Universidade Federal de Pernambuco
Centro de Informática

Graduação em Ciência da Computação

Análise de Sentimentos e Mineração de Links em uma Rede de Co-ocorrência de Hashtags

Arthur Cavalcanti Alem

TRABALHO DE GRADUAÇÃO

Recife
15 de março de 2013

Universidade Federal de Pernambuco
Centro de Informática

Arthur Cavalcanti Alem

Análise de Sentimentos e Mineração de Links em uma Rede de Co-ocorrência de Hashtags

Trabalho apresentado ao Programa de Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: *Ricardo Bastos Cavalcante Prudêncio*

Recife
15 de março de 2013

Dedico este trabalho aos meus pais, Sílvia e Topaulo, que sempre me apoiaram.

Agradecimentos

Quero agradecer primeiramente aos meus pais, Sílvia e Topaulo, pelo apoio, suporte, amor e carinho que sempre me deram.

Aos meus amigos, Rafael, Vinícius e Matheus, com os quais passei inúmeros fins de semana no CIn desenvolvendo projetos.

Ao Time UFPE da maratona de programação e seus integrantes, com os quais tive a chance de fazer parte de um projeto enriquecedor que me abriu muitas oportunidades.

À minha amiga e namorada, Adriana, que compartilhou junto comigo todas as experiências durante a graduação, incluindo as duas citadas acima.

Aos professores do Centro, por eu ter chegado aonde estou, em especial ao meu orientador Ricardo, por sua boa vontade e disponibilidade em me ajudar, e a coach Liliane, por sua dedicação e garra com o projeto da maratona de programação.

Resumo

O Twitter, assim como blogs e outras redes sociais, representa atualmente uma fonte quase inesgotável de informações, devido a milhões de usuários expressando livremente as suas opiniões. Para uso em aplicações, sejam financeiras, políticas ou acadêmicas, é altamente desejável a capacidade de se extrair estas opiniões, de forma a se obter um resumo da percepção pública com relação a tópicos específicos, sejam estes produtos, empresas, celebridades ou até mesmo partidos políticos.

Para isto, técnicas de análise de sentimento tradicionais, como a classificação de texto, vêm sendo amplamente empregadas, com relativo sucesso. No entanto, esta classificação automática pode ser consideravelmente aprimorada ao se fazer uso da estrutura de grafo que envolve os dados em questão. Neste contexto, este trabalho tem como objetivo avaliar os conceitos de Mineração de Opiniões e Assortatividade sendo aplicados em uma rede real montada a partir da co-ocorrência de *hashtags* em mensagens no Twitter.

Palavras-chave: Classificação Coletiva, Mineração de Links, Hashtag, Twitter, Análise de Sentimentos.

Abstract

Twitter, as well as blogs and other social networks, portray an almost endless source of information, due to millions of users freely expressing their opinions every day. The ability to extract these opinions is highly desirable, whether it be for use in financial, political or academic applications, as it can provide the means to digest the general public perception regarding specific topics, such as products, companies, celebrities and even political parties.

To this end, traditional sentiment analysis techniques, such as text classification, have been widely deployed, obtaining relatively good results. However, this automatic classification can be considerably improved by taking into consideration the graph-based structure of the underlying data. In this context, this project aims to evaluate the concepts of Opinion Mining and Assortativity when applied to a real network consisting of *hashtags* and their respective co-occurrences, as observed in tweets collected from Twitter.

Keywords: Collective Classification, Link Mining, Hashtag, Twitter, Sentiment Analysis.

Sumário

Lista de Figuras

Lista de Tabelas

CAPÍTULO 1

Introdução

Em plataformas como o Twitter, os usuários tendem a expressar livremente e publicamente os seus sentimentos, o que cria um meio ideal de se capturar as opiniões comuns sobre diversos tópicos, como empresas, produtos ou celebridades. Esta informação, no entanto, acaba por ser muito extensa, inviabilizando que se analise na íntegra todas as opiniões expressadas, e tornando atrativo o uso de ferramentas automáticas capazes de extrair o sentimento geral contido nos dados.

Estas ferramentas, contudo, tradicionalmente consideram as entidades analisadas de forma individual. Neste caso, o alvo das opiniões costuma ser avaliado com base exclusivamente em suas próprias características, sem assimilar informações relativas a outras entidades que já foram ou também serão analisadas, e que possam ter relação com ela. Por exemplo, para se classificar o gênero de um novo livro, uma técnica padrão de aprendizagem de máquina iria percorrer apenas o texto do mesmo, avaliando quais palavras são encontradas. Porém, se levarmos em consideração quais os gêneros de outros livros escritos pelo mesmo ator, isto pode nos auxiliar na determinação do gênero do novo livro.

Isto se torna especialmente útil quando consideramos que já foi amplamente observado que várias redes, incluindo as redes sociais, tendem a favorecer desproporcionalmente a ligação entre nós que possuem atributos ou características similares, um conceito denominado assortatividade [5] [6]. Com base nisso, muitos algoritmos relacionados à Mineração de Links assumem que este conceito seja aplicável à rede em questão, na tentativa de se aproveitar disto para aprimorar os resultados obtidos.

Neste contexto, este trabalho tem como objetivo explorar o uso de hashtags no Twitter como uma nova abordagem para se classificar tópicos em relação à opinião dos usuários, uma vez que a análise baseada apenas em classificação de textos tende a não ser tão precisa por considerar as entidades como sendo independentes, não fazendo uso da informação de links entre as mesmas [1]. Utilizando técnicas de classificação coletiva, os tópicos serão classificados automaticamente como relacionados a percepções positivas ou negativas dos usuários do Twitter em um determinado período, baseado principalmente nos outros tópicos aos quais estão frequentemente atrelados, ao invés de apenas na análise de sentimento dos textos das próprias mensagens. São exploradas diversas estratégias para otimização dos resultados, incluindo a combinação com técnicas de classificação de texto, e a precisão obtida ao término deste trabalho se mostra significativamente superior à obtida através do uso exclusivo de algoritmos do estado-da-arte para classificação de texto pura, corroborando a idéia de que a assortatividade também está presente nesta rede de co-

ocorrência de *hashtags*, e que a informação da estrutura da rede e seus respectivos links possuem um grande potencial para a classificação automática de entidades.

