UNIVERSIDADE FEDERAL DE OURO PRETO

Detectando Avaliações Spam em uma Rede Social Baseada em Localização

Helen de Cássia Sousa da Costa Lima
Universidade Federal de Ouro Preto

Orientador: Fabrício Benevenuto

Coorientador: Luiz Henrique de Campos Merschmann

Dissertação submetida ao Instituto de Ciências Exatas e Biológicas da Universidade Federal de Ouro Preto para obtenção do título de Mestre em Ciência da Computação

Ouro Preto, Julho de 2013

Detectando Avaliações Spam em uma Rede Social Baseada em Localização

Helen de Cássia Sousa da Costa Lima
Universidade Federal de Ouro Preto

Orientador: Fabrício Benevenuto

Coorientador: Luiz Henrique de Campos Merschmann

L732d Lima, Helen de Cássia Sousa da Costa.
Detectando avaliações spam em uma rede social baseada em localização [manuscrito] / Helen de Cássia Sousa da Costa Lima – 2013.
67f.: il. color.; graf.; tab.

Orientador: Prof. Dr. Fabrício Benevenuto de Souza.
Orientador: Prof. Dr. Luiz Henrique de Campos Merschmann.

Dissertação (Mestrado) - Universidade Federal de Ouro Preto. Instituto de Ciências Exatas e Biológicas. Departamento de Computação. Programa de Pós-graduação em Ciência da Computação.
Área de concentração: Ciência da Computação.

1. Redes sociais - Teses. 2. Spam (Mensagens eletrônicas) - Teses. 3. Ciências sociais – Análises de redes - Teses. I. Souza, Fabrício Benevenuto de. II. Merschmann, Luiz Henrique de Campos. III. Universidade Federal de Ouro Preto. IV. Título.

CDU: 519.876.3

Catalogação: sisbin@sisbin.ufop.br

CDU: 669.162.16

MINISTÉRIO DA EDUCAÇÃO E DO DESPORTO
Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Biológicas – ICEB
Programa de Pós-Graduação em Ciência da Computação

Ata da Defesa Pública de Dissertação de Mestrado

Aos 19 dias do mês de julho de 2013, às 10 horas e 30 minutos na Sala de Seminários do DECOM no Instituto de Ciências Exatas e Biológicas (ICEB), reuniram-se os membros da banca examinadora composta pelos professores: **Prof. Dr. Fabrício Benevenuto de Souza (presidente e orientador), Prof. Dr. Luiz Henrique de Campos Merschmann (co-orientador), Profa. Dra. Mirella Moura Moro e Dr. Fabrício Jailson Barth**, aprovada pelo Colegiado do Programa de Pós-Graduação em Ciência da Computação, a fim de arguirem a mestranda **Helen de Cássia Souza da Costa**, com o título **"Detectando Avaliações Spam em uma Rede Social Baseada em Localização"**. Aberta a sessão pelo presidente, coube à candidata, na forma regimental, expor o tema de sua dissertação, dentro do tempo regulamentar, sendo em seguida questionado pelos membros da banca examinadora, tendo dado as explicações que foram necessárias.

Recomendações da Banca:

(X) Aprovada sem recomendações

( ) Reprovada

( ) Aprovada com recomendações: _____

Banca Examinadora:

Prof. Dr. Fabrício Benevenuto de Souza

Prof. Dr. Luiz Henrique de Campos Merschmann

Profa. Dra. Mirella Moura Moro

Dr. Fabrício Jailson Barth

Prof. Dr. Haroldo Gambini Santos
Coordenador do Programa de Pós-Graduação em Ciência da Computação
DECOM/ICEB/UFOP

Ouro Preto, 19 de julho de 2013.

Dedico este trabalho a todos os professores do Programa de Pós-Graduação em Ciência da Computação da UFOP. A garra e determinação de vocês me inspira e motiva a fazer sempre o meu melhor.

Detecção de Avaliações Spam em uma Rede Social Baseada em Localização

Resumo

Redes sociais baseadas em localização (*Location-based Social Networks* - LBSNs) são um novo tipo de sistema da Web 2.0 que vem atraindo cada vez mais novos usuários. Redes como Foursquare e Yelp permitem que o usuário compartilhe a sua localização geográfica com sua rede social através de *smartphones* que possuem GPS, busquem por locais interessantes e também postem avaliações em locais existentes. Ao permitir que os usuários comentem sobre os locais, LBSNs cada vez mais têm que lidar com diferentes formas de ataques, que visam a propaganda de mensagens não solicitadas nas avaliações sobre os locais. *Spammers* podem prejudicar a confiança dos usuários no sistema, comprometendo assim o seu sucesso em promover interações sociais baseadas em localização.

Neste trabalho, investigamos a tarefa de identificar diferentes tipos de spam em avaliações de uma popular LBSN brasileira, chamada Apontador. Com base em uma coleção de avaliações pré-classificada fornecida pelo Apontador e em informações coletadas sobre usuários e locais, identificamos três tipos de avaliações irregulares que denominamos como *Comercial local*, *Boca-suja* e *Poluidora*. Em seguida, utilizamos o nosso estudo de caracterização em uma abordagem de classificação que foi capaz de diferenciá-las com alta precisão. Particularmente, a nossa abordagem de classificação plana foi capaz de detectar corretamente 77% das avaliações comerciais locais, 64% das poluidoras, 50% das bocas-sujas, classificando erroneamente apenas cerca de 5% das avaliações não-spam. Além disso, nossos resultados experimentais mostraram que, mesmo com um pequeno subconjunto de atributos (contendo 10 atributos), a nossa abordagem de classificação

foi capaz de atingir uma acurácia alta (75%). Mesmo quando usamos apenas um dos tipos de atributos, como por exemplo atributos de conteúdo, nossa classificação produz benefícios significativos, com acurácia de aproximadamente 68%.

Detecting Tip Spam in Location-based Social Networks

Abstract

Location Based Social Networks (LBSNs) are recent Web 2.0 systems that are attracting new users in exponential rates. LBSNs like Foursquare and Yelp allow users to share their geographic location with friends through smartphones equipped with GPS, search for interesting places as well as post tips about existing locations. By allowing users to comment on locations, LBSNs increasingly have to deal with new forms of spammers, which aim at advertising unsolicited messages on tips about locations. Spammers may jeopardize the trust of users on the system, thus, compromising its success in promoting location-based social interactions.

In this work, we investigated the task of identifying different types of tip spam on a popular Brazilian LBSN system, namely Apontador. Based on a labeled collection of tips provided by Apontador as well as crawled information about users and locations, we identified three types of irregular tips, namely Local Marketing, Pollution and, Bad-mouthing. We leveraged our characterization study towards a classification approach able to differentiate these tips with high accuracy. Particularly, our flat classification approach was able to detect correctly 77% of the local marketing tips, 64% of pollution tips, 50% of bad-mouthing tips, making few mistakes (about 5% of non-spam tips). Additionally, our experimental results show that even with a small subset of attributes (containing 10 attributes), our classification approach was able to reach high accuracy (75%). Our classification could also produce significant benefits even when using only one set of attributes, being the best performance for content attributes, with an accuracy of almost 68%.

Declaração

Esta dissertação é resultado de meu próprio trabalho, exceto onde referência explícita é feita ao trabalho de outros, e não foi submetida para outra qualificação nesta nem em outra universidade.

Helen de Cássia Sousa da Costa Lima

Agradecimentos

Em primeiro lugar agradeço a Deus, que me capacitou para a realização deste trabalho.

Aos meus pais, Paulo e Edna, pelo amor, dedicação e por não medirem esforços pela minha educação. Tudo que fizeram por mim foi fundamental para que eu chegasse até aqui.

Ao meu marido, Marcos, por seu amor, companheirismo e renúncias. Se não fosse pela sua insistência, eu não teria passado por essa experiência tão enriquecedora que foi o meu mestrado.

Ao meu orientador, Fabrício Benevenuto, por compartilhar seu conhecimento, pelas oportunidades e principalmente, por ser tão motivador.

Ao meu coorientador, Luiz Merschmann, pela disposição em nos ajudar quanto à realização deste trabalho, sua parceria foi fundamental.

Ao Fabrício Barth, por suas valiosas contribuições para o trabalho.

Aos meus irmãos em Cristo, tanto da igreja quanto da minha família, pelo carinho e orações.

Aos amigos que fiz durante o mestrado, pelo companheirismo, boas conversas e pelas dificuldades compartilhadas durante as disciplinas.

À CAPES e à UFOP, pelo apoio financeiro durante o mestrado.

E ao Apontador, pelos dados fornecidos, que tornaram possível a realização desse trabalho.

Sumário

Lista de Figuras

Lista de Tabelas

"Eu tive muitas coisas que guardei em minhas mãos, e as perdi. Mas tudo o que guardei nas mãos de Deus, eu ainda possuo."

— Martin Luther King

Nomenclatura

API	*Application Programming Interface*
CDF	*Cumulative Distribution Function*
GPS	*Global Positioning System*
LBSN	*Location-based Social Network*
MDA	*Mean Decrease Accuracy*
RBF	*Radial Basis Function*
RF	*Random Forest*
SAC	*Serviço de Atendimento ao Consumidor*
SVM	*Support Vector Machine*
URL	*Uniform Resource Locator*

Capítulo 1

Introdução

Redes sociais baseadas em localização (*Location-based Social Networks* - LBSNs) são um novo tipo de sistema da Web 2.0 que vem atraindo cada vez mais novos usuários. Aproximadamente uma em cada cinco pessoas que possuem um *smartphone* acessa esse tipo de serviço através do seu dispositivo móvel [8]. Redes como Foursquare e Yelp permitem que o usuário compartilhe a sua localização geográfica com sua rede social através de *smartphones* que possuem GPS.

No Brasil, uma LBSN popular é o Apontador[1], um sistema que possui as principais características de redes como Foursquare e Yelp. Permite que usuários busquem por locais, cadastrem locais e façam *check-in* em locais usando um *smartphone*. Adicionalmente, o Apontador contém uma das funcionalidades mais interessantes de LBSN, a de permitir que os usuários postem avaliações em locais existentes. Através dessas avaliações (dicas) e de um *smartphone* com acesso a uma LBSN, um usuário pode não só encontrar locais próximos para visitar, como também ler sugestões sobre o que pedir, o que comprar e até mesmo o que evitar em um local específico. Portanto, avaliações funcionam como um repositório online de recomendações sobre locais específicos.

1.1 Problemas e Objetivos

Embora atraente como um mecanismo para enriquecer a experiência do usuário no sistema, avaliações abrem oportunidade para usuários disseminarem mensagens não solicitadas. LBSNs cada vez mais têm que lidar com diferentes formas de ataques, que visam a

[1] www.apontador.com.br

propaganda de mensagens não solicitadas em vez de legítimas avaliações sobre os locais. A maioria dos estudos nesta área são focados em opinião spam (*Opinion Spam*). Uma opinião spam é uma opinião não confiável que deliberadamente engana os leitores, dando avaliações positivas a um local a fim de promovê-lo e/ou dando avaliações negativas a fim de prejudicar a reputação do mesmo.

No entanto, pouco se sabe sobre as atividades de outros tipos de ataques. Esses ataques podem ser avaliações que não estão relacionadas com o local, mas sim com anúncios, pornografias e conteúdos irrelevantes que não são relacionados ao local. Esse tipo de comportamento também pode prejudicar a confiança dos usuários no sistema, comprometendo assim o seu sucesso em promover interações sociais baseadas em localização. Avaliações spam também podem comprometer a paciência e satisfação do usuário com o sistema, pois ele precisa separar spam do que vale a pena ler. Além disso, a literatura disponível é muito limitada em fornecer um entendimento profundo do presente problema.

Este trabalho tem como objetivo identificar diferentes tipos de avaliações spam em LBSNs, seguindo uma abordagem baseada em três etapas. A primeira etapa é categorizar avaliações spam em três diferentes classes considerando uma base de dados cedida pelo Apontador, que contém avaliações spam pré-classificadas manualmente. Em seguida, analisar vários atributos extraídos do conteúdo das avaliações e do comportamento dos usuários no sistema, com o intuito de entender o potencial desses atributos para distinguir entre diferentes classes de avaliações spam. E por fim, investigar a viabilidade da aplicação de um método de aprendizado de máquina supervisionado para identificar essas classes de avaliação spam.

