sociedades com amizades. Os membros/usuários fazem amigos através das interações sociais e a troca de informações. A análise da formação de amizades nas redes sociais online pode ajudar a entender diversos problemas sociais e psicológicos, tal como, geração de comunidades, identificação de interesses e formação de opiniões. Pode também facilitar bastante a combinação de usuários ou a recomendação de amigos.

A importância da recomendação de amigos está em dois aspectos, no aspecto macroscópico, fazer esta conexão de pessoas é crítico para o crescimento inicial e desenvolvimento posterior das redes sociais online. No aspecto microscópico, um membro em uma rede social online pode se frustrar tentando encontrar pessoas com as quais possa realizar uma conexão ou até mesmo uma amizade dentro de um grande grupo de usuários. Sugerir usuários relevantes com interesses em comum para cada indivíduo pode melhorar a experiência do usuário.

A grande quantidade de conteúdos gerados pelos usuários nas redes sociais impões vários desafios para minerar os interesses e comportamentos regulares dos usuários para que possam ser feitas recomendações. As informações sobre o comportamento do usuário está, muitas vezes, espalhado pelos links das redes sociais e em conteúdo que reflete os interesses dos usuários, tal como, um perfil auto gerado, tags semânticas, ação do navegador, interação com outros membros e assim em diante.

Na presente dissertação, o algoritmo SVM foi escolhido por suas diversas vantagens para ser utilizado na recomendação de amigos nas redes sociais online. Este trabalho foca no problema de recomendação em duas perspectivas: a primeira foca nos links das redes sociais e a segunda trabalha em cima do conteúdo gerado pelos usuários. Além disso, os fatores cruciais que influenciam a aceitação de recomendações foram explorados nessa pesquisa. As principais contribuições e inovações apresentadas por esta pesquisa são:

- Quantificação dos atributos:Foi desenvolvido o módulo de atributos, ou seja, os atributos para aplicar o algoritmo SVM, os atributos escolhidos provaram ser os fatores de maior influência para a recomendação de amigos. Este módulo desempenha um importante papel na exatidão do resultado;
- Aplicação de SVM para sistema de recomendação: Foram realizados os testes com o algoritmo SVM, aplicando a proposta deste trabalho, e o SVM provou ser um algoritmo melhor e mais marcante que naïve bayes e random forest;

- Implementação do sistema: Este sistema consiste de três partes, processamento de dados, computação os indexes dos atributos, computação do SVM, realização da recomendação com um sistema utilizando SVM, que nunca havia sido aplicado nesta área;
- Aplicação do sistema em dados reais: Neste trabalho foram realizados experimentos utilizando os dados do Microblog Tencent Weibo. O resultado obtido se mostrou mais preciso que as recomendações reais feitas pelo Tencent Weibo.

Os experimentos deste trabalho mostraram que o modelo SVM proposto apresenta um desempenho eficiente e com boa exatidão na recomendação de amigos nas redes sociais. O resultado do SVM é 72% melhor que os métodos usados para comparação, os algoritmos Naïve Bayes e Random Forest. Foram considerados diferentes tamanhos de amostras para testar a eficiência e desempenho destes modelos. O resultado mostrou que o algoritmo SVM é melhor para amostras de diversos tamanhos.

Além disso, também explorou-se os fatores cruciais que possuem maior influência na decisão do usuário. Um experimento foi conduzido, onde um atributo foi excluído da recomendação de cada vez. O resultado mostra que as conexões sociais e os interesses em comum possuem uma influência significativa no momento em que o usuário toma sua decisão.

O sistema de recomendações deste trabalho ainda dá espaço para algumas melhorias:

- A variação do tempo para os usuários: Esta dissertação não leva em consideração as mudanças de interesse que ocorrem com o passar do tempo;
- Aglomeração populacional: Neste trabalho não foi considerado o interesse entre diferentes grupos. Os atributos de idade e gênero não influenciaram no resultado, no futuro seria interessante criar outro módulo para considerar estas características;
- Implementação Online: O sistema, em sua forma atual, só suporta cálculos de forma off-line, ele não pode dar feedback em tempo real.

Podemos levar em consideração nos trabalhos futuros, para complementar o modelo desenvolvido pela atual pesquisa.