1.1 Motivação

O uso de ferramentas de análises de sentimentos no Twitter tem crescido significativamente desde o início da rede social, alcançando diversos usos, incluindo políticos, econômicos e acadêmicos. Em 2012, durante o ano eleitoral americano, foi utilizada a análise de sentimentos para acompanhar o apoio estimado com relação aos partidos políticos democrata e republicano. Em especial, foi comparado o resultado obtido após os debates presidenciais, com as pesquisas feitas por telefone, e em todos os três debates os resultados obtidos automaticamente através do acompanhamento pelo Twitter foram precisamente similares aos colhidos pelas pesquisas. Isto aponta para um futuro onde não haverá mais a necessidade de pesquisas por telefone, limitadas a uma pequena amostragem, quando comparada à internet onde uma quantidade imensamente maior de indivíduos expõem deliberadamente as suas opiniões acerca de todos os tópicos.

Dois anos antes, outro estudo [11] fez uso da análise de sentimentos com propósito financeiro, também com base em informações postadas livremente, como no Twitter e outras redes. Foi desenvolvida uma ferramenta para a empresa Dow Jones, onde através do sentimento expressado pelo público, era possível detectar precocemente tendências emergentes com relação a empresas ou produtos que fossem impactar as suas ações na bolsa, de forma que a Dow Jones podia então atuar a tempo de garantir um lucro máximo nas operações, manuseando as ações de acordo com a tendência emergente, negativa ou positiva.

Em paralelo a isto, o uso da Classificação Coletiva vem recentemente também crescendo muito, e demonstrando ótimos resultados em diversos estudos, tendo inclusive já sido aplicado a *hashtags* no Twitter com propósitos semelhantes ao deste trabalho, como em [1] e [8]. Em cenários onde há uma estrutura de grafo subjacente aos dados, a Classificação Coletiva provê uma vantagem considerável quando comparada a técnicas tradicionais que avaliam as entidades de forma "livre de contexto", independente das outras.

Portanto, uma combinação de ambas Análise de Sentimentos e Classificação Coletiva parece nada menos do que uma evolução natural do direcionamento de estudos com base no Twitter e outras redes sociais abertas, prometendo ótimo desempenho e uma gama enorme de aplicações possíveis.

1.2 Estrutura do Documento

Esta monografia está dividida em sete capítulos. O Capítulo 2 apresenta uma breve introdução à Análise de Sentimentos, assim como os pontos fracos das técnicas tradicionais, que levaram à exploração da Mineração de Links. O Capítulo 3 introduz a Classificação Coletiva, citando trabalhos relacionados, e detalhando a divisão dos algoritmos feita pela ferramenta utilizada. O Capítulo 4 explica o processo de montagem da rede, desde o procedimento de coleta dos dados, até a formação dos grafos utilizados para os testes.

No Capítulo 5 é detalhado as abordagens para a classificação inicial do conjunto de *hashtags*, assim como as opções para os cálculos de estimativas necessárias. No Capítulo 6, são apresentados e comparados os resultados obtidos, de forma agregada e individual, assim como uma análise mais detalhada dos mesmos. Por último, no Capítulo 7, são feitas as conclusões acerca dos resultados obtidos, limitações do sistema e possíveis trabalhos futuros.

CAPÍTULO 2

Análise de Sentimentos

Mineração de Opiniões, ou Análise de Sentimentos, se trata da utilização de técnicas para identificar e extrair informações subjetivas contidas em um dado material. Geralmente estas técnicas envolvem algoritmos de processamento de linguagem natural e mineração de texto, para determinar, por exemplo, se comentários acerca de um determinado produto têm um caráter positivo ou negativo – mas podem também ir mais além, detalhando um sentimento como sendo de raiva, tristeza ou decepção, ao invés de apenas "negativo". Esta análise pode ainda ser realizada em diversos níveis de granularidade, desde documentos inteiros, até cada sentença individual.

Na tentativa de identificar a orientação da opinião de usuários, várias técnicas de Análise de Sentimento já foram propostas, a maioria delas baseadas em classificação de textos [3]. Neste caso, é comum se utilizar algum conjunto de palavras opinativas previamente classificadas, para auxiliar a determinação da polaridade da informação subjetiva sendo avaliada. Porém, isto resulta em algumas desvantagens, como principalmente o custo de se montar tal lista de palavras de forma manual, ou o ruído de se tentar gerá-la de forma automática. Uma terceira opção é utilizar uma lista já existente, porém palavras subjetivas podem representar sentimentos totalmente opostos em diferentes contextos ou domínios, como um monitor *grande* e uma câmera fotográfica *grande*, ou ainda um refrigerante *gelado* e uma pizza *gelada*. Para contribuir com a lista de problemas, há ainda a dificuldade em lidar com casos de sarcasmo e ironia, que são especialmente difíceis de tratar em texto, como explorado em [13].

Como exemplo de uso relevante e comercial, em [11] os autores abordam o uso de ferramentas de análises de sentimentos no Twitter para auxiliar no mercado americano de ações. Foi desenvolvida uma ferramenta para a companhia Dow Jones, com o objetivo de detectar tendências emergentes significativas com relação a produtos e companhias específicas através da análise dos *tweets* sendo postados em tempo real, ou seja, basicamente tentar prever a subida ou queda de ações através de mensagens na rede social, de forma a antecipar as movimentações de transações e obter o máximo de lucro possível em cima disto.

Outro cenário popular é a cobertura na mídia sobre o sentimento geral com relação a tópicos populares ou polêmicos, como as eleições presidenciais. Nas eleições americanas de 2012, ferramentas de análises de sentimento foram utilizadas para prever, com alta precisão, as variações no suporte a Romney ou Obama como consequência dos debates presidenciais, utilizando para isto as conversações em mídias sociais como o Twitter, durante e após os debates. Isto é apenas mais um exemplo de como a opinião emitida nas

redes sociais pode representar um sinalizador relevante, do qual se podem extrair informações valiosas com o devido monitoramento e análise.

Estudos sugerem, no entanto, que a classificação automática de dados pode ser consideravelmente aprimorada ao se considerar a vizinhança destes em uma estrutura de grafo [4], como é observado na Classificação Coletiva. Isto acontece porque muitos conjuntos de dados hoje são mais bem representados como uma coleção de objetos interligados entre si, como é o caso de redes sociais, onde o atributo amizade pode representar um link entre duas pessoas distintas. Mineração de Links diz respeito a técnicas de mineração de dados que explicitamente levam em conta estas ligações ao construir modelos descritivos ou preditivos destes dados, e abrange tarefas como detecção de grupos, predição de links, e classificação coletiva, entre outras [2].