1.2 Contribuições do Trabalho

As principais contribuições deste trabalho são:

- Identificação de diferentes tipos de avaliações spam, caracterizando três tipos de avaliações irregulares no Apontador, classificadas como: *(i) Comercial Local*, avaliações contendo propagandas sobre o local alvo ou sobre serviços relacionados ao local; *(ii) Poluidora*, avaliações com conteúdo irrelevante ou não relacionado com o local ou mesmo perguntas, fazendo com que a área reservada para avaliações se torne um SAC (Serviço de Atendimento ao Consumidor); e *(iii) Boca-suja*, avali-

ações caracterizadas por conter comentários agressivos sobre o local, seu dono ou outros usuários, geralmente contendo palavras ofensivas e de baixo calão.

- Detecção automática de diferentes tipos de avaliação spam. Nossa abordagem não foi somente capaz de identificar corretamente uma parte significativa de avaliações spam e não-spam, mas também foi capaz de diferenciar avaliações spam em três diferentes classes.

1.3 Publicações

Resultados parciais desta pesquisa foram publicados e apresentados em evento científico através da seguinte publicação:

- COSTA, H.; BENEVENUTO, F; MERSCHMANN, L. H. C.: *Detecting tip spam in location-based social networks*. In: *Proceedings of the Annual ACM Symposium on Applied Computing (SAC)*, 2013, Coimbra, Portugal.

Especificamente, abordamos o problema da detecção de spam em avaliações, criando uma pequena coleção de teste composto por avaliações spam e não-spam, e aplicamos uma estratégia de classificação binária para detectar avaliações spam [9].

Um segundo trabalho foi submetido para evento científico e aguarda resposta dos revisores:

- COSTA, H.; BENEVENUTO, F; MERSCHMANN, L. H. C; BARTH, F.: *Pollution, Bad-mouthing, and Local Marketing: The Underground of Location-based Social Networks*. In: *Proceedings of the Annual ACM Conference on Online Social Networks (COSN'13)*, 2013, Boston, USA.

Sendo uma versão muito mais sólida e completa, este segundo trabalho investiga a viabilidade de detectar spam em avaliações de LBSN considerando uma coleção de teste muito maior, um conjunto mais rico de atributos, bem como diferentes classes de comportamentos maliciosos e oportunistas.

1.4 Organização dos Capítulos

O restante do trabalho está organizado da seguinte forma. O **Capítulo 2** aborda os esforços dos trabalhos relacionados. Em seguida, o **Capítulo 3** descreve nossa estratégia para categorizar as avaliações spam em diferentes classes. No **Capítulo 4**, investigamos vários atributos e suas habilidades para distinguir entre os diferentes tipos de classes. Depois, no **Capítulo 5**, é descrita e avaliada nossa estratégia para detectar os três tipos de classes de spam. Finalmente, no **Capítulo 6** concluímos o trabalho e discutimos direções para trabalhos futuros.

Capítulo 2

Trabalhos Relacionados

Neste capítulo apresentamos trabalhos que estão relacionados ao nosso em três aspectos diferentes: no uso de classificação, na detecção de spam em redes sociais, na caracterização de LBSNs ou na detecção de spam em avaliações.

2.1 Classificação

A classificação é o processo de encontrar um modelo (ou função) que descreva e distingua classes de dados ou conceitos, a fim de usar este modelo para predizer a classe de objetos cujo rótulo de classe é desconhecido [12]. A geração do modelo é baseada na análise de um conjunto de dados de treino cujo rótulo de classe é conhecido. Tal modelo pode ser representado de várias formas, como regras de classificação, árvores de decisão, fórmulas matemáticas, redes neurais, etc.

Quando a classe dos dados de treino é fornecida, a classificação também é conhecida como aprendizagem supervisionada. Por outro lado, também existem algoritmos de aprendizagem não-supervisionados (ou algoritmos de agrupamento), em que o rótulo de classe não é conhecido e o número de classes que devem ser aprendidas também pode não ser conhecido com antecedência. Sendo assim, algoritmos de agrupamento podem ser usados para gerar classes de objetos. Nesses algoritmos, os objetos são agrupados de acordo com o princípio da maximização da similaridade intra-classe e da minimização da similaridade inter-classe. Isto é, *clusters* (grupos) de objetos são formados de modo que objetos de um mesmo *cluster* têm alta similaridade quando comparados entre si e são muito dissimilares quando comparados com objetos de outros *clusters*. Cada *cluster*

formado pode ser visto como uma classe de objetos, a partir da qual regras podem ser geradas. Algoritmos de grupamento também pode ser usados para facilitar a formação de hierarquia de classes que agrupe eventos semelhantes.

Métodos não-supervisionados podem não ser a melhor opção para tarefas de classificação, pois, por não terem nenhuma orientação sobre a informação da classe, podem gerar *clusters* que não são desejáveis. Por outro lado, métodos supervisionados exigem uma quantidade significativa de dados rotulados, cuja geração é frequentemente cara e demorada. Sendo assim, uma alternativa para resolver esses problemas é utilizar algoritmos de aprendizagem semi-supervisionados, que aprendem a partir de objetos de dados rotulados e não rotulados. Uma das técnicas de aprendizagem semi-supervisionada é o agrupamento baseado em restrições, que depende de rótulos fornecidos pelos usuários ou de restrições que orientem o algoritmo a fazer um particionamento mais apropriado dos dados. Isso inclui modificar a função objetivo com base em restrições, ou inicializar e restringir o processo de agrupamento com base em exemplos de objetos rotulados.

Conteúdo spam tem sido observado em várias aplicações, como e-mail, ferramentas de busca na Web e blogs. Com isso, várias técnicas de detecção e combate de spam têm sido propostas e muitas delas utilizam métodos de classificação para detectar spam. Wu *et al.* [24] analisaram uma grande base de dados de e-mails spam contendo imagens e identificaram uma série de características visuais úteis para identificar esse tipo de conteúdo spam. Em seguida, propuseram um novo filtro anti-spam utilizando uma abordagem de classificação supervisionada. Semelhantemente, Lin *et al.* [17] abordaram o problema da detecção de spam em blog (*splog*). *Splogs* são blogs indesejáveis, destinados para atrair o tráfego de ferramentas de busca, utilizados exclusivamente para promover *sites* associados. Baseados em atributos extraídos de regularidades temporais e estruturais de conteúdo, tempo de postagem e *links*, propuseram uma abordagem de classificação supervisionada para detecção desse tipo de spam. Adicionalmente, Castillo *et al.* [7] apresentaram um sistema de detecção de spam que usa a topologia do grafo da Web, explorando as dependências de *links* entre as páginas da Web, bem como o conteúdo das próprias páginas. Baseados nas características extraídas dos conteúdos das páginas, utilizaram aprendizagem supervisionada para criar um classificador base. Em seguida, eles utilizaram técnicas de agrupamento na topologia do grafo de propagação de *links* para identificar novas classes. E por fim, utilizaram essas classes como novos atributos para retreinar o classificador base que construíram.

Em nosso trabalho utilizamos aprendizagem supervisionada para criação de um classificador e os algoritmos adotados em nossos experimentos foram o SVM (*Support Vector*

Machine) [21] e RF (*Random Forest*) [5], descritos na seção 5.2.

2.2 Detecção de Spam em Outras Redes Sociais

Conteúdo spam também tem sido observado em sistemas de redes sociais online. Benevenuto *et al.* [4] abordaram a questão da detecção de *spammers* e promovedores de vídeo. Para fazer isso, coletaram uma grande quantidade de dados de usuários do YouTube, com mais de 260 mil usuários. Baseados em uma base de dados de usuários rotulada e em um conjunto de atributos de comportamento do usuário, eles aplicaram uma abordagem de classificação utilizando aprendizagem de máquina para diferenciar usuários legítimos, *spammers* e promovedores.

Ghosh *et al.* [11], investigaram a atividade de *link farming* no Twitter, caracterizada por usuários, especialmente *spammers*, que tentam adquirir um grande número de seguidores na rede social. *Link farming* não só aumenta a popularidade, mas também contribui para a influência perceptível de um usuário, refletida por exemplo, na melhoria do ranqueamento dos seus *tweets* pelas ferramentas de busca. Através da análise de *links* adquiridos por mais de 40.000 contas de *spammers* suspensas do Twitter, eles descobriram que um pequeno grupo de usuários legítimos, populares e altamente ativos, como por exemplo, Barack Obama, fazem parte da maioria das atividades de *link farming*. Esses usuários acabam praticando involuntariamente *link farming* quando tentam acumular capital social ao seguir de volta indiscriminadamente usuários que os seguem. Então, *spammers* acabam explorando esse tipo de comportamento para ganhar seguidores e reputação na rede. Para desencorajar os capitalistas sociais de seguir usuários desconhecidos, os autores propuseram um esquema de ranqueamento que penaliza usuários quando eles seguem *spammers*.

Adicionalmente, Benevenuto *et al.* [3] abordaram o problema de detectar *spammers* no Twitter. Usando uma base de dados de usuários rotulada manualmente, eles aplicaram uma abordagem de classificação utilizando aprendizagem de máquina para diferenciar usuários legítimos de *spammers*. Da mesma forma, Lee *et al.* [16] criaram *honeypots* sociais para identificar um conjunto de *spammers* no MySpace e no Twitter. Eles definem *honeypots* sociais como ferramentas de sistemas de informação que monitoram o comportamento de *spammers* e registrar suas informações (por exemplo, seus perfis e outros conteúdos criados por eles em comunidades de redes sociais). Além de mostrarem que honeypots sociais são precisos na identificação de *spammers*, eles propuseram um

método de aprendizado de máquina para detectar *spammers* nesses dois sistemas.

Gao *et al.* [10] mediram e analisaram tentativas de disseminar conteúdo spam no Facebook. Baseados em uma base de dados de mensagens de mural do Facebook, eles identificaram mensagens de campanhas spam. Para isso, construíram um modelo onde cada mensagem postada representa um nó e dois nós são conectados se compartilham uma mesma URL ou se compartilham conteúdo de texto semelhante. Este processo criou uma série de subgrafos conectados que compartilhavam mensagens de murais suspeitas. Usando evidências de comportamento dual de atividades em rajada e comunicação distribuída, eles identificaram subconjuntos de mensagens que exibiam propriedades de campanhas de spam maliciosas. E ao analisaram essas mensagens, descobriram que *phishing* é de longe o ataque mais popular no Facebook.

Embora esses métodos tenham inspirado a abordagem utilizada aqui, nosso trabalho é complementar a eles, já que investigamos spam em um ambiente diferente, identificando suas características específicas, como atributos de proximidade geográfica, que nos permitem diferenciar com precisão as classes de spam em avaliações de locais.

2.3 Caracterização de LBSNs

Existem muitos esforços que tentam caracterizar e compreender o uso de LBSNs. Particularmente, Scellato *et al.* [19] analisaram as propriedades sociais, geográficas e geosociais das redes sociais que fornecem informações de localização sobre seus usuários. Eles mostraram que LBSNs são caracterizadas pela curta distância geográfica em laços de amizade, enquanto que em outros tipos de redes sociais, como Twitter e o LiveJournal, os usuários têm ligações de comprimentos heterogêneos. Complementarmente, Allamanis *et al.* [1] modelaram e identificaram padrões da evolução temporal em LBSNs. Baseados em *snapshots* da rede social do Gowalla, incluindo os locais visitados pelos usuários e suas conexões sociais, eles descobriram que o grau dos nós e a distância geográfica simultaneamente influenciam na criação de um novo *link* social e também, que a popularidade de um local e a popularidade de usuários que visitam esse local ajudam a predizer quais conexões sociais serão estabelecidas.