Referências

[1] Kddcup. `http://www.kddcup2012.org/`. vii, 37

[2] Netflix. `https://www.netflix.com/?locale=pt-BR/`. 1

[3] Amazon. `http://www.amazon.com/`. 1

[4] Delicious. `https://delicious.com/`. 1

[5] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):734–749, 2005. 1

[6] Gediminas Adomavicius and Alexander Tuzhilin. Expert-driven validation of rule-based user models in personalization applications. *Data Mining and Knowledge Discovery*, 5(1-2):33–58, 2001. 1

[7] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186. ACM, 1994. 1, 20

[8] ebay. `http://www.ebay.com/`. 1

[9] youtube. `http://www.youtube.com/`. 1, 2

[10] facebook. `http://www.facebook.com/`. 2

[11] twitter. `http://www.twitter.com/`. 2

[12] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995. 4, 6

[13] Sung-Hwan Min and Ingoo Han. Recommender systems using support vector machines. In *Web Engineering*, pages 387–393. Springer, 2005. 4

[14] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992. 6

[15] Tariq Mahmood and Francesco Ricci. Improving recommender systems with adaptive conversational strategies. In *Proceedings of the 20th ACM conference on Hypertext and hypermedia*, pages 73–82. ACM, 2009. 20

[16] Francesco Ricci, Lior Rokach, and Bracha Shapira. *Introduction to recommender systems handbook*. Springer, 2011. 20

[17] Paul Resnick and Hal R Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997. 20

[18] Dietmar Jannach, Markus Zanker, Markus Jessenitschnig, and Oskar Seidler. Developing a conversational travel advisor with advisor suite. *Information and Communication Technologies in Tourism 2007*, pages 43–52, 2007. 20

[19] Greg Linden, Brent Smith, and Jeremy York. Amazon. com recommendations: Item-to-item collaborative filtering. *Internet Computing, IEEE*, 7(1):76–80, 2003. 21

[20] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 154–161. ACM, 2005. 21

[21] Ioannis Konstas, Vassilios Stathopoulos, and Joemon M Jose. On social networks and collaborative recommendation. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 195–202. ACM, 2009. 21

[22] Mao Ye, Peifeng Yin, and Wang-Chien Lee. Location recommendation for location-based social networks. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 458–461. ACM, 2010. 21

[23] Weigang Li, Jianya Zheng, and Yang Liu. Retweeting prediction using relationship committed adjacency matrix. *Brazilian Workshop on Social Network Analysis and Mining-BraNAM2013*, 2013:1561–1572, 2013. 21

[24] Rong Jin, Joyce Y Chai, and Luo Si. An automatic weighting scheme for collaborative filtering. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 337–344. ACM, 2004. 21

[25] Jun Wang, Arjen P De Vries, and Marcel JT Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 501–508. ACM, 2006. 21

[26] Jialie Shen, Bin Cui, John Shepherd, and Kian-Lee Tan. Towards efficient automated singer identification in large music databases. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 59–66. ACM, 2006. 22, 24

[27] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008. 24

[28] Bernard Widrow and Samuel D Stearns. Adaptive signal processing. *Englewood Cliffs, NJ, Prentice-Hall, Inc., 1985, 491 p.*, 1, 1985. 24

[29] Thorsten Joachims. *Text categorization with support vector machines: Learning with many relevant features*. Springer, 1998. 24

[30] David D Lewis, Robert E Schapire, James P Callan, and Ron Papka. Training algorithms for linear text classifiers. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 298–306. ACM, 1996. 24

[31] Yiming Yang and Xin Liu. A re-examination of text categorization methods. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 42–49. ACM, 1999. 24

[32] Robin Burke. Knowledge-based recommender systems. *Encyclopedia of library and information systems*, 69(Supplement 32):175–186, 2000. 25, 26

[33] Joseph A Konstan, Bradley N Miller, David Maltz, Jonathan L Herlocker, Lee R Gordon, and John Riedl. Grouplens: applying collaborative filtering to usenet news. *Communications of the ACM*, 40(3):77–87, 1997. 25

[34] Michael J Pazzani. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, 13(5-6):393–408, 1999. 25

[35] Alexander Felfernig and Robin Burke. Constraint-based recommender systems: technologies and research issues. In *Proceedings of the 10th international conference on Electronic commerce*, page 3. ACM, 2008. 25