Um dos pontos fracos das abordagens de análises de sentimentos tradicionais se dá justamente por esta limitação de considerar as instâncias de forma totalmente independente, sem levar em conta outras instâncias às quais ela está conectada, ou até mesmo as que estão presentes no mesmo contexto. Por exemplo, para se classificar o gênero de um livro, pode ser muito útil considerar quais os gêneros dos outros livros escritos pelo mesmo autor – informação que é totalmente desprezada pelos algoritmos de aprendizagem de máquina que consideram cada instância isolada. Da mesma forma, para se classificar o tópico de uma página web, as páginas às quais ela está ligada através dos hyperlinks podem fornecer informações relevantes.

Isto não significa que as técnicas tradicionais de mineração de opiniões estejam obsoletas – em muitos contextos simplesmente não é aplicável ou relevante a informação inter-instâncias. Além disso, como veremos mais adiante, muitas destas técnicas acabam na verdade sendo incorporadas como partes das estratégias baseadas em mineração de links, então a diferença é que agora, ao invés de serem consideradas de forma isolada, elas são apenas um módulo de um sistema ainda mais potente.

CAPÍTULO 3

Classificação Coletiva

Considerando que estamos lidando com uma rede baseada quase exclusivamente nos links entre as entidades, a Classificação Coletiva passa a ser uma abordagem muito mais apropriada do que a aprendizagem de máquina tradicional, que considera as instâncias de forma independente.

A Classificação Coletiva vem ganhando destaque em pesquisas mais recentes, porém já foi estreada nos mais diversos ramos, desde áreas de investigação criminal, como detecção de fraudes e contraterrorismo, até mesmo reconstrução de imagens e rotulação de patentes, documentos, e filmes [2]. Os modelos mais comuns costumam explorar conceitos já bem observados, como o de assortatividade, onde nós de uma rede tendem a possuir desproporcionalmente mais conexões com aqueles nós com os quais compartilham características em comum. Frequentemente fazem uso também da suposição de Campos Aleatórios de Markov de primeiro grau, no qual se assume que cada entidade é influenciada apenas pelos seus vizinhos diretos (ou seja, de primeiro grau), como forma de simplificar os modelos.

O propósito central é aproveitar informações que não pertencem diretamente à entidade sendo classificada, mas sim a entidades às quais a mesma está conectada, uma vez que se assume que entidades similares se conectam com maior frequência na rede em questão. Caso haja uma quantidade considerável de vizinhos conhecidos, a informação extraída a partir deles pode compensar a falta de informação conhecida sobre o nó original. Isto acaba gerando complicações adicionais quando surgem dependências cíclicas, onde para classificar um nó, é necessário conhecer a classe de um segundo nó, cuja classificação também está aguardando o resultado do primeiro. Para resolver problemas desta natureza, vários algoritmos já foram propostos, como serão apresentados na subseção 3.2.3.

3.1 Trabalhos Relacionados

Como mostrado em [2], a Classificação Coletiva é um ramo não tão explorado dentre as técnicas de mineração de links, mas que vem recebendo mais atenção recentemente.

Em [8], os autores exploram o uso da mineração de links para aprimorar e aperfeiçoar a geração de sumários de opiniões, também com base em *hashtags* no Twitter. O objetivo é ir além da separação de *hashtags* em apenas três tipos (tópico, sentimento e misto), e tentar extrair automaticamente ao que aquela *hashtag* se refere, qual a opinião geral observada a respeito, e por que, quais argumentos defendem aquela opinião. Ou seja,

ao invés de apenas se classificar a polaridade da *hashtag*, é também criado todo um resumo sobre ela.

Já em [3] e [9], o tema abordado é a predição de alinhamento político de usuários do Twitter, também com base na Classificação Coletiva. Os estudos mostram como é possível, com alta taxa de precisão, identificar a qual partido político americano um usuário provavelmente está atrelado, utilizando a informação de com quais outros usuários ele está relacionado, seja através de um caráter de seguido, ou de seguidor. A Classificação Coletiva demonstra grande avanço sobre as abordagens dependentes exclusivamente de análise de textos, e a aplicação se torna excepcionalmente atrativa em anos de eleições, onde tanto os próprios partidos, como também a mídia, desejam ser capazes de acompanhar em tempo real qualquer informação que ajude a estimar as tendências políticas e o impacto de eventos como um debate presidencial.

[1] abrange uma abordagem semelhante à deste trabalho, porém mais limitada, onde o autor também tem como objetivo principal avaliar a polaridade das *hashtags* no Twitter à partir da Classificação Coletiva, mas não explora tantas alternativas para aprimorar os resultados obtidos. O autor se limita a fazer uso do significado literal das *hashtags* como informação supervisionada a ser utilizada para melhorar o desempenho da classificação, sem a possibilidade de atribuir um peso maior aos resultados estimados pelo classificador de texto, e são utilizadas versões ultrapassadas de alguns algoritmos, contribuindo para um pior desempenho.

Estudos observados em [4] [6] dão maior ênfase à classificação em si, e não a uma aplicação ou cenário específico, sendo testado o desempenho das técnicas de classificação coletiva em diversas áreas, de classificação de páginas à partir dos hyperlinks para outras páginas, até a classificação de patentes e artigos com base em suas referências. Estes estudos mostram, contudo, que em todas as áreas exploradas, a utilização do conhecimento referente aos links entre entidades obtém maior desempenho quando apenas a própria classificação dos vizinhos é levada em conta, e não as outras propriedades associadas ao mesmo. Utilizar os termos presentes nos vizinhos acaba sendo prejudicial para a classificação, o que no nosso cenário de rede de co-ocorrência de *hashtags*, seria o equivalente a se considerar os termos contidos em *tweets* que possuem uma *hashtag* que em algum momento já apareceu em conjunto com a *hashtag* que se está avaliando, independente de esta estar presente no *tweet* em questão. Os autores chegam às conclusões de que tal informação acaba por representar um nível elevado de ruído para o nó original, não sendo muito relevantes na classificação do mesmo, o que causa o prejuízo no desempenho.