Uma análise de três LBSNs, a saber, Foursquare, Gowalla e Brightkite, identificou as principais propriedades dos grafos que conectam os usuários destes sistemas [20]. Como exemplo, usuários de LBSN possuem fraca correlação positiva entre o número de

amigos e a distância geográfica média entre eles. Além disso, usuários que participam de triângulos sociais possuem triângulos geograficamente mais amplos à medida que o grau deles aumenta. Também, *links* geograficamente mais longos tendem a surgir entre usuários com mais amigos, enquanto que *links* de usuários conectados com menos amigos tendem a ser muito mais curtos. Adicionalmente, Noulas *et al.* [18] analisaram comportamentos de usuários, suas dinâmicas de *check-in* e a presença de padrões geo-temporais no Foursquare. Eles observaram que padrões geo-temporais dos usuários e informações de *check-in* indicam um consenso geral da atividade do usuário em um determinado tempo e lugar. Desse modo, sistemas de recomendações poderiam se beneficiar dessas informações em suas aplicações.

Em um esforço recente, Vasconcelos *et al.* [22] coletaram o Foursquare para caracterizar o comportamento de usuários com base em informações de *tips*, *dones* e *to-Dos*. Usando o algoritmo de agrupamento Maximização de Expectativas, eles agruparam usuários em quatro grupos, sendo um deles caracterizado por conter um grande número de avaliações spam. Desse modo, eles apresentaram a primeira evidência de spam em LBSNs.

2.4 Detecção de Spam em Avaliações

No contexto de avaliações sobre produtos, Jindal and Liu [14] investigaram a detecção de opinião spam em avaliações de produtos, com base na análise de avaliações da amazon.com. Opiniões spam são opiniões falsas que deliberadamente enganam os leitores quando fazem comentários positivos sem mérito do objeto-alvo, a fim de promover esse objeto e/ou quando fazem comentários negativos injustos ou mal-intencionados, a fim de prejudicar a reputação desse objeto. Eles propuseram um modelo para detectar opiniões prejudiciais, com base em avaliações duplicadas (cópias). Este modelo inspirou algumas métricas propostas em nosso trabalho.

Recentemente, Molavi *et al.* [15] abordaram o problema em que os usuários criam múltiplas identidades e usam essas identidades para fornecer avaliações positivas sobre seu próprio conteúdo ou avaliações negativas sobre o conteúdo de outros. Eles desenvolveram um sistema chamado Iolaus para mitigar o efeito da manipulação de serviços de classificação de conteúdo online, como avaliações de locais em LBSNs.

Diferentemente desses esforços, nosso trabalho explora outros tipos de spam em avali-

ações de LBSNs, sendo portanto, complementar aos trabalhos deles.

Capítulo 3

Coleção de Dados

No intuito de avaliar a nossa proposta para detectar diferentes classes de spam, precisamos de um coleção de teste de avaliações rotuladas entre as categorias identificadas, a saber, comercial local, poluidora, boca-suja e não-spam. No entanto, não temos conhecimento de nenhuma coleção deste tipo disponível publicamente para alguma LBSN, sendo necessário construirmos uma.

Neste trabalho foram utilizadas avaliações postadas no Apontador. Com 15 milhões de usuários distintos mensais, o Apontador é um *website* brasileiro de anúncio de locais e serviços que conta com uma base de dados georreferenciada contendo aproximadamente $7,5$ milhões de pontos de interesses no Brasil a serem pesquisados.

Nós construímos nossa base de dados a partir de uma coleção de avaliações de locais rotulada manualmente como "spam" ou "não-spam" por moderadores do próprio Apontador. Nossa base da dados é composta por dois conjuntos da dados, um contendo avaliações classificadas como poluidora, comercial local, boca-suja e não-spam e outro contendo dados que coletamos, a fim de melhorar os atributos utilizados para diferenciar as classes de avaliações. Em seguida, descrevemos os dois conjuntos de dados.

3.1 Base da Dados Rotulada

Embora spam apresente diferentes aspectos em diferente ambientes, é definido na maioria de suas formas como mensagem eletrônica não solicitada, principalmente propaganda enviada indiscriminadamente a usuários [13]. Em LBSNs, spam ocorre principalmente

na forma de avaliações que visam difundir propaganda.

Diferentemente do Foursquare, que permite que pessoas reivindiquem ser o dono do seu local de negócio, que pode ter sido cadastrado no sistema por outra pessoa, o Apontador não possui essa funcionalidade, permite que qualquer usuário cadastre um local e não necessariamente esse usuário é o dono do local. Como consequência, o Apontador tem sido processado por empresas que acreditam estar sendo difamadas nas avaliações postadas em seus locais. Neste cenário, pudemos obter uma base de dados do Apontador contendo avaliações rotuladas manualmente como "spam" ou "não-spam" por seus moderadores. Eles inspecionaram manualmente avaliações postadas durante o mês de Setembro de 2011, identificando **3.668** avaliações como spam. Para que pudéssemos utilizar uma base da dados balanceada, ou seja, com o mesmo número de instâncias para ambas as classes, o Apontador também nos forneceu 3.668 avaliações não-spam do mesmo período do ano.

Com o intuito de identificar os diferentes tipos de spam em avaliações, pedimos a um grupo de voluntários do nosso grupo de pesquisa que fizessem uma verificação manual das avaliações spam. Ao mesmo tempo, como a classificação manual do Apontador depende de julgamento humano para decidir quando uma avaliação é spam ou não, também investigamos se os voluntários concordavam com a classificação realizada pelos moderadores do Apontador. Sendo assim, pedimos aos voluntários que fizessem uma verificação manual de todas as avaliações spam, classificando-as em spam ou não-spam e, ao mesmo tempo, que fornecessem um nome que fosse capaz de descrever uma categoria da avaliação.

Os voluntários classificaram **130** avaliações como não-spam e **3.538** como spam. Apesar de $3,5\%$ de avaliações terem sido classificadas como não-spam, observou-se um alto nível de concordância com a classificação realizada pelo Apontador, o que reflete um alto nível de confiança nessa classificação humana. Sendo assim, decidimos considerar para o nosso estudo a base de dados obtida do Apontador retirando apenas as avaliações que classificamos como não-spam.

Ao analisar as categorias fornecidas, pudemos identificar três classes de avaliações spam: comercial local, poluidora e boca-suja. Avaliações comerciais locais são propagandas sobre o local alvo ou sobre serviços relacionados ao local. Por exemplo, em um barzinho, um *spammer* postou a seguinte avaliação comercial local:

"DOMINGUES VERDURAS FONE (11) 35442210 CEL (11) 88222290"

Poluidoras são avaliações que possuem conteúdo irrelevante ou não relacionado com o local ou mesmo perguntas, fazendo com que a área reservada para avaliações se torne um tipo de SAC (Serviço de Atendimento ao Consumidor). A seguir temos um exemplo de avaliação poluidora, que foi postada em uma imobiliária:

"Boa tarde. Não consigo de jeito nenhum falar com vocês.... Tem uma casa p alugar na rua ponta grossa, tenho muito interesse. Qual outro telefone que vocês tem"

Finalmente, bocas-sujas são avaliações caracterizadas por conter comentários agressivos sobre o local, seu dono ou outro usuário, geralmente contendo palavras ofensivas e de baixo calão. Um exemplo de avaliação boca-suja encontrada na nossa base de dados pode ser visto a seguir:

"Essa empresa é péssima pois não paga seus colaboradores no dia certo. E seu dono é semi-analfabeto e sua equipe administrativa é despreparada..."

A Tabela 3.1 resume como as avaliações rotuladas estão distribuídas entre as classes de spam.

Tabela 3.1: Classe de spam

Classe	**Número de Avaliações**	**Porcentagem**
Comercial Local	1.063	30,1%
Poluidora	1.716	48,5%
Boca-suja	759	21,4%
Total	3.538	100%

Com o intuito de utilizar uma base de dados balanceada, selecionamos aleatoriamente 3.538 avaliações classificados como não-spam. Em síntese, nossa base de dados classificada contém **7.076** avaliações divididas igualmente como spam e não-spam. As avaliações spam têm uma segunda classificação, sendo 30,1% delas classificadas como comercial local, 48,5% como poluidora e 21,4% como boca-suja. Essas avaliações foram postadas por **4.494** usuários distintos em **5.585** locais diferentes.

A Figura 3.1 apresenta os conjuntos de usuários distintos de cada uma das classes que temos na base de dados. O diagrama mostra que um mesmo usuário posta avaliações classificadas em mais de uma classe, indicando um comportamento dual de *spammer*,

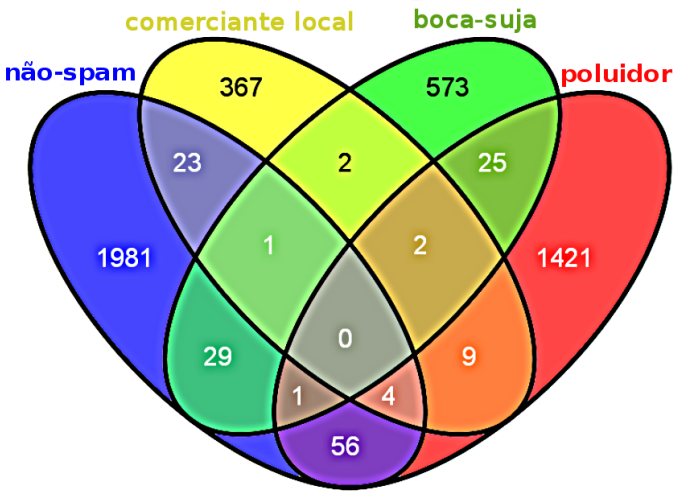


Figura 3.1: Diagrama de Venn dos usuários distintos

que as vezes tenta se comportar como um usuário legítimo. Com base nessa observação, decidimos focar nossas análises e experimentos de classificação nas avaliações ao invés de usuários.

Além dos rótulos das classes, cada avaliação contém as seguintes informações:

- conteúdo da avaliação;
- data em que a avaliação foi postada;
- quantidade de *clicks* no link "Esta avaliação me ajudou" da avaliação;
- quantidade de clicks no link "Reportar abuso" da avaliação;
- identificador único do usuário que fez a avaliação;
- identificador único do local em que a avaliação foi postada;
- identificador único da avaliação.

3.2 Dados Coletados do Apontador

A base da dados cedida pelo Apontador não contém informações suficientes do local e do usuário que nos permitam explorar atributos valiosos capazes de diferenciar as diferentes classes de avaliação. No entanto, através da identificação única dos locais

e dos usuários, conseguimos coletar tais informações adicionais utilizando a API do Apontador[1], que nos permitiram explorar atributos relacionados com locais e usuários. Nós desenvolvemos um coletor em Python para reunir informações para cada local que apareceu nas avaliações da base de dados, totalizando 5.585 locais distintos. Cada local coletado contém as seguintes informações:

- identificação única do local;
- nome;
- descrição;
- contador de *clicks*;
- número de avaliações postadas no local;
- número de recomendações;
- categoria (por exemplo, restaurante, hotel, hospital e etc.);
- endereço e telefone;
- latitude e longitude;
- informações sobre o usuário que fez o cadastro do local (ou seja, o dono do local).

Além de coletar locais, desenvolvemos um segundo coletor para reunir informações sobre a rede social do usuário (lista de seguidores e seguidos), assim como todas as avaliações postadas por ele. Ao coletar a lista de seguidores e seguidos de um usuário, novos usuários foram descobertos e também foram coletados. Nós executamos este processo de forma recursiva até que todos os usuários descobertos fossem coletados, o que corresponde a um componente fracamente conectado do grafo da rede social do Apontador. Através dos 4.494 usuários distintos que apareceram na base de dados classificada, descobrimos e coletamos um grafo de rede social contendo **137.464** usuários. Para cada usuário coletado, reunimos as seguintes informações:

- nome;
- número de locais registrados pelo usuário;

[1] http://api.apontador.com.br/

- número de avaliações postadas;
- número de fotos postadas.