[36] Alexander Felfernig, Gerhard Friedrich, Dietmar Jannach, and Markus Zanker. An integrated environment for the development of knowledge-based recommender applications. *International Journal of Electronic Commerce*, 11(2):11–34, 2006. 25

[37] Markus Zanker, Markus Jessenitschnig, and Wolfgang Schmid. Preference reasoning with soft constraints in constraint-based recommender systems. *Constraints*, 15(4):574–595, 2010. 25

[38] Alexander Felfernig, Erich Teppan, and Bartosz Gula. Knowledge-based recommender technologies for marketing and sales. *International Journal of Pattern Recognition and Artificial Intelligence*, 21(02):333–354, 2007. 26, 27

[39] Robin Burke. Interactive critiquing forcatalog navigation in e-commerce. *Artificial Intelligence Review*, 18(3-4):245–267, 2002. 26

[40] Robin D Burke, Kristian J Hammond, and BC Yound. The findme approach to assisted browsing. *IEEE Expert*, 12(4):32–40, 1997. 26

[41] James Reilly, Jiyong Zhang, Lorraine McGinty, Pearl Pu, and Barry Smyth. A comparison of two compound critiquing systems. In *Proceedings of the 12th international conference on Intelligent user interfaces*, pages 317–320. ACM, 2007. 26

[42] Alexander Felfernig, Klaus Isak, Kalman Szabo, and Peter Zachar. The vita financial services sales support environment. In *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*, volume 22, page 1692. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2007. 27

[43] Shari Trewin. Knowledge-based recommender systems. *Encyclopedia of Library and Information Science: Volume 69-Supplement 32*, page 180, 2000. 27

[44] Francesco Ricci and Quang Nhat Nguyen. Acquiring and revising preferences in a critique-based mobile recommender system. *Intelligent Systems, IEEE*, 22(3):22–29, 2007. 27

[45] Fabiana Lorenzi and Francesco Ricci. Case-based recommender systems: a unifying view. In *Intelligent Techniques for Web Personalization*, pages 89–113. Springer, 2005. 27

[46] Zhenhui Jiang, Weiquan Wang, and Izak Benbasat. Multimedia-based interactive advising technology for online consumer decision support. *Communications of the ACM*, 48(9):92–98, 2005. 27

[47] Robin Burke. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4):331–370, 2002. 28

[48] Jilin Chen, Werner Geyer, Casey Dugan, Michael Muller, and Ido Guy. Make new friends, but keep the old: recommending people on social networking sites. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 201–210. ACM, 2009. 29

[49] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 287–296. ACM, 2011. 29

[50] Przemyslaw Kazienko, Katarzyna Musial, and Tomasz Kajdanowicz. Multidimensional social network in the social recommender system. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 41(4):746–759, 2011. 30

[51] Jie Bao, Yu Zheng, and Mohamed F Mokbel. Location-based and preference-aware recommendation using sparse geo-social networking data. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pages 199–208. ACM, 2012. 30

[52] Xiwang Yang, Harald Steck, and Yong Liu. Circle-based recommendation in online social networks. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1267–1275. ACM, 2012. 31

[53] I-Hsing Ting, Pei-Shan Chang, Shyue-Liang Wang, et al. Understanding microblog users for social recommendation based on social networks analysis. *J. UCS*, 18(4):554–576, 2012. 31

[54] Mathias M Adankon and Mohamed Cheriet. Model selection for the ls-svm. application to handwriting recognition. *Pattern Recognition*, 42(12):3264–3270, 2009. 32

[55] Yu-Dong Cai, Pong-Wong Ricardo, Chih-Hung Jen, and Kuo-Chen Chou. Application of svm to predict membrane protein types. *Journal of Theoretical Biology*, 226(4):373–376, 2004. 32

[56] Salman Jamali and Huzefa Rangwala. Digging digg: Comment mining, popularity prediction, and social network analysis. In *Web Information Systems and Mining, 2009. WISM 2009. International Conference on*, pages 32–38. IEEE, 2009. 32

[57] Wei Pan, Nadav Aharony, and Alex Pentland. Composite social network for predicting mobile apps installation. In *AAAI*, 2011. 32