Por fim, [7] apresenta um estudo no qual os algoritmos originais concebidos para classificação coletiva são “desmembrados” em múltiplos módulos, os quais podem ser recombinados de forma quase arbitrária, efetivamente gerando novos algoritmos. Como resultado do estudo, é provida a ferramenta que foi desenvolvida, a qual é utilizada neste

trabalho, e apresentada na seção a seguir. Além disso os autores também realizam um estudo de caso, onde apresentam os resultados de testes sistemáticos realizados com o auxílio da ferramenta, que permitem explorar os pontos fortes e fracos de cada módulo dos algoritmos. Propõem também alguns módulos próprios, baseados em modelos simples, mas que obtiveram bons resultados nos testes, e que sugerem serem utilizados como patamar inicial para comparação com algoritmos mais sofisticados, como uma medida de quão "fácil" é a classificação em uma determinada rede. Por último, é citado ainda um cenário que sugere que determinados nós podem possuir maior relevância na classificação da rede, de forma que estudos futuros possam ser considerados na área de se determinar quais nós seriam mais vantajosos de se descobrir a classificação real, em um critério de custo-benefício, para auxiliar na classificação do restante da rede.

3.2 Netkit

Para os algoritmos de classificação coletiva, utilizaremos a ferramenta Netkit[1], cujos autores [7] defendem a idéia de que, para uma comparação mais eficiente entre os algoritmos da literatura, é possível separá-los em 3 etapas principais: A classificação local, a relacional, e a coletiva propriamente dita, também chamada de "inferência coletiva". Os algoritmos originais foram concebidos como "pacotes" com as 3 etapas pré-definidas e agrupadas, porém é possível separá-las em módulos e recombinar ou substituir estes respectivos módulos, efetivamente criando novas abordagens. Aproveitar a possível sinergia entre módulos diferentes, portanto, permitiria a criação de algoritmos com desempenho possivelmente superior aos originais.

3.2.1 Classificação Local

Para se obter uma estimativa de a qual classe uma entidade pertence, o primeiro passo é observar os atributos locais da própria entidade. O classificador local então trabalhará em cima destes atributos, que podem ser desde simples valores numéricos, até um conjunto de textos, para determinar a probabilidade inicial de o nó pertencer a cada classe possível. No contexto de análise de sentimentos, o classificador local corresponde aos algoritmos tradicionais de aprendizagem de máquina [7], e geralmente o atributo em questão são textos associados à entidade sendo classificada, que no nosso caso são os *tweets* que contém a dada *hashtag*. Isto é, um classificador de texto que avalia uma dada *hashtag* com base nos *tweets* que a contém, nada mais é do que um classificador local neste contexto, atribuindo uma classe à *hashtag* sem levar em consideração as outras *hashtags* às quais ela está associada, ou qualquer outra informação pertinente ao grafo formado.

[1] http://netkit-srl.sourceforge.net/

Também é possível utilizar um classificador trivial, que simplesmente atribui a todos os nós a mesma probabilidade observada na distribuição das classes originalmente conhecidas, ou, em um caso ainda mais trivial, simplesmente não atribui nada. Para o primeiro caso, o "conjunto de treinamento" é utilizado para estimar diretamente a distribuição de probabilidade. Caso 60% dos nós conhecidos sejam de uma classe, 30% de uma segunda classe e 10% da terceira, todos os nós desconhecidos receberiam estas mesmas porcentagens como sendo a probabilidade de pertencerem a cada classe respectiva, o que faria com que uma quarta classe, não presente no conjunto inicialmente conhecido, nunca aparecesse nas classificações. Esta abordagem também só é relevante para os algoritmos coletivos que mantém as probabilidades incertas durante todo o processo, pois quando a classificação intermediária segue um modelo binário, como é o caso do Iterative Classification, apenas a maior probabilidade é considerada, o que significa que todos os nós seriam inicialmente estimados como pertencendo à mesma classe majoritária, resultando em uma baixa performance na classificação final.

Neste segundo caso, se possível, a abordagem mais adequada é simplesmente não atribuir nada aos nós desconhecidos, fazendo com que, na implementação do Netkit, estes sejam ignorados até terem uma probabilidade atribuída a eles pelos outros módulos de classificação. Contudo, nem todos os algoritmos suportam esta abordagem. Uma falha deste método, ainda, é que ele não seria capaz de classificar nós que estejam em um componente conexo separado, que não contenha nós conhecidos, daí a importância de se utilizar um classificador local apropriado para inicializar as estimativas de forma individual para cada nó.

3.2.2 Classificação Relacional

É nesta etapa que a Mineração de Links é utilizada, diferenciando as estratégias de uma simples aplicação direta dos algoritmos de aprendizagem de máquina tradicionais, que consideram cada nó de forma independente. Aqui serão considerados todos os vizinhos conhecidos de uma entidade, ou seja, aqueles nós que co-ocorrem com a *hashtag* sendo classificada, e suas características são utilizadas para refinar o cálculo das probabilidades do vértice em questão. Estudos mostram, no entanto, que considerar os próprios atributos locais dos vizinhos é na verdade prejudicial para a classificação [4] [6], chegando a de fato piorar a precisão da mesma, sendo uma melhor estratégia levar em conta apenas as classes (ou as respectivas probabilidades) destes vizinhos. Estes estudos foram inicialmente realizados com base na classificação de páginas web, relacionadas a outras páginas através de *hyperlinks*, onde considerar os termos presentes nas páginas vizinhas, equivalente a considerarmos o texto de *tweets* de *hashtags* vizinhas, não ajudava na classificação. O mesmo foi observado também para a classificação de documentos, como artigos de áreas diversas que referenciavam uns aos outros, e também para o cenário de patentes, onde as mesmas utilizavam outras patentes, que por sua vez possuíam conteúdo não necessariamente relacionado às patentes iniciais.

Há vários algoritmos capazes de realizar esta etapa, baseados nos mais diversos modelos matemáticos, desde Campos Gaussianos e Campos Aleatórios de Markov, até puramente no conceito de assortatividade, introduzido anteriormente. O Netkit, porém, não é capaz de realizar classificação multivariada, como alguns dos algoritmos originais foram concebidos, tendo portanto sido adaptados para o modelo univariado, onde apenas as classes dos vizinhos adjacentes e do nó atual são levadas em consideração.