Capítulo 4

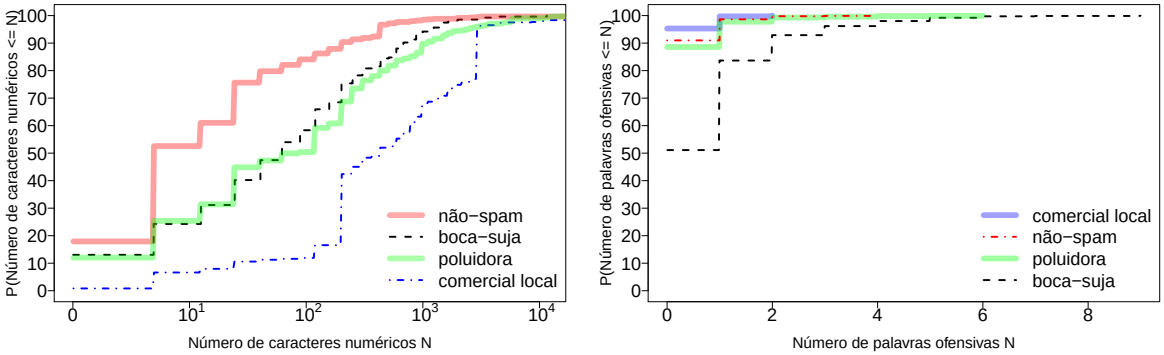
Identificando Comportamentos em LBSN

Ao contrário de usuários comuns de LBSNs, pessoas que postam spam têm como objetivo o conteúdo comercial (por exemplo, propaganda), a auto-promoção e a depreciação de ideias e reputação [13]. O comerciante local, o poluidor, o boca-suja e o usuário não-spam têm objetivos diferentes no sistema, e portanto, esperamos que eles também se comportem de maneira diferente (por exemplo, em relação ao que eles postam e quantas vezes usam o sistema) para alcançar os seus objetivos. Sendo assim, neste capítulo vamos analisar vários atributos que refletem o comportamento do usuário no sistema com o objetivo de investigar o poder discriminativo desses atributos para distinguir cada classe de avaliação das outras. Consideramos quatro grupos de atributos: atributos de conteúdo, atributos de usuário, atributos de local e atributos sociais, discutidos a seguir.

4.1 Atributos de Conteúdo

Atributos de conteúdo são propriedades do texto postado pelo usuário numa avaliação. Os seguintes atributos foram gerados para cada avaliação da nossa base da dados:

- número de palavras do texto;
- número de caracteres numéricos no texto (isto é, 1,2,3);
- número de palavras ou expressões que estão contidas numa lista popular de con-



(a) Número de caracteres numéricos (b) Número de palavras ofensivas

Figura 4.1: CDFs dos atributos de conteúdo

teúdo spam[1] e num conjunto de regras em português do SpamAsassin[2] que contém expressões regulares para conteúdo spam que é comum aparecer no corpus de e-mails;

- número de caracteres maiúsculos;
- número de palavras com todos os caracteres em maiúsculo;
- número de URL's no texto;
- número de endereços de e-mail;
- número de telefones;
- número de informações de contato no texto (que representa a soma do número de endereços de e-mail e do número de telefones);
- número de palavras ofensivas no texto;
- valor de "Tem ofensiva palavra";
- valor do coeficiente Jaccard.

Para calcular a métrica **número de palavras ofensivas no texto**, construímos uma lista de palavras ofensivas em português com base em listas disponíveis na Web

[1]Lista de conteúdo spam: `codex.wordpress.org/Spam_Words`
[2]Regras do SpamAsassin `github.com/ppadron/spamassassin-pt-br`

e com a ajuda de voluntários, que sugeriram palavras que ainda não tínhamos listado. Esta lista está disponível em uma página do GitHub[3] para que outras pessoas possam adicioná-la à sua lista de moderação. Um segundo atributo relacionado com palavras ofensivas é o **Tem ofensiva palavra**. Se a avaliação tem pelo menos uma palavra ofensiva, o valor desse atributo é 1, caso contrário, 0.

Também calculamos o **coeficiente Jaccard** [2] entre o texto da avaliação e o texto das outras avaliações do mesmo usuário. O cálculo é feito utilizando uma comparação baseado nos elementos 2-gramas (duas palavras em sequência) do texto. O coeficiente Jaccard $J(A, B)$ entre duas avaliações A e B é a interseção de seus elementos 2-gramas dividido pela união dos mesmos, como mostra a Equação 4.1. Um coeficiente Jaccard J igual a 0 significa que as duas avaliações não têm elementos em comum, enquanto J próximo de 1 indica que ambas as avaliações compartilham a maioria de seus elementos.

$$J(A, B) = |A \cap B|/|A \cup B| \tag{4.1}$$

Além desses atributos relacionados ao conteúdo do texto, também foram considerados outros dois atributos que estão relacionados com a avaliação, mas não diretamente com o texto. Esses atributos são: número de *clicks* no link "Esta avaliação me ajudou" e número de *clicks* no link "Reportar abuso". No total foram considerados 18 atributos relacionados ao conteúdo das avaliações.

Para ilustrar o potencial poder discriminativo dos atributos extraídos do conteúdo das avaliações, podemos observar na Figura 4.1(a) que as avaliações comerciais locais têm mais caracteres numéricos no texto do que as outras avaliações. Isso acontece porque é comum publicar contatos junto com uma propaganda. De fato, notamos que aproximadamente 55.0% das avaliações comerciais locais têm pelo menos um contato no texto. Outro atributo interessante, número de palavras ofensivas no texto de cada tipo de avaliação, está representado na Figura 4.1(b). O gráfico mostra que aproximadamente 50.0% das avaliações bocas-sujas têm pelo menos uma palavra ofensiva no texto. Enquanto que para as outras classes, cerca de 90.0% das avaliações não têm palavras ofensivas no texto.

[3]Lista de palavras ofensivas: urlgithub.com/spam-detection/badwords-pt-br

4.2 Atributos de Usuário

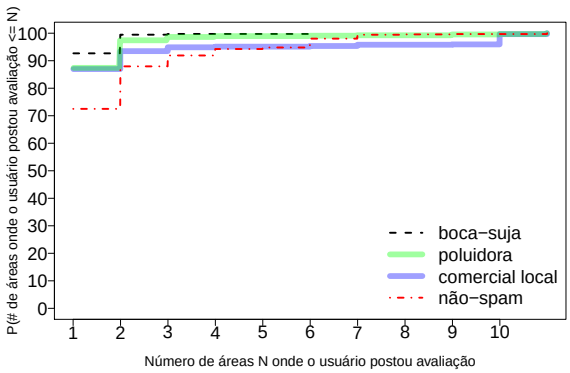
O segundo grupo de atributos consiste de propriedades específicas do comportamento do usuário no sistema. Consideramos os seguintes atributos de usuário:

- número de locais cadastrados pelo usuário;
- número de avaliações postadas pelo usuário;
- número de fotos postadas pelo usuário;
- distância entre todos os locais avaliados pelo usuário;
- número de áreas diferentes onde o usuário postou uma avaliação.

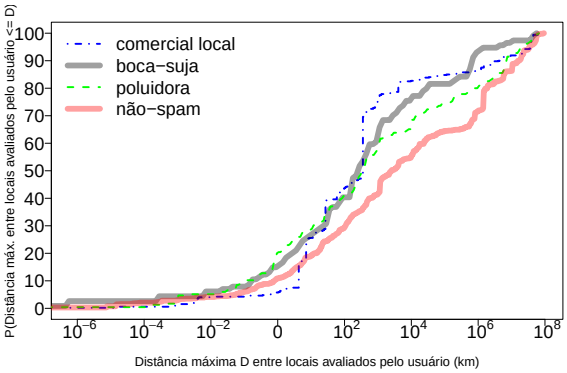
Para calcular o atributo **distância entre todos os locais avaliados pelo usuário**, medimos a distância entre cada par de locais avaliados pelo usuário, considerando apenas usuários que avaliaram mais de um local diferente. Caso contrário, o valor deste atributo é 0. Em seguida, calculamos a distância entre cada par de locais, utilizando a informação de latitude e longitude deles. E para calcular o atributo **número de áreas diferentes onde o usuário postou uma avaliação**, definimos uma área como sendo a posição geográfica do primeiro local avaliado pelo usuário. Se o próximo local avaliado pelo mesmo usuário está em um raio de 50 km da primeira área, então o local pertence à mesma área. Caso contrário, uma nova área é criada, e assim por diante.

Neste conjunto de atributos pudemos investigar as proximidades geográficas dos locais avaliados pelos quatro tipos de usuários encontrados na base de dados. Para isso, foram consideradas apenas avaliações que foram postadas por usuários que fizeram pelo menos duas avaliações em locais diferentes no Apontador. Para cada classe de avaliação, a porcentagem de avaliações que pertencem a usuários que postaram pelo menos duas avaliações em locais diferentes são: 48.6% das avaliações bocas-sujas, 80.8% das comerciais locais, 38.4% das poluidoras e 54.8% das avaliações não-spam. Podemos notar que a menor porcentagem de avaliações pertence à classe poluidora, indicando que *spammers* do tipo poluidores tipicamente usam o sistema apenas uma vez, ou seja, eles podem ser usuários iniciantes.

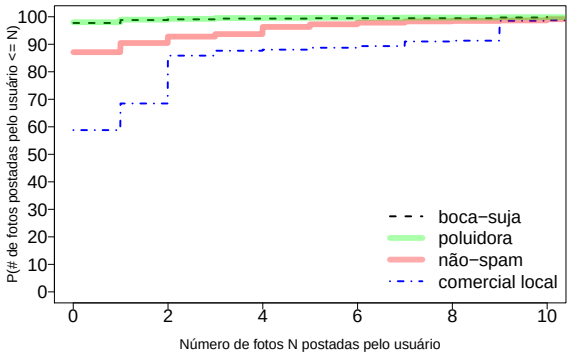
A Figura 4.2(a) indica o número de diferentes áreas onde o usuário postou avaliações. Podemos observar que apenas 8% dos comerciantes locais postaram avaliações em mais de uma área. Isso indica que eles agem em certas regiões e, portanto, visam postar seus



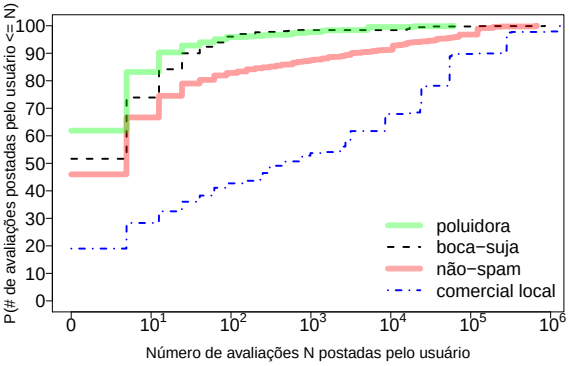
(a) Número de áreas onde o usuário postou uma avaliação



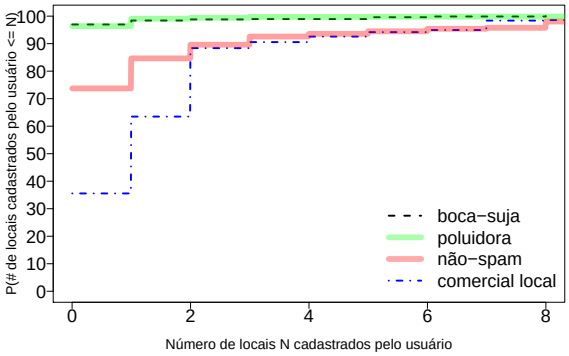
(b) Distância máxima entre locais avaliados pelo usuário



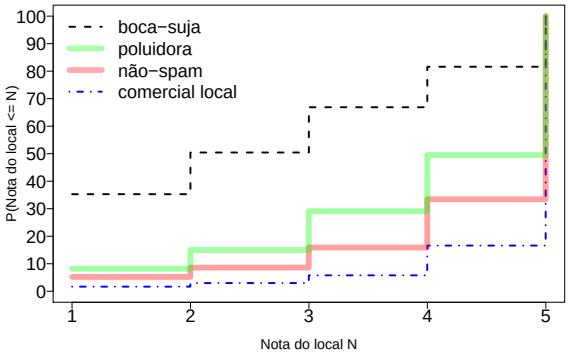
(c) Número de fotos postadas pelo usuário



(d) Número de avaliações postadas pelo usuário



(e) Número de locais cadastrados pelo usuário



(f) Nota do local

Figura 4.2: CDFs dos atributos de usuário e de local

anúncios em áreas específicas. Por outro lado, observamos que a maior porcentagem de usuários que postaram avaliações em mais de uma área, aproximadamente 30%, pertence aos não-spammers. De fato, não-spammers tendem a postar avaliações em locais muito distantes um do outro. Para medir isso, calculamos a maior distância entre todos os locais onde um usuário postou uma avaliação. Intuitivamente, pode-se esperar que não-spammers tendam a visitar e postar avaliações apenas em locais próximos de suas casas ou trabalhos, enquanto que *spammers* tendam a postar avaliações em locais aleatórios. No entanto, quando analisamos a maior distância entre locais onde o usuário postou uma avaliação, observamos que 60% dos não-*spammers* têm a maior distância entre locais maior que 1000 km. Em comparação com os comerciantes locais, apenas 20% deles têm a maior distância entre locais maior do que este mesmo valor. Sendo assim, podemos supor que usuários comuns usam amplamente LBSNs quando viajam ou visitam diferentes regiões geográficas, a fim de receber avaliações sobre locais que não estão familiarizados, enquanto que comerciantes locais e outros *spammers* não têm a mesma motivação para fazer isso. Estas observações sugerem que comportamentos de usuário extraídos de atributos de proximidades geográficas podem ajudar a diferenciar as classes de avaliação.