[58] Zhi Yang, Christo Wilson, Xiao Wang, Tingting Gao, Ben Y Zhao, and Yafei Dai. Uncovering social network sybils in the wild. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 259–268. ACM, 2011. 33

[59] Chi-Yao Tseng and Ming-Syan Chen. Incremental svm model for spam detection on dynamic email social networks. In *Computational Science and Engineering, 2009. CSE'09. International Conference on*, volume 4, pages 128–135. IEEE, 2009. 33

[60] Benjamin Markines, Ciro Cattuto, and Filippo Menczer. Social spam detection. In *Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web*, pages 41–48. ACM, 2009. 33

[61] Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology*, 61(12):2544–2558, 2010. 33

[62] Haifeng Liu, Ee-Peng Lim, Hady W Lauw, Minh-Tam Le, Aixin Sun, Jaideep Srivastava, and Young Kim. Predicting trusts among users of online communities: an epinions case study. In *Proceedings of the 9th ACM conference on Electronic commerce*, pages 310–319. ACM, 2008. 34

[63] Gae-won You, Seung-won Hwang, Zaiqing Nie, and Ji-Rong Wen. Socialsearch: enhancing entity search with social network matching. In *Proceedings of the 14th International Conference on Extending Database Technology*, pages 515–519. ACM, 2011. 34

[64] Jing Li, Xiukun Wang, Kai Sun, and Jiankang Ren. Recommendation algorithm with support vector regression based on user characteristics. In *Proceedings of the 9th International Symposium on Linear Drives for Industry Applications, Volume 3*, pages 455–462. Springer, 2014. 34

[65] Chinese microblogger character analysis using svm. 35

[66] Mark EJ Newman. Power laws, pareto distributions and zipf's law. *Contemporary physics*, 46(5):323–351, 2005. 35

[67] Zan Huang, Hsinchun Chen, and Daniel Zeng. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Transactions on Information Systems (TOIS)*, 22(1):116–142, 2004. 35

[68] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 253–260. ACM, 2002. 36

[69] Xuan Nhat Lam, Thuc Vu, Trong Duc Le, and Anh Duc Duong. Addressing cold-start problem in recommendation systems. In *Proceedings of the 2nd international conference on Ubiquitous information management and communication*, pages 208–211. ACM, 2008. 36

[70] ebay. `http://www.baifendian.com/`. 36

[71] Tencent weibo. `http://t.qq.com/`. 37

[72] Xin Li, Lei Guo, and Yihong Eric Zhao. Tag-based social interest discovery. In *Proceedings of the 17th international conference on World Wide Web*, pages 675–684. ACM, 2008. 44

[73] Edans FO De Sandes, Li Weigang, and Alba Cristina MA de Melo. Logical model of relationship for online social networks and performance optimizing of queries. In *Web Information Systems Engineering-WISE 2012*, pages 726–736. Springer, 2012. 45

[74] Orit Shechtman. The coefficient of variation as an index of measurement reliability. In *Methods of Clinical Epidemiology*, pages 39–49. Springer, 2013. 47

[75] Xiao-Feng Hui and Jie Sun. An application of support vector machine to companies' financial distress prediction. In *Modeling decisions for artificial intelligence*, pages 274–282. Springer, 2006. 51

[76] Vladimir Vapnik. *The nature of statistical learning theory*. springer, 2000. 51

[77] Nai-yang Deng and Ying-jie Tian. A new method of data mining: support vector machines. *Science Publication, Beijing City*, 2004. 51

[78] LK Luo, H Peng, QS Zhang, and CD Lin. A comparison of strategies for unbalance sample distribution in support vector machine. In *Industrial Electronics and Applications, 2006 1ST IEEE Conference on*, pages 1–5. IEEE, 2006. 51, 52

[79] Jiawei Han, Micheline Kamber, and Jian Pei. *Data mining: concepts and techniques*. Morgan kaufmann, 2006. 52

[80] Tatjana Eitrich and Bruno Lang. On the optimal working set size in serial and parallel support vector machine learning with the decomposition algorithm. In *Proceedings of the fifth Australasian conference on Data mining and analystics-Volume 61*, pages 121–128. Australian Computer Society, Inc., 2006. 54

[81] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al. A practical guide to support vector classification, 2003. 54