3.2.3 Inferência Coletiva

Esta etapa, responsável pela classificação coletiva propriamente dita, é responsável por utilizar as duas anteriores para inferir as classificações dos nós desconhecidos, especialmente aqueles que estão apenas indiretamente conectados aos nós originalmente conhecidos. A idéia central é tentar maximizar a probabilidade conjunta de todos os nós pertencerem às respectivas classes estimadas, o que a priori requer a comparação entre uma quantidade exponencial de combinações: Cada um dos N nós pode pertencer a X classes, portanto existem X^N configurações finais possíveis. Existem algoritmos que resolvem este problema de maneira exata e de forma mais eficiente do que este pior caso, porém continuam tendo complexidade exponencial, não sendo viáveis para aplicações reais. Por exemplo, o algoritmo *junction-tree* [10] é capaz de fornecer soluções exatas para grafos arbitrários, no entanto possui complexidade exponencial ao maior clique no grafo, o que representa um problema nos casos de redes sociais, e também da rede de co-ocorrência de *hashtags*, que possuem cliques consideravelmente grandes.

Atualmente, utilizam-se principalmente algoritmos heurísticos, que estimam uma configuração próxima do ótimo, que mesmo não sendo a resposta ideal, ainda é satisfatória. Para tratar as dependências cíclicas, quando um nó depende da classe de um segundo nó desconhecido, cuja classe também depende da classificação do primeiro, estes algoritmos geralmente trabalham de forma iterativa, analisando todos os nós simultaneamente no tempo T_x para estimar seus valores no tempo T_{x+1}, repetindo este procedimento até atingir a convergência ou um limite de iterações. São nestas repetições que o classificador coletivo reaplica o classificador relacional, maximizando o benefício obtido [16], e para as quais é necessário uma estimativa inicial para se utilizar no tempo T_0. Em algoritmos como o Relaxation Labeling, o tempo é "congelado" durante cada repetição, para então todos os nós serem atualizados simultaneamente com suas novas probabilidades; Já em algoritmos como o Gibbs Sampling, os nós são ordenados de forma aleatória, e depois processados sequencialmente, sendo que os nós são imediatamente atualizados, o que significa que os vizinhos de um nó poderão estar defasados, alguns já atualizados para o tempo T_{x+1} enquanto os outros permanecem em T_x.

Há ainda um outro ramo heurístico, não baseado em iterações, mas sim nos algoritmos de distância mínima em um grafo – aqui, a heurística se dá no fato de se manter apenas um ou *K* melhores estados a cada passo da busca, uma vez que manter todos os estados também possuiria custo proibitivo. Nestas buscas, os nós são ordenados, e as

arestas possuem peso relacionado a uma fórmula logarítmica das probabilidades de pertencer a cada classe, de forma que a distância de um nó para o nó seguinte seja menor através da aresta que representa a escolha de classe que tem maior probabilidade para aquele nó sendo classificado.

Muitos dos algoritmos para classificação coletiva vieram na realidade de outros ramos, sendo mais recentemente adaptados para as mais diversas tarefas de classificações. Um exemplo disto foi o Relaxation Labeling, originalmente concebido para a reconstrução de imagens [2], "classificando" os *pixels* perdidos com base no restante da imagem, levando em conta as cores de cada pixel, e posteriormente detectar bordas de imagens, classificando pixels como pertencendo a bordas horizontais ou verticais em uma dada imagem.

CAPÍTULO 4

Montagem da Rede

A etapa inicial da implementação deste trabalho consistiu na coleta dos dados e montagem da rede. Como mencionado anteriormente, a rede será formada através da co-ocorrência de *hashtags* no Twitter, onde os vértices são as próprias *hashtags*, e os links representam duas *hashtags* que apareceram em um mesmo *tweet*.

4.1 Coleta dos Dados

Para a coleta dos dados, foi utilizada a API open-source Twitter4J[2], que permite a busca de mensagens (*tweets*) no Twitter. Esta API provê a habilidade de se refinar a busca a partir de diversos filtros, como palavras-chave, *hashtags*, localização, usuário, datas e etc. Foram inicialmente selecionadas 12 *hashtags* de tópicos populares, como "#Christmas", "#Apple" e "#Obama", para serem utilizados como pivôs para a busca. Após uma amostragem inicial, o conjunto de *hashtags* foi incrementado com 8 tópicos adicionais que apareciam em elevadíssima frequência associados a múltiplos pivôs.

O conjunto final de *hashtags* utilizadas como ponto de partida para a coleta dos *tweets* ficou composto pelos 20 termos a seguir: "Android", "Apple", "Christmas", "Egypt", "Facebook", "Google", "Gplus", "Greece", "Hobbit", "Holidays", "Instagram", "Ios", "Ipad", "Iphone", "Microsoft", "Obama", "Usa", "Windows", "Xfactor", "Youtube". Foram então coletados todos os *tweets* disponíveis em que pelo menos uma destas *hashtags* estivessem presentes, e que foram "twitados" no período entre 17/12/2012 e 24/12/2012, sem haver diferenciação entre letras maiúsculas ou minúsculas, ou seja, *hashtags* como "#Google" e "#google" foram consideradas como sendo iguais.

Isto totalizou 3,14 milhões de *tweets*, que passaram por alguns tratamentos. Primeiro, a API para buscas no Twitter possui limitações na quantidade de termos e regras utilizadas simultaneamente, o que limitou a busca a ser realizada com subconjuntos de 4 *hashtags* por vez. Isto abre a possibilidade de *tweets* repetidos serem obtidos através de "*seeds*" diferentes, como no caso de uma mensagem utilizar duas *hashtags* de conjuntos separados. Após uma análise inicial, foi constatado também que certas *hashtags* eram frutos de *spam* originados por aplicativos que geravam dezenas de milhares de *tweets* automaticamente. Um exemplo disto foi a *hashtag* "#gameinsight", que gerava mensagens

[2] http://twitter4j.org/

a partir de jogos cada vez que o usuário avançava em alguma etapa ou recebia alguma premiação no jogo, como neste *tweet*:

"*New 5 class "Curious neighbour" award received!*
24 #android #gameinsight #androidgames"

Por último, foram desconsideradas também *hashtags* que continham apenas caracteres de outro alfabeto, que acabavam sendo substituídos totalmente por interrogações. No caso, todas as *hashtags* compostas apenas por 4 caracteres chineses, japoneses, russos ou árabes, se transformavam na *hashtag* "#????", tornando-a inválida e aleatoriamente conectada a uma multidão de tópicos com alta frequência. Foram consideradas como válidas apenas as *hashtags* que possuíssem mais de 50% dos caracteres válidos. Após a remoção destas *hashtags* inválidas, dos *tweets* repetidos ou gerados por *spam*, e dos *tweets* que possuíssem apenas uma única *hashtag* (tornando-o irrelevante para uma rede de co-ocorrência), restaram 1,5 milhões de *tweets* para serem utilizados na montagem da rede – este corte, de mais de 50% em relação ao total coletado, teria sido ainda mais drástico caso a coleta inicial tivesse sido dada com base em *tweets* aleatórios, ao invés de pré-determinados por *hashtags* usadas como "*seeds*" iniciais, uma vez que a porcentagem de *tweets* aleatórios que contém ao menos uma *hashtag* é inferior a 20%, como observado em amostragem inicial e em pesquisas relacionadas [1] [8].