Outra descoberta relacionada com os atributos do usuário é que os usuários bocas-sujas e poluidores interagem menos com as ferramentas do sistema. A Figura 4.2(d) mostra que cerca de 74% das avaliações bocas-sujas e 83% das avaliações poluidoras pertencem a usuários que postaram até 10 avaliações, enquanto que apenas 28% de avaliações comerciais locais pertencem a usuários que postaram até 10 avaliações. Isso indica que os *spammers* bocas-sujas e poluidores têm comportamentos semelhantes e interagem menos com as ferramentas do sistema do que os comerciantes locais. Outro atributo que confirma isso é o número de fotos postadas pelo usuário. A Figura 4.2(c) mostra a CDF deste atributo. Observamos que 97% das avaliações bocas-sujas e poluidoras pertencem a usuários que não publicaram nenhuma foto, enquanto que menos de 60% das avaliações comerciais locais pertencem a usuários que não postaram nenhuma foto.

Finalmente, observamos que os comerciantes locais cadastraram mais locais do que os outros usuários, confirmando que eles têm uma boa interação com o sistema. Na verdade, a sua interação é maior até do que a interação dos usuários não-spam. A Figura 4.2(e) mostra que apenas 35% das avaliações comerciais locais foram postadas por usuários que não cadastraram nenhum local no sistema, enquanto que cerca de 74% das avaliações não-spam e 94% das avaliações bocas-sujas ou poluidoras pertencem a

usuários que não cadastraram nenhum local. Este fato indica que, além de publicar anúncios, os comerciantes locais cadastram seus negócios no Apontador.

4.3 Atributos de Local

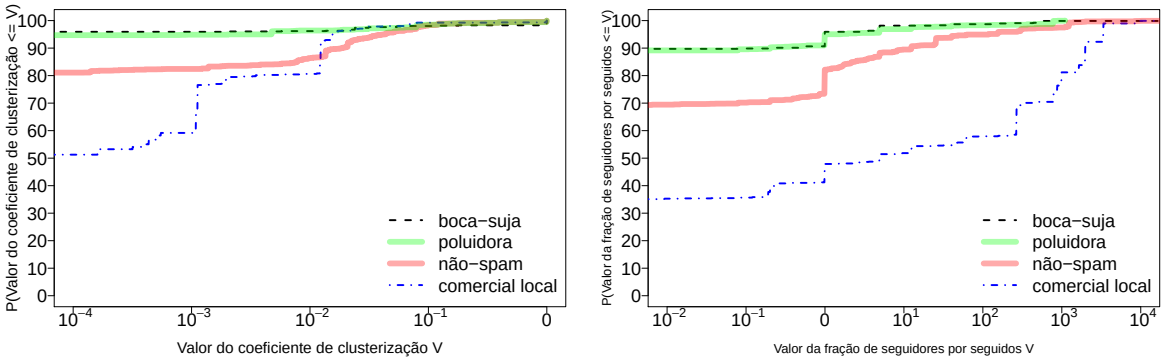
O terceiro grupo de atributos está relacionado com o local onde a avaliação foi postada. Nós selecionamos cinco atributos de local:

- número de *clicks* na página do local;
- número de avaliações do local;
- nota do local (numa escala de classificação de 5 pontos, sendo 1 a pior nota e 5 a melhor);
- número de *clicks* no botão "recomendo" do local;
- número de *clicks* no botão "não recomendo" do local.

Analisando este conjunto de atributos, descobrimos que avaliações comerciais locais são em sua maioria postadas em locais com notas altas. A Figura 4.2(f) mostra que mais de 80% das avaliações comerciais locais são postados em locais que têm nota 5, o que indica que comerciantes locais tendem a postar seus anúncios em locais bem avaliados por outros usuários, ou seja, locais populares. Este comportamento pode estar relacionado com tentativas de aumentar a visibilidade de suas avaliações. Também podemos notar que avaliações bocas-sujas tendem a ser postadas em locais de má qualidade, visto que 70% delas foram postadas em locais que têm nota 1, 2 ou 3. Isso pode indicar que este local alvo é realmente um local ruim ou ainda, que essas avaliações bocas-sujas fazem parte de um ataque para diminuir a reputação (nota) do local.

4.4 Atributos Sociais

O quarto conjunto de atributos captura as relações estabelecidas entre os usuários através da rede social. A ideia é que estes atributos possam capturar padrões específicos de interação que possam ajudar a diferenciar os diferentes tipos de usuários encontrados na base de dados. Nossa representação da rede social dos usuários é dada por um



(a) Coeficiente de clusterização

(b) Fração de seguidores por seguidos

Figura 4.3: CDFs dos atributos sociais

grafo $G(V, A)$, onde V é o conjunto de vértices e A é o conjunto de arestas. Seja G_i um grafo direcionado que modela a rede de usuários, V_i o conjunto de usuários e A_i o conjunto de ligações de amizade (seguidor e seguido) entre usuários. Então, uma aresta $(v \rightarrow u) \in A_i$ se, e somente se, v é seguidor de u. Vale ressaltar que G_i é direcionado, ou seja, $(v \rightarrow u) \in A_i$ não implica $(u \rightarrow v) \in A_i$.

Nós selecionamos os seguintes atributos extraídos da rede social, os quais capturam o nível de interação (social) do usuário correspondente:

- coeficiente de clusterização;
- *betweenness*;
- reciprocidade;
- assortatividade;
- grau de entrada (número de seguidores);
- grau de saída (número de seguidos);
- grau;
- fração do número de seguidores pelo número de seguidos;
- Pagerank.

O **grau de entrada** de um vértice v, $k_e(v)$, é o total de arestas que incidem no vértice. Da mesma forma, o **grau de saída** de um vértice v, $k_s(v)$, é o total de arestas que têm origem no vértice. O **grau** de de um vértice v é dado pela soma de $k_e(v)$ e $k_s(v)$.

O **coeficiente de clusterização** de um vértice v, $cc(v)$, é a razão entre do número de arestas existentes entre os vizinhos de um vértice e o total de arestas possíveis entre os vizinhos de v. Funciona como uma medida da densidade da comunicação, não apenas entre dois usuários, mas entre os vizinhos dos vizinhos.

Outra métrica interessante que calculamos é a **reciprocidade** de cada usuário. A reciprocidade R de um vértice é calculada como mostra a Equação 4.2, onde $Out(v)$ é o conjunto de vértices que o vértice v segue (seguidos) e $In(v)$ é o conjunto de vértices que seguem o vértice v (seguidores). A reciprocidade mede a probabilidade de um vértice ser seguido por cada vértice que ele/ela segue.

$$R(x) = \frac{|Out(v) \cap In(v)|}{|Out(v)|} \tag{4.2}$$

Já a **assortatividade** de um usuário é definida como a fração do grau (de entrada ou de saída) do vértice pela média do grau (de entrada ou de saída) de seus vizinhos. Calculamos a assortatividade de um vértice para os quatro tipos de correlações grau/grau (ou seja, entrada/entrada, entrada/saída, saída/entrada e saída/saída).

A métrica ***betweenness*** está relacionada à centralidade dos vértices, medindo o número de caminhos mínimos que passam por esse vértice. O *betweenness* de um vértice indica a importância desse vértice no grafo em termos de sua localização. Vértices com maior *betweenness* estão em mais caminhos mínimos e, portanto, são mais importantes para a estrutura do grafo. Em outras palavras, seja $\sigma(u, v)$ o total de caminhos mínimos entre u e v, e $\sigma_w(u, v)$ o número de caminhos mínimos entre u e v que passam por w. Então, o *betweenness* do vértice w pode ser calculado como mostra a Equação 4.3.

$$B(w) = \sum_{u \in V, v \in V} \frac{\sigma_w(u, v)}{\sigma(u, v)} \tag{4.3}$$

Finalmente, também calculamos o algoritmo **Pagerank** [6] no grafo social. A intu-

ição por trás do PageRank é que um vértice é importante se existem muitos vértices apontando para ele (seguidores) ou se existem vértices importantes, ou seja, bem ranqueados, apontando para ele. O pesos numéricos que o algoritmo PageRank assinala para cada vértice podem ser usados como indicadores da importância dos vértices em termos de sua participação na LBSN. O cálculo do PageRank (PR) de um vértive v, $PR(v)$, pode ser definido como mostra a Equação 4.4, onde $In(v)$ é o conjunto de vértices que apontam para o vértice v (seguidores), N_u denomina o número de arestas que saem do vértice u e o parâmetro d é um fator que pode ter valor entre 0 e 1.

$$PR(v) = (1 - d) + d \sum_{u \in In(v)} \frac{PR(u)}{N_u} \tag{4.4}$$

Analisando os atributos sociais, observamos que os *spammers* bocas-sujas e poluidores têm menos seguidores do que seguidos. Como os *spammers* podem não estar interessados em estabelecer relacionamentos verdadeiros, podemos esperar diferentes padrões para *spammers* e usuários comuns em termos do número de entrada e saída de ligações do grafo do Apontador. Porém, em vez de simplesmente medir o número de seguidores (grau de entrada) e seguidos (grau de saída), também calculamos a fração de seguidores por seguidos, como mostra a Figura 4.3(b). Podemos ver que bocas-sujas e poluidores têm um valor menor da fração de seguidores por seguidos em comparação com comerciantes locais e não-*spammers*. De fato, algumas classes de *spammers* podem tentar seguir outros usuários na tentativa de ser seguido de volta. No entanto, a maioria dos usuários não seguem *spammers*, o que produz uma fração menor de seguidores por seguidos para os bocas-sujas e poluidores. Ainda assim, os comerciantes locais mostraram um valor maior da fração, maior até do que o valor para não-*spammers*. Isto confirma que esta classe de *spammmer* tem interagido mais com o sistema do que os não-*spammers*.

Observamos também que os amigos dos usuários bocas-sujas e poluidores não estão bem conectados. O coeficiente de clusterização mede o quão conectados são os amigos (seguidores e seguidos) de um usuário. A Figura 4.3(a) mostra a CDF para o coeficiente de clusterização dos usuários. Podemos ver que os comerciantes locais e não-*spammers* são mais fortemente conectados do que poluidores e bocas-sujas.