4.2 Grafos Resultantes

A partir dos *tweets* coletados e tratados, consideramos cada *hashtag* existente como sendo um nó, e cada co-ocorrência distinta de pares de *hashtags* como sendo uma aresta entre os mesmos. As co-ocorrências são definidas da seguinte forma: Dado que um *tweet* possua por exemplo 4 *hashtags* 'a', 'b', 'c' e 'd', todos os pares resultantes são considerados: 'ab', 'ac', 'ad', 'bc', 'bd', e 'cd'. As arestas não são direcionadas e possuem peso, determinado pela soma de ocorrências daquele par.

Isto resultou em um grafo composto por 270 mil nós e pouco mais de 3 milhões de arestas, com todos os vértices pertencendo ao mesmo componente conexo. Devido à intratabilidade de uma rede deste porte com os algoritmos existentes atualmente, foram realizados cortes para torna-la mais compacta e expressiva. Foram utilizados dois limiares (ou *thresholds*) distintos para o peso mínimo de uma aresta, e em cada caso, vértices que não possuem mais arestas são eliminados. Considerando apenas arestas com peso maior ou igual a 10, obtemos um grafo composto por 20 mil nós e 218 mil arestas; Considerando as arestas com peso maior ou igual a 50, o grafo resultante possui 5 mil nós e 25 mil arestas. Em ambos os casos a rede permaneceu conexa, o que ocorre principalmente devido a *hashtags* de uso abundante, como "#twitter", e/ou genérico, como "#lol", que se conectam diretamente a todas as 20 *hashtags* iniciais, fazendo com que apenas cortes muito severos sejam capazes de dividir o grafo em múltiplos componentes não triviais. Estes grafos são

apresentados abaixo. As visualizações foram geradas a partir da ferramenta Gephi[3], capaz de gerar imagens de grafos sob diferentes *layouts*, e calcular informações estatísticas como a intermediação dos vértices e a modularidade do grafo, além de possuir algoritmos como o de detecção de comunidades em um grafo.

Nas figuras abaixo, as cores representam "comunidades" diferentes, determinadas pelo algoritmo de detecção de comunidades do Gephi, que se baseia apenas na ligação entre os nós e seus respectivos pesos. Já os tamanhos dos vértices foram determinados com base na centralidade de intermediação, onde vértices maiores possuem maior intermediação.

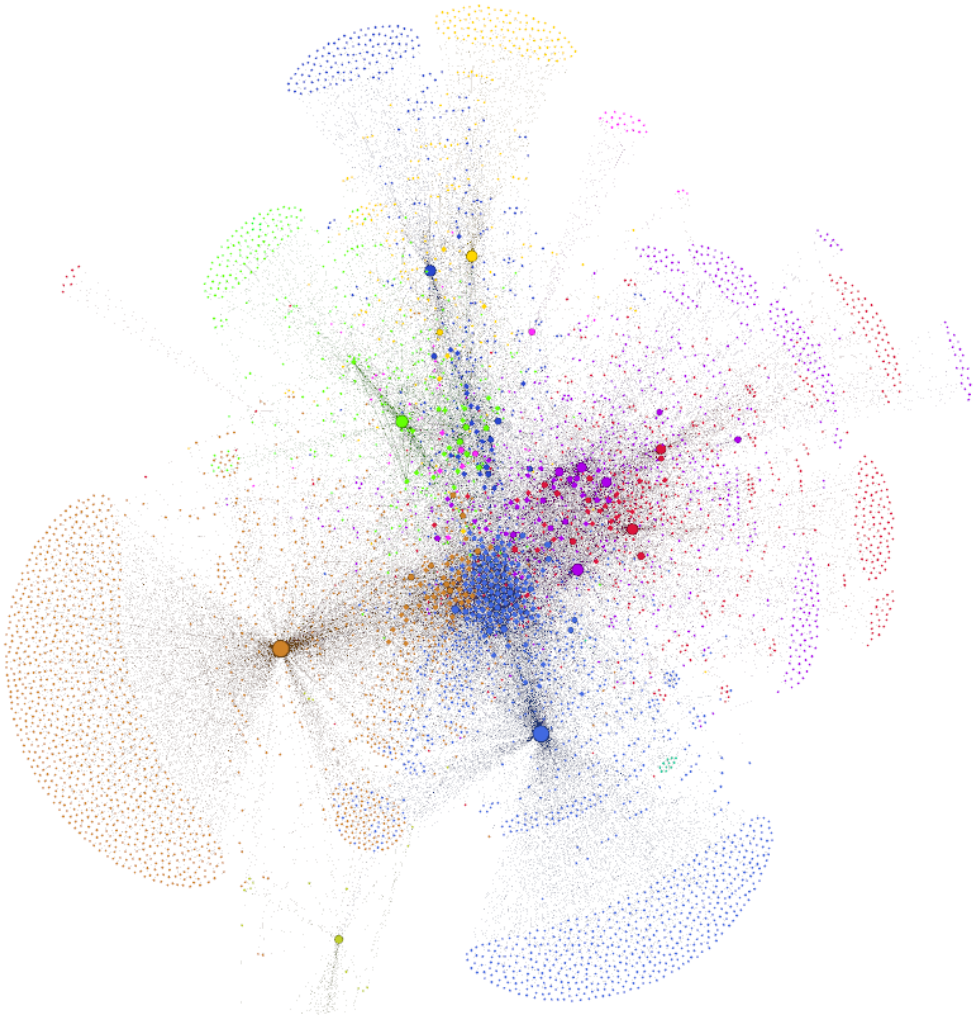

Figura 4.1 Grafo composto por 5097 nós, visualizado através da ferramenta Gephi. As cores representam "comunidades" diferentes, e o tamanho dos vértices é proporcional à intermediação dos mesmos. As duas maiores comunidades são a laranja, centrada em torno da *hashtag* "#Christmas", e a azul clara, correspondente a "#Instagram".