4.5 Importância dos Atributos

Nós avaliamos o poder relativo dos 44 atributos calculados para discriminar cada classe das outras através de um ranqueamento dos atributos feito pelo **cálculo de importância de atributo** do classificador *Random Forest* (RF) [5], que é usado na seção 5. O ranqueamento que utilizamos foi o Diminuição Média da Precisão (*Mean Decrease Accuracy* - MDA) de um atributo. Quanto mais a precisão da floresta aleatória (*random forest*) diminui devido à adição de um único atributo, menos importante é considerado o atributo e, portanto, atributos com um MDA grande são menos importantes para a classificação dos dados. A Tabela 4.1 sumariza os resultados, mostrando a posição no *ranking* dos atributos de cada conjunto (conteúdo, usuário, local e social) de acordo com o ranqueamento feito pelo cálculo de importância de atributo. Podemos notar que os 15 atributos mais discriminativos são distribuídos entre as quatro categorias, o que demonstra a importância de termos investigado cada uma delas.

Analisando os quatro conjuntos de atributos, observamos que poluidores e bocas-sujas têm um comportamento semelhante no sistema. Por exemplo, eles interagem menos com as ferramentas do sistema e têm menos seguidores do que seguidos. Em contrapartida, os comerciantes locais têm um comportamento completamente diferente dos outros *spammers*. Eles têm uma boa interação com o sistema e um valor elevada da fração seguidores por seguidos em comparação com os outros usuários. Na maioria dos gráficos, podemos diferenciar o comportamento de não-*spammers* com o comportamento de *spammers*, mas, surpreendentemente, foi possível observar que os comerciantes locais usam mais as ferramentas do sistema do que não-*spammers*, como podemos observar na Figura 4.2(e), que mostra que os comerciantes locais cadastram mais locais no sistema do que os não-*spammers*.

Tabela 4.1: *Ranking* dos atributos

Categoria	Ranking MDA	Descrição
Conteúdo	**2**	Número de URLs no texto
18 atributos	**4**	Número de contato no texto
	5	Número de endereços de e-mail no texto
	6	Número de palavras
	10	Número de palavras ofensivas
	12	Número de caracteres numéricos
	13	Número de caracteres maiúsculos
	17	Valor de "Tem palavra ofensiva"
	18,19,21,26,29	Valor do coeficiente Jaccard (mediana, min., médio, max., desvio)
	23	Número de palavras com todos os caracteres em maiúsculo
	24	*Clicks* no link "Esta avaliação me ajudou"
	25	Número de telefones no texto
	41	Número de palavras ou regras spam
	43	*Clicks* no link "Reportar abuso"
Usuário	**9**	Número de avaliações postadas pelo usuário
9 atributos	**15**,28,31,34,38	Distância entre os locais avaliados (mediana, max., desvio, médio, min.)
	16	Número de locais cadastrados pelo usuário
	30	Número de áreas diferentes onde o usuário postou uma avaliação
	32	Número de fotos postadas pelo usuário
Local	**1**	Número de avaliações do local
5 atributos	**3**	Nota do local
	8	*Clicks* no botão "Não recomendo" do local
	11	*Clicks* no botão "Recomendo" do local
	20	*clicks* na página do local
Social	**7**	Fração de seguidores por seguidos
12 atributos	**14**	Grau de entrada (número de seguidores)
	22	Grau
	27,35,36,44	Assortatividade (entrada/saída, saída/entrada, entrada/entrada e saída/saída)
	33	Pagerank
	37	Betweenness
	39	Reciprocidade
	40	Coeficiente de clusterização
	42	grau de saída (número de seguidos)

Capítulo 5

Detectando Tipos de Spam em Avaliações

Neste capítulo, vamos investigar a viabilidade da aplicação de um algoritmo de aprendizado supervisionado, a fim de detectar avaliações boca-suja, poluidora e comercial local no Apontador. Para fazer isso, o algoritmo de aprendizagem utiliza os atributos descritos na seção anterior, construindo um modelo de classificação por meio da análise do conjunto de instâncias de treinamento (avaliações) representadas por um vetor de valores de atributos e um rótulo de classe. Numa segunda etapa, o modelo de classificação é utilizado para classificar instâncias de teste (avaliações) entre as classes: não-spam, boca-suja, poluidora ou comercial local.

Neste trabalho, estamos usando um conjunto de treinamento contendo dados rotulados fornecidos pelo Apontador (ver Capítulo 3). Em cenários práticos, várias iniciativas podem ser usadas para obter uma base de dados rotulada (por exemplo, voluntários que ajudam a identificar spam, profissionais contratados para periodicamente classificar manualmente uma amostra de avaliações, etc.). Adicionalmente, existem estratégias semi-supervisionadas na literatura que são capazes de obter resultados da classificação próximos de abordagens supervisionadas com uma base de dados rotulada bem mais reduzida [25]. Nosso objetivo aqui é avaliar a eficácia de métodos de aprendizagem supervisionados para a tarefa de detectar as diferentes classes de spam em avaliações que identificamos. Deixamos como trabalho futuro o esforço de reduzir a base de dados rotulada para essa tarefa.

O nosso problema pode ser visto como um problema de classificação hierárquica,

uma vez que há uma hierarquia de classes pré-definida, como mostra a Figura 5.1. Nessa hierarquia, o primeiro nível é composto pelas classes spam (S) e não-spam (NS), já o segundo nível por classes descendentes de spam, a saber, boca-suja (BS), poluidor (PL) e comercial local (CL). Para resolver o problema, duas abordagens de classificação foram considerados: a plana e a hierárquica.

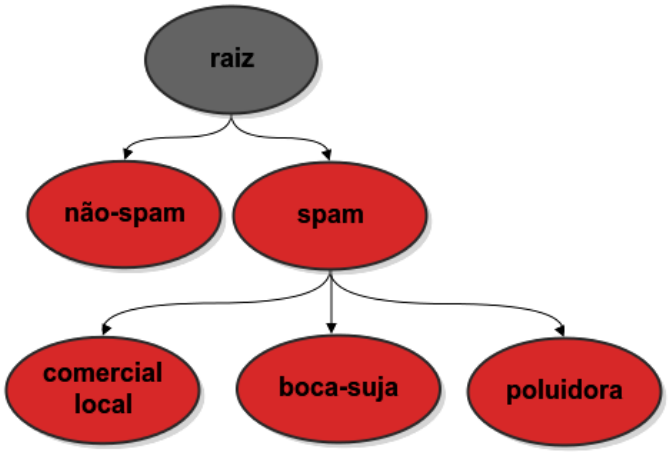


Figura 5.1: Ilustração da hierárquica de classes

A abordagem de classificação plana, que é a maneira mais simples para lidar com problemas de classificação, ignora a hierarquia das classes, realizando predições diretamente nos nós folha, como mostra a Figura 5.2. Sendo assim, um classificador único é treinado para diferenciar as avaliações entre não-spam, boca-suja, poluidora e comercial local. Diferentemente, a abordagem hierárquica leva em conta a hierarquia das classes usando uma perspectiva da informação local. Entre as diferentes formas de utilizar essa informação local, neste trabalho estamos considerando um classificador local por nó pai. Nessa abordagem, para cada nó pai da hierarquia, um classificador é construído para distinguir entre seus nós filhos. Essa abordagem é muitas vezes associada com a estratégia de predição *top-down* na fase de classificação. Portanto, a classificação de uma nova instância é processada um nível de cada vez.

Para o nosso problema, um classificador do nó raiz é treinado para separar avaliações spam de avaliações não-spam, e outro, um classificador do nó spam, é treinado para distinguir avaliações spam entre boca-suja, poluidora e comercial local (nós filhos da classe spam). Depois, durante a fase de classificação, uma instância de teste é classificada no primeiro nível de hierarquia pelo classificador do nó raiz. Se a instância é classificada como spam, então o classificador do nó spam atribuirá uma das classes filhas de spam (boca-suja, poluidora ou comercial local) para essa instância.

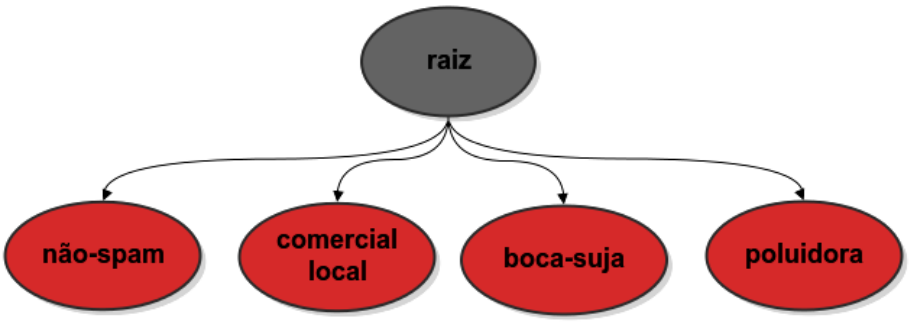


Figura 5.2: Ilustração plana das classes

A seguir, na Seção 5.1, apresentamos as métricas usadas para avaliar os resultados experimentais. Na seção 5.2 descrevemos os métodos de classificação adotados neste trabalho e a configuração experimental utilizada. Os resultados obtidos com as abordagens plana e hierárquica são apresentados nas Seções 5.3 e 5.4, respectivamente. Por fim, na Seção 5.5, discutimos o impacto de reduzir o conjunto de atributos na eficácia da classificação.

5.1 Métricas de Avaliação

Para avaliar a eficácia dos nossos experimentos de classificação, adotamos métricas comumente usadas em Aprendizagem de Máquina e Recuperação de Informação [2]. Para explicar essas métricas no contexto do nosso problema, vamos usar a seguinte matriz de confusão:

		Classe predita			
		NS	***CL***	***PL***	***BS***
	NS	a	b	c	d
Classe	***CL***	e	f	g	h
verdadeira	***PL***	i	j	k	l
	BS	m	n	o	p

Figura 5.3: Matriz de confusão

onde a indica a porcentagem de avaliações não-spam (NS) que foram classificadas corretamente, b indica a porcentagem de avaliações não-spam que foram incorretamente

classificadas como comercial local (CL), c indica a porcentagem de avaliações não-spam classificadas incorretamente como poluidora (PL), e d a porcentagem de avaliações não-spam classificadas incorretamente como boca-suja (BS). O mesmo raciocínio pode ser aplicado para interpretar as outras entradas da matriz de confusão.

As seguintes métricas foram consideradas na nossa avaliação: *recall*, *precision*, *F-measure* (F1), Micro-F1 (ou acurácia) e Macro-F1. O *recall* (R) e o *precision* (P) de uma classe i são definidos como:

$$R_i = \frac{TP_i}{TP_i + FN_i}, \; P_i = \frac{TP_i}{TP_i + FP_i}, \tag{5.1}$$

onde TP_i (*true positive*) é o número de instâncias atribuídas corretamente a classe i, FN_i (*false negative*) é o número de instâncias que pertencem à classe i, mas não são atribuídos à classe i pelo classificador e FP_i (*false positive*) é o número de instâncias que não pertencem à classe i mas são atribuídos incorretamente a classe i pelo classificador. Os valores na diagonal principal da matriz de confusão mostrados na Figura 5.3 representam o *recall* das classes NS, CL, PL e BS.

A métrica *F-measure*, uma forma padrão de resumir *precision* e *recall*, é definida como:

$$F1_i = 2 \times \frac{P_i \times R_i}{P_i + R_i} \tag{5.2}$$

A métrica *F-measure* atinge o seu melhor valor em 1 (indicando uma predição perfeita) e o pior valor em 0. O *F-measure* global de todo o problema de classificação pode ser calculado de duas maneiras diferentes: Micro-F1 e Macro-F1. O Micro-F1 é calculado a partir dos valores de *precision* e *recall* globais para todas as classes, calculados da seguinte forma:

$$R = \frac{\sum_{i=1}^{m} TP_i}{\sum_{i=1}^{m}(TP_i + FN_i)}, \; P = \frac{\sum_{i=1}^{m} TP_i}{\sum_{i=1}^{m}(TP_i + FP_i)}, \tag{5.3}$$

onde m é o número de classes. O Micro-F1 é então definido como:

$$\text{Micro-F1} = 2 \times \frac{P \times R}{P + R} \tag{5.4}$$

Basicamente, essa métrica dá importância igual para cada avaliação, independentemente da sua classe, e portanto, tende a ser dominada pelo desempenho do classificador nas classes majoritárias. Diferentemente, a métrica Macro-F1 dá a mesma importância para cada classe, independentemente de seu tamanho relativo. Portanto, é mais afetada pelo desempenho do classificador nas classes minoritárias. O Macro-F1 é obtido calculando a média dos valores de métrica *F-measure* de cada classe:

$$\text{Macro-F1} = \frac{\sum_{i=1}^{m} F1_i}{m} \tag{5.5}$$

onde m é o número de classes.

5.2 Os Classificadores e a Configuração Experimental

Neste trabalho, os experimentos foram realizados utilizando os classificadores SVM (*Support Vector Machine*)[21] e RF (*Random Forest*)[5], que são o estado da arte em técnicas de classificação. Para a abordagem de classificação plana, foram avaliados os dois classificadores, SVM e RF. Para a abordagem de classificação hierárquica, apenas o classificador que obteve o melhor desempenho na classificação plana foi avaliado.

5.2.1 Descrição e Configuração do SVM

Considerando que as instâncias de treinamento podem ser interpretadas como pontos no espaço, a ideia básica do SVM é encontrar um hiperplano ótimo de separação, isto é, uma fronteira de decisão que separe (com margem maximizada) os dados de treinamento em duas porções de um espaço N-dimensional. Em seguida, dada uma nova instância a ser classificada, ela é mapeada no mesmo espaço e sua classe é escolhida baseado na sua posição em relação ao hiperplano de separação. Para lidar com dados que não são linearmente separáveis, o SVM transforma os dados de treinamento originais para uma dimensão mais elevada usando um mapeamento não-linear. Uma vez que os dados

foram mapeados em um novo espaço dimensional, o SVM procura por um hiperplano de separação linear neste novo espaço. Embora esse método tenha sido originalmente concebido para a classificação binária, diferentes extensões foram propostas na literatura para torná-lo adequado para problemas multi-classe.

Em nossos experimentos, utilizamos a implementação do SVM fornecida pelo Weka [23], um software livre e de código aberto amplamente utilizado para mineração de dados. Utilizamos o *kernel* RBF (*Radial Basis Function*) para permitir que os modelos do SVM conseguissem realizar separações mesmo com fronteiras muito complexas. Com o objetivo de encontrar os melhores parâmetros de ajuste para a base de dados utilizada neste trabalho, executamos um algoritmo de otimização de parâmetros chamado GridSearch, que também é encontrado na ferramenta Weka, para os parâmetros do RBF que podem ser variados em busca de um melhor resultado, o c (custo) e o $gama$. Como resultado deste processo de ajuste de parâmetros, para a abordagem de classificação plana, em que um único classificador é treinado, encontramos os valores $c = 100$ e $gama = 1$, que foram adotados em nossos experimentos.

5.2.2 Descrição e Configuração do Random Forest

O RF é um algoritmo *ensemble*, ou seja, é um técnica de mineração de dados que combina classificadores, predizendo a classe de uma instância elegendo a maioria dos votos feitos pelos classificadores. Sendo assim, o RF constrói muitas árvores de decisão (floresta) e escolhe a classe com maior número de votos em todas as árvores da floresta. Cada árvore de decisão é cultivada a partir de um subconjunto aleatório de instâncias da base de treinamento. Além disso, durante o processo de crescimento da árvore, um subconjunto aleatório dos atributos disponíveis é utilizado para determinar a melhor separação para cada nó da árvore. Para a classificação de uma nova instância (avaliação), a mesma percorre cada uma das árvores de decisão da floresta. Em seguida, cada árvore dá uma classificação para a instância, ou seja, uma classe recebe um voto daquela árvore. A predição do RF é a classe que teve a maioria dos votos.

Os experimentos feitos com o RF foram executados utilizando o algoritmo implementado também pelo software Weka. Com o intuito de achar o melhor conjunto de parâmetros para a nossa base de dados de treinamento, assim como feito para o classificador SVM, executamos o algoritmo de otimização de parâmetros GridSearch para os seguintes parâmetros do classificador: $numFeatures$ (usado na seleção aleatória de atributos) e $numTrees$ (número de árvores a serem geradas). Como resultado, para a abordagem de

classificação plana, encontramos os valores $numFeatures = 7$ e $numTrees = 195$, que foram adotados em nossos experimentos. Quanto à abordagem hierárquica, uma vez que dois classificadores são construídos, encontramos dois conjuntos diferentes de valores de parâmetros: $numFeatures = 6$ e $numTrees = 155$ para o classificador treinado para distinguir instâncias entre spam e não-spam, e $numFeatures = 4$ e $numTrees = 195$ para o classificador treinado para distinguir instâncias entre as classes boca-suja, poluidora e comercial local.

5.2.3 Particionamento da Base de Dados

O desempenho preditivo foi medido usando o método de validação cruzada (CV - *Cross-Validation*) 5-fold $\times$ 10 ($10 \times 5 - CV$). Em cada teste $5 - CV$, a base de dados original é dividida em 5 conjuntos exclusivos, dos quais quatro são usados como dados de treinamento e o conjunto restante é usado para testar o classificador. O processo de classificação é, então, repetido 5 vezes, com cada um dos cinco conjuntos utilizados apenas uma vez como dados de teste, produzindo assim, 5 resultados. Todo o processo $5 - CV$ foi repetido 10 vezes com diferentes sementes (*seeds*) utilizadas para embaralhar a base de dados original, produzindo 50 resultados potencialmente diferentes. Portanto, nossos resultados para cada classificador são médias de 50 execuções. E com 95% de confiança, os resultados apresentados nas Seções 5.3 e 5.4 não diferem da média em mais de 2% e 1%, respectivamente.

5.3 Classificação Plana

Como mencionado anteriormente, na abordagem de classificação plana, um único classificador é treinado a partir da base de dados de treinamento contendo instâncias associadas com as classes não-spam, boca-suja, poluidora e comercial local. Então, dada uma nova instância a ser classificada, o classificador atribui a ela uma dessas classes de treinamento.

A Figura 5.4 mostra a matriz de confusão obtida como resultado de nossos experimentos para a classificação plana usando o classificador SVM. Cada valor apresentado corresponde ao percentual de X avaliações que foram classificadas como avaliação Y, onde X e Y são as classes de avaliações (não-spam (NS), comercial local (CL), poluidora (PL) e boca-suja (BS)). Os valores em negrito na matriz indicam o *recall* das

classes. Como podemos observar, 95% das avaliações não-spam, 71.4% das comercial local, 58% das poluidoras e 43% das boca-suja foram corretamente classificadas pelo SVM. Apesar dos bons resultados obtidos para as classes comercial local e não-spam (*recall* > 70%), uma fração significativa das avaliações poluidoras e boca-suja foram classificadas erroneamente. Entre esses erros de classificação, temos 30.6% das poluidoras e 26.2% das boca-suja que foram classificados incorretamente como não-spam. Além disso, avaliações boca-suja foram erroneamente classificados como poluidoras em 28.9% dos casos.

Resumindo os resultados da classificação, o valor do Micro-F1 é 76.9%, o que significa que o classificador está predizendo a classe correta para quase 77% das avaliações. Os valores do *F-measure* por classe são 87.4%, 78.0%, 61.5% e 50.9%, para as classes não-spam, comercial local, poluidora e boca-suja, respectivamente, resultando em um Macro-F1 igual a 69.4%.

		Classe predita			
		NS	***CL***	***PL***	***BS***
	NS	**95.0%**	0.7%	2.3%	2.0%
Classe	***CL***	6.1%	**71.4%**	21.1%	1.4%
verdadeira	***PL***	30.6%	4.9%	**58.0%**	6.5%
	BS	26.2%	1.9%	28.9%	**43.0%**

Figura 5.4: Classificação plana usando SVM

A Figura 5.5 mostra a matriz de confusão obtida como resultado de nossos experimentos para a abordagem de classificação plana usando o classificador RF. O *recall* das classes estão em negrito e indicam que 95.2% das avaliações não-spam, 77.1% das comerciais locais, 64.2% das poluidoras e 50% das bocas-sujas foram corretamente classificados pelo RF. Mais uma vez, apesar dos bons resultados obtidos para as classes comercial local e não-spam (*recall* > 75%), uma fração significativa das avaliações poluidoras e bocas-sujas foram classificadas erroneamente. No entanto, quando comparado com os resultados obtidos pelo classificador SVM, podemos observar que o classificador RF obteve os melhores valores de *recall* para todas as classes avaliadas. Além disso, em relação às métricas Micro-F1 e Macro-F1, o RF também superou o classificador SVM, Alcançou um Micro-F1 = 80.1% e Macro-F1 = 73.8%, este último calculado a partir dos seguintes valores de *F-measure* por classe: 89.5% para não-spam, 82.4% para comercial local, 66.7% para os poluidora e 56.8% para boca-suja.

		Classe predita			
		NS	***CL***	***PL***	***BS***
	NS	**95.2%**	0.5%	2.6%	1.7%
Classe	***CL***	5.1%	**77.1%**	16.9%	0.9%
verdadeira	***PL***	23.9%	4.4%	**64.2%**	7.5%
	BS	20.4%	1.4%	28.2%	**50.0%**

Figura 5.5: Classificação plana usando Random Forest

5.4 Classificação Hierárquica

Devido ao fato do classificador RF ter obtido os melhores resultados na classificação plana, decidimos avaliar a abordagem de classificação hierárquica utilizando apenas este classificador.

Como explicamos no começo deste capítulo, neste trabalho estamos considerando um classificador local por nó pai na abordagem hierárquica. Isso significa que, para o nosso problema, dois classificadores são treinados. O primeiro é usado para separar as classes do primeiro nível de hierarquia (não-spam e spam). E é treinado a partir dos dados de treinamento, que contém avaliações rotuladas como não-spam ou spam. O segundo classificador é construído para distinguir entre as classes comercial local, poluidora e boca-suja (classes filhas da classe spam). Desta forma, este é treinado a partir da mesma base de dados de treinamento utilizada pelo primeiro classificador, excluindo as avaliações não-spam e detalhando os rótulos das avaliações spam em comercial local, poluidora e boca-suja.

A Figura 5.6 mostra a matriz de confusão obtida como resultado da primeira etapa da classificação hierárquica.

		Classe predita	
		NS	***S***
Classe	***NS***	**93.1%**	6.9%
verdadeira	***S***	13.7%	**86.3%**

Figura 5.6: Classificação hierárquica usando Random Forest (primeira etapa)

A Figura 5.7 mostra a matriz de confusão obtida como resultado da segunda etapa da classificação hierárquica.

O resultado final destas duas fases de classificação são agregados na matriz de con-

		Classe predita		
		CL	***PL***	***BS***
Classe verdadeira	***CL***	**80.5%**	18.2%	1.3%
	PL	5.1%	**85.3%**	9.6%
	BS	1.5%	37.5%	**61.0%**

Figura 5.7: Classificação hierárquica usando Random Forest (segunda etapa)

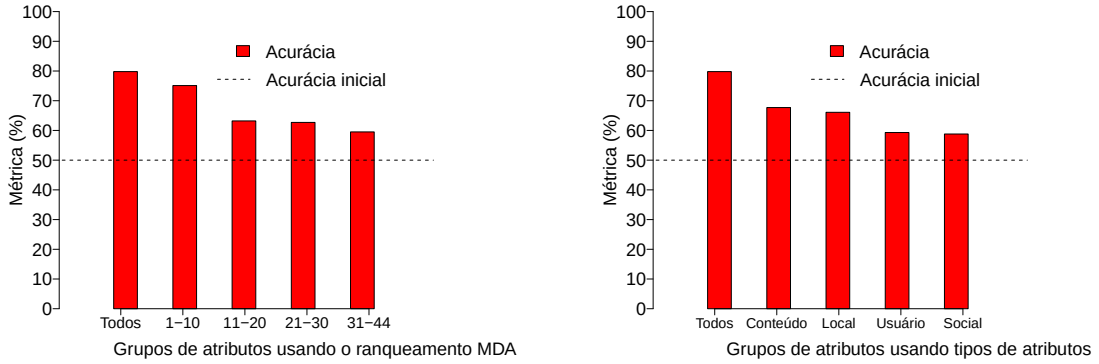
fusão apresentada na Figura 5.8. Mais uma vez, o *recall* das classes estão em negrito. Quando comparamos esses resultados com os melhores resultados de classificação plana apresentados na Figura 5.5, podemos verificar que, em relação ao *recall*, apesar de um resultado um pouco pior para a classe não-spam, a abordagem hierárquica proporcionou melhores resultados para a classe poluidora e boca-suja, que foram as classes com o pior desempenho em todos os experimentos. Além disso, analisando as métricas globais de classificação, enquanto que os melhores resultados para a abordagem de classificação plana alcançou Micro-F1 = 80.1% e Macro-F1 = 73.8%, a abordagem de classificação hierárquica obteve Micro-F1 = 80.4% e Macro-F1 = 74.6% (calculada a partir dos seguintes valores de *F-measure* por classe: 90% para a classe não-spam, 82.8% para a comercial local, 67.5% para a poluidora e 58.1% para boca-suja). Dessa forma, a abordagem de classificação hierárquica alcançado desempenho um pouco melhor do que a abordagem de classificação plana.