[3] http://gephi.org/

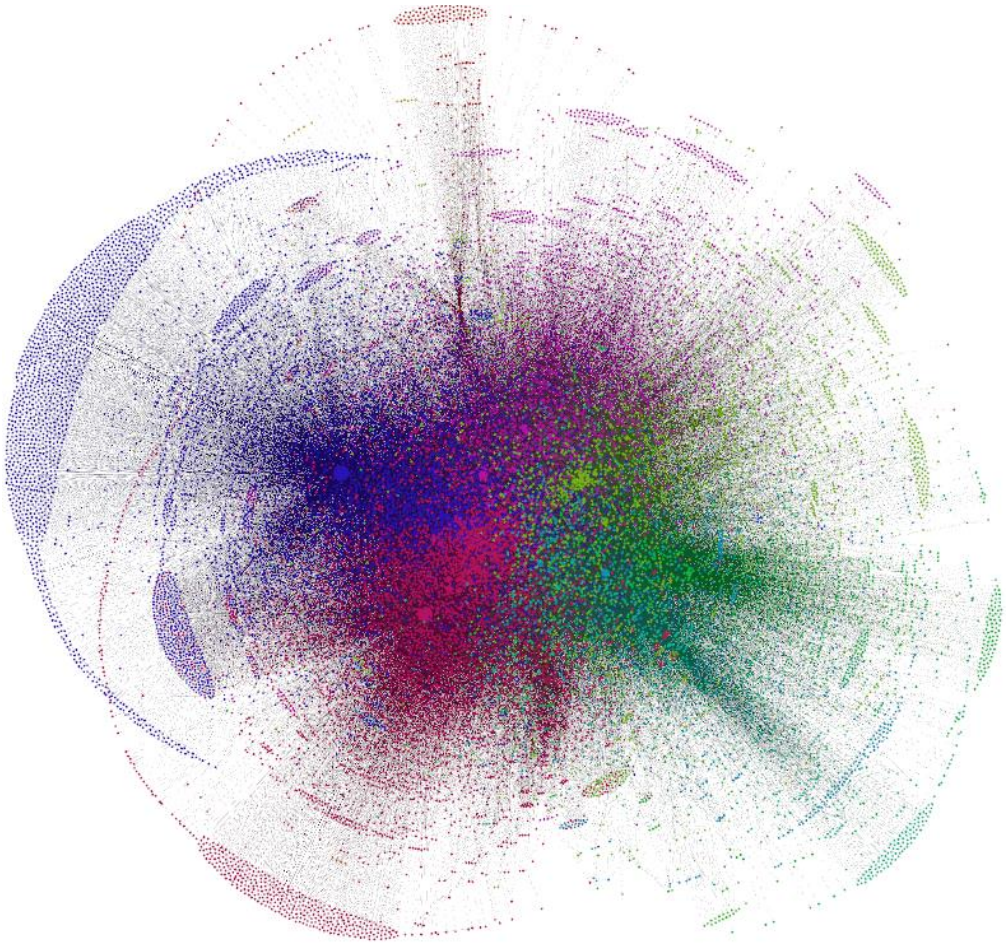

Figura 4.2 Grafo composto por 20560 nós, visualizado através da ferramenta Gephi. Nesta imagem, "#Christmas" corresponde à cor azul, e "#Instagram" ao vermelho.

CAPÍTULO 5

Classificação Inicial das Hashtags

Para executar os testes e avaliar o desempenho dos mesmos, é necessário um conjunto inicial de *hashtags* cuja classificação já seja conhecida. De forma geral, classificações manuais realizadas por humanos tendem a ser mais precisas, porém muito custosas do ponto de vista prático e do tempo requerido, chegando a ser inviáveis para conjuntos de dados muito extensos. Levando isto em conta, assim como observado na preferência de muitas pesquisas relacionadas, a abordagem utilizada neste trabalho foi a de anotações automáticas das classes para um conjunto reduzido de nós, cuja confiabilidade na classificação fosse alta o bastante para minimizar o nível de ruído.

5.1 Classificação baseada em Palavras-chave

O primeiro método utilizado para classificar um conjunto inicial das *hashtags* foi baseado em [1]. O autor defende que as *hashtags* utilizadas no Twitter podem ser de três tipos: Tópico, Sentimento, ou Misto. Um tópico seria alguma entidade ou assunto, como "#Google" ou "#NewYear"; Já um sentimento, algum termo utilizado para representar ou expressar emoções subjetivas, como "#like" ou "#angry"; E misto seria, intuitivamente, quando ambos um tópico e um sentimento estão presentes na mesma *hashtag*, como em "#ILoveApple".

Na pesquisa citada, o autor comprova que ao se considerar o significado literal da própria *hashtag* como informação supervisionada para a classificação, obtém-se uma melhora significativa no desempenho obtido. Ou seja, independente dos *tweets* associados à *hashtag* "#like", dado que sabemos que a própria *hashtag* representa literalmente um sentimento com polaridade fortemente positiva, podemos simplesmente atribuir tal polaridade ao nó, e assumir que, neste caso, "#like" é um representante da classe positiva.

Com base nisso, foram selecionados um conjunto inicial de palavras obtidas através de um *lexicon* disponível online[4], acrescidas de algumas gírias e termos informais, resultando em um conjunto de 220 palavras, representando 110 sentimentos fortemente positivos ("happy", "awesome", "love") e 110 sentimentos fortemente negativos ("upset", "horrible", "hate"). Para cada *hashtag* no grafo, se ela possui alguma destas palavras como uma *substring* sua, como no caso de "#ILoveApple", a polaridade correspondente lhe é atribuída.

[4] http://www.enchantedlearning.com/wordlist/

Após este procedimento, o conjunto de nós classificados no grafo pequeno, de 5 mil nós, ficou composto por 308 instâncias, sendo 236 positivas e 72 negativas, uma proporção de aproximadamente 3:1. No grafo maior, de 20 mil vértices, isto resultou em 1166 sentimentos classificados, sendo 890 positivos e 276 negativos, aproximadamente a mesma proporção observada no primeiro caso.

Em ambos os casos, o total de *hashtags* que foram classificadas beira em torno de apenas 6%, porém com altíssimo grau de confiabilidade. Existem alguns raros ruídos, como no caso da *hashtag* "#assad", referente ao presidente da Síria Bashar al-Assad, contendo a *substring* "sad", porém a frequência é insignificante quando comparado a outros métodos de classificação automática, não impactando o ótimo desempenho da abordagem.