		Classe predita			
		NS	***CL***	***PL***	***BS***
	NS	**93.1%**	0.6%	4.2%	2.1%
Classe	***CL***	4.3%	**77.1%**	17.5%	1.1%
verdadeira	***PL***	19.7%	4.1%	**68.4%**	7.8%
	BS	13.1%	1.3%	32.8%	**52.8%**

Figura 5.8: Resultados finais da classificação hierárquica

5.5 O Impacto de Reduzir o Conjunto de Atributos

Na Seção 4.5, avaliamos o poder relativo dos atributos considerados em nossa base de dados em discriminar cada classe de avaliação das outras classes. No entanto, tão importante quanto entender a relevância desses atributos, é avaliar se um desempenho



(a) Ranqueamento MDA

(b) Ranquenamento por tipo de atributo

Figura 5.9: Desempenho do classificador com subconjuntos de atributos

competitivo de classificação ainda pode ser alcançado quando temos menos atributos ou apenas um dos diferentes tipos de atributos (conteúdo, usuário, local ou social). Este tipo de análise é importante pelas seguintes razões. Em primeiro lugar, uma vez que é esperado que *spammers* evoluam e adaptem suas estratégias para enganar sistemas anti-spam, no decorrer do tempo, alguns atributos podem se tornar menos importante, enquanto outros podem ganhar importância. Em segundo lugar, dado a enorme dimensão de base de dados relacionadas a redes sociais, alcançar resultados precisos de classificação com a redução da base de dados é algo desejável para acelerar o processo de classificação e melhorar o modelo de interpretabilidade.

A fim de avaliar o desempenho do classificador considerando diferentes subconjuntos de atributos, realizamos experimentos utilizando subconjuntos de 10 atributos que ocupam posições contíguas no ranqueamento MDA (ou seja, os primeiros 10 melhores atributos, os próximos 10 atributos e assim por diante) apresentados na Tabela 4.1. A Figura 5.9(a) mostra o valor da métrica acurácia (ou Micro-F1) quando utilizamos todos os atributos, quando utilizamos diferentes subconjuntos de atributos e quando utilizamos um classificador inicial, que considera todas as avaliações como não-spam (isto é, quando não utilizamos um classificador treinado). Também realizamos experimentos usando subconjuntos de acordo com cada tipo de atributo (conteúdo, usuário, local e social) da base de dados. A Figura 5.9(b) mostra os resultados obtidos desses experimentos.

Como pode ser observado na Figura 5.9, nossa classificação proporciona ganhos em relação à acurácia inicial em todos os subconjuntos de atributos avaliados, isto é,

mesmo atributos mal ranqueados têm algum poder discriminativo. Além disso, melhorias significativas em relação à acurácia inicial podem ser alcançadas mesmo quando apenas um tipo de atributo (por exemplo, atributos sociais) considerado em nossos experimentos pode ser obtido.

Capítulo 6

Conclusão e Trabalhos Futuros

Neste trabalho, abordamos o problema de detectar diferentes tipos de spam em avaliações de uma popular LBSN brasileira. Através de uma base de dados cedida pelo Apontador, contendo avaliações spam e não-spam rotuladas manualmente, fizemos uma categorização das avaliações spam em três diferentes classes: comercial local, boca-suja e poluidora. A fim de identificar características capazes de distinguir essas classes, coletamos o site do Apontador e reunimos informações de locais, usuários e do grafo social deles com mais de 137.000 usuários. Em seguida, utilizando nossa base de dados rotulada, investigamos as características e comportamentos de diferentes tipos de usuários que estão postando avaliação spam e além disso, fizemos uma caracterização das avaliações, revelando diversos aspectos comportamentais, tanto dos usuários da LBSN quanto do conteúdo das avaliações, capazes de diferenciar as classes de avaliação.

Submetemos nossas descobertas à uma técnica de classificação supervisionada utilizando aprendizagem de máquina, que foi capaz de distinguir de forma eficaz avaliações não-spam, comerciais locais, bocas-sujas e poluidoras. Particularmente, a nossa abordagem de classificação plana foi capaz de detectar corretamente 77% das avaliações comerciais locais, 64% das poluidoras, 50% das bocas-sujas, classificando erroneamente apenas cerca de 5% das avaliações não-spam. Portanto, a nossa abordagem representa uma alternativa promissora ao invés de simplesmente considerar todas as avaliações como não-spam ou selecionar aleatoriamente avaliações para inspeção manual. Também investigamos uma versão hierárquica da nossa abordagem, que proporcionou resultados ainda melhores para identificar as classes poluidora e boca-suja, que foram as classes com o pior desempenho em todos os experimentos. Finalmente, nossos resultados experimentais mostraram que, mesmo com um pequeno subconjunto de atributos (contendo

10 atributos), a nossa abordagem de classificação foi capaz de atingir uma acurácia alta (75%). Mesmo quando usamos apenas um dos tipos de atributos, como por exemplo atributos de conteúdo, nossa classificação produz benefícios significativos, com acurácia de aproximadamente 68%.

Esperamos que a identificação, caracterização e diferenciação de classes de spam em LBSNs apresentadas aqui possam também ter implicações para outros sistemas baseados em avaliação e também possam ser combinadas com outras estratégias de defesa. Como exemplo, notamos que avaliações bocas-sujas são postadas em locais com notas baixas. Assim, após a detecção de avaliações bocas-sujas, pode-se tentar diferenciar se elas são avaliações verdadeiras de usuários que não gostaram mesmo do local ou se elas estão relacionadas a um ataque maliciosamente combinado contra à classificação (nota) do local. Isto poderia ser feito por meio de um mecanismo de defesa de classificação, como o Iolaus, proposto por Molavi *et al.* [15].

Outra possibilidade importante que a nossa abordagem revela está relacionada com as avaliações comerciais locais. Notamos que os comerciantes locais são usuários ativos que cadastram locais, e portanto, contribuem positivamente para o sistema em alguns aspectos. Ao identificá-los, o sistema poderia oferecer-lhes um contrato de publicidade para seus serviços em certas áreas do site como "avaliações patrocinadas" ao invés de simplesmente remover suas avaliações do local ou até mesmo de expulsá-los do sistema.

Como trabalho futuro, pretendemos explorar a possibilidade de generalização do processo de detecção de spam para outras redes sociais. Por exemplo, na Figura 5.9(b), podemos observar que quando utilizamos apenas os atributos de conteúdo, a nossa classificação ainda atinge uma acurácia alta (aproximadamente 70%). E atributos de conteúdo são atributos que podem ser encontrados em qualquer rede social. Sendo assim, uma das etapas para a generalização do processo seria identificar atributos com potencial discriminativo que possam ser generalizados.

Como contribuição final, planejamos deixar nossa base de dados de avaliações spam disponível para a comunidade acadêmica em breve.

Referências Bibliográficas

[1] Miltiadis Allamanis, Salvatore Scellato, and Cecilia Mascolo. Evolution of a location-based online social network: Analysis and models. In *ACM Int'l Conference on Internet Measurement (IMC)*, pages 145–158, 2010.

[2] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999.

[3] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida. Detecting spammers on twitter. In *7th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS)*, 2010.

[4] F. Benevenuto, T. Rodrigues, V. Almeida, J. Almeida, and M. Gonçalves. Detecting spammers and content promoters in online video social networks. In *Int'l ACM Conference on Research and Development in Information Retrieval (SIGIR)*, pages 620–627, 2009.

[5] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[6] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117, 1998.

[7] C. Castillo, D. Donato, A. Gionis, V. Murdock, and F. Silvestri. Know your neighbors: Web spam detection using the web topology. In *Int'l ACM Conference on Research and Development in Information Retrieval (SIGIR)*, pages 423–430, 2007.

[8] comScore. Nearly 1 in 5 smartphone owners access check-in services via their mobile device, `http://bit.ly/mgaCIG`, Acessado em janeiro de 2013.

[9] Helen Costa, Fabricio Benevenuto, and Luiz Henrique de Campos Merschmann. Detecting tip spam in location-based social networks. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing (SAC)*, pages 724–729, 2013.

[10] Hongyu Gao, Jun Hu, Christo Wilson, Zhichun Li, Yan Chen, and Ben Y. Zhao. Detecting and characterizing social spam campaigns. In *ACM Int'l Conference on Internet Measurement (IMC)*, pages 35–47, 2010.

[11] Saptarshi Ghosh, Bimal Viswanath, Farshad Kooti, Naveen Kumar Sharma, Korlam Gautam, Fabricio Benevenuto, Niloy Ganguly, and Krishna Gummadi. Understanding and Combating Link Farming in the Twitter Social Network. In *Int'l World Wide Web Conference (WWW'12)*, pages 61–70, 2012.

[12] Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2000.

[13] P. Heymann, G. Koutrika, and H. Garcia-Molina. Fighting spam on social web sites: A survey of approaches and future challenges. *IEEE Internet Computing*, 11:36–45, 2007.

[14] N. Jindal and B. Liu. Opinion spam and analysis. In *ACM International Conference of Web Search and Data Mining (WSDM)*, pages 219–230, 2008.

[15] Arash Molavi Kakhki, Chloe Kliman-Silver, and Alan Mislove. Iolaus: Securing Online Content Rating Systems. In *Int'l World Wide Web Conference (WWW'13)*, pages 919–930, 2013.

[16] Kyumin Lee, James Caverlee, and Steve Webb. Uncovering social spammers: social honeypots + machine learning. In *ACM Int'l Conference on Research and Development in Information Retrieval (SIGIR)*, pages 435–442, 2010.

[17] Y. Lin, H. Sundaram, Y. Chi, J. Tatemura, and B. Tseng. Detecting splogs via temporal dynamics using self-similarity analysis. *ACM Transactions on the Web (TWeb)*, 2(1):1–35, 2008.

[18] A. Noulas, S. Scellato, C. Mascolo, and M. Pontil. An empirical study of geographic user activity patterns in foursquare. In *Int'l AAAI Conference on Weblogs and Social Media (ICWSM)*, 2011.

[19] Salvatore Scellato, Cecilia Mascolo, Mirco Musolesi, and Vito Latora. Distance matters: geo-social metrics for online social networks. In *ACM SIGCOMM Workshop on Online social networks (WOSN)*, pages 8–8, 2010.

[20] Salvatore Scellato, Anastasios Noulas, Renaud Lambiotte, and Cecilia Mascolo. Socio-spatial properties of online location-based social networks. In *Int'l AAAI Conference on Weblogs and Social Media (ICWSM)*, 2011.

[21] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research (JMLR)*, 6:1453–1484, 2005.

[22] M. Vasconcelos, S. Ricci, J. Almeida, F. Benevenuto, and V. Almeida. Tips, dones and to-dos: Uncovering user profiles in foursquare. In *ACM Int'l Conference of Web Search and Data Mining (WSDM)*, pages 653–662, 2012.

[23] I. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[24] C. Wu, K. Cheng, Q. Zhu, and Y. Wu. Using visual features for anti-spam filtering. In *IEEE Int'l Conference on Image Processing (ICIP)*, pages 509–512, 2005.

[25] Xiaojin Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.