5.2 Classificação baseada em Estimativas

Para complementar o conjunto inicial obtido através do procedimento descrito na seção anterior, foi utilizado uma estratégia mista baseada em [1] e [3], uma vez que conhecer apenas 6% dos nós existentes pode ser uma barreira significativa para os algoritmos de classificação coletiva.

A abordagem utilizada nesta seção na realidade cumpre dois papéis. O propósito inicial era obter uma maneira alternativa de se estimar as probabilidades iniciais dos nós não-classificados, para prover esta informação aos algoritmos de inferência coletiva. O Relaxation Labeling necessita de uma estimativa inicial da chance de cada nó pertencer a cada classe, sem a qual não é possível iniciar o algoritmo. Caso tal estimativa não seja informada, o Netkit utiliza a distribuição inicial das classes dentre os nós conhecidos, para ser atribuída a todos os nós. Ou seja, se inicialmente são conhecidos 80 nós com polaridade positiva e 20 nós com polaridade negativa, e não seja informada a probabilidade dos nós restantes pertencerem a cada uma das classes, assume-se que todos tenham a mesma chance de 80% para a classe positiva e 20% para a negativa, o que obviamente não é a melhor alternativa. Já o Iterative Classification, por sua vez, não mantém estas probabilidades durante a sua execução – ele sempre atribui ao nó de forma completa a classificação cuja probabilidade seja maior, o que, no cenário descrito acima, significaria que todos os nós não-conhecidos iniciariam a execução do algoritmo já sendo considerados da classe positiva, o que é ainda mais catastrófico. Outra opção menos desbalanceada, no caso apenas do Iterative Classification, é simplesmente ignorar os nós que ainda não se tem informações a respeito, deixando para apenas considera-los em iterações futuras, quando estes já tivessem sido influenciados pelos seus vizinhos, obtendo assim alguma classificação.

O segundo papel, como mencionado inicialmente, seria aproveitar as *hashtags* cuja estimativa inicial fosse surpreendentemente alta, para complementarem o conjunto inicial como sendo *hashtags* de polaridade já pré-definida. Para isto, foi utilizada, dentre outros critérios, o limiar de 95% de probabilidade na estimativa inicial. Ou seja, se uma dada

hashtag, ao ter sua probabilidade inicial estimada, possuísse mais do que 95% de chance de pertencer à classe positiva ou negativa, ela seria removida do conjunto de *hashtags* desconhecidas e inserida no conjunto de *hashtags* pré-classificadas, como sendo da respectiva classe estimada.

Para obter estas estimativas iniciais, foi utilizado um classificador de texto baseado em Support Vector Machine (SVM)[5], estado-da-arte no quesito de análise de sentimentos em texto [1] [8] [9], para classificar os *tweets* atribuídos a cada *hashtag*, e que por sua vez contribuem para a classificação da *hashtag* através do critério de "votação". Os resultados agregados referentes aos *tweets* individuais, portanto, representam a distribuição de probabilidade das respectivas *hashtags* como uma proporção direta, através da fórmula:

$$\Pr(y_i|h_i) = \frac{\sum_{\tau \in \mathcal{T}_i} \Pr_{y_i}(\tau)}{\sum_{\tau \in \mathcal{T}_i} \Pr_{pos}(\tau) + \sum_{\tau \in \mathcal{T}_i} \Pr_{neg}(\tau)} \quad (5.1)$$

Onde h_i representa a iésima *hashtag*, $y_i \in$ {pos,neg}, e $\mathcal{T}_i$ são todos os *tweets* que contém a *hashtag* h_i.

O SVM, como qualquer classificador de texto tradicional, necessita de um conjunto de textos utilizados para treinamento, através dos quais são extraídos vetores de atributos (*features*) considerados relevantes para a classificação. Neste trabalho foi utilizada uma abordagem simples, devido principalmente ao fato de o detalhamento do SVM fugir do escopo abordado. Como *features*, foram considerados os mesmos termos utilizados no *lexicon* de sentimentos da seção anterior, e foram selecionados um *tweet* para cada *feature*, que possuísse apenas ela, como conjunto de treinamento, sendo pré-classificados com a polaridade da *feature* correspondente, para então serem utilizados pelo SVM para induzir a classificação dos *tweets* restantes.

A fórmula (5.1), no entanto, como proposta por [1], desconsidera os *tweets* objetivos, de polaridade neutra, pois a saída do SVM é binária, o que significa que $\Pr_{pos}(T) \in \{0,1\}$ e o mesmo para a probabilidade negativa. No entanto, observando alguns casos específicos, podemos notar que mesmos os votos neutros podem ser significativos. Um exemplo disto é um caso em que uma determinada *hashtag*, esteja associada a 500 *tweets* neutros e 10 *tweets* positivos. A fórmula (5.1) atribui a esta *hashtag* uma probabilidade de 100% de chance de ser positiva, o que não é muito preciso, dado que a vasta maioria das vezes em que ela foi empregada, foi em um contexto neutro, induzindo-nos a pensar que as poucas vezes em que ela estava de fato associada a um *tweet* positivo, tal polaridade havia sido atribuída por algum outro termo que não esta *hashtag*.

[5] http://svmlight.joachims.org/

Como uma alternativa para a fórmula (5.1), este trabalho propõe, portanto, a seguinte fórmula, que leva em conta o "peso" de *tweets* neutros, aproximando as estimativas em direção à probabilidade de 50%:

$$\Pr(y_i|h_i) = \frac{2\sum_{\tau \in \mathcal{T}_i} \Pr_{y_i}(\tau) + |\mathcal{T}_{ni}|}{2|\mathcal{T}_i|} \quad (5.2)$$

Onde $\mathcal{T}_{ni}$ representa todos os *tweets* neutros com a *hashtag* h_i, ou seja, aqueles em que $\Pr_{pos}(T) = \Pr_{neg}(T) = 0$. No caso anterior, com 500 *tweets* neutros e 10 positivos, a nova fórmula irá então atribuir a probabilidade de aproximadamente 51% para a classe positiva, e 49% para a negativa, o que é um pouco mais sensato do que o resultado anterior de 100% e 0%. Nos casos onde não há *tweets* neutros, o resultado permanece exatamente igual.

Podemos afirmar que a fórmula (5.2) é mais "conservadora" que a (5.1), ao levar em conta estes *tweets* neutros. Exemplos observados que dão suporte a esta abordagem são, por exemplo, a *hashtag* #love, que dentre os *tweets* coletados, mais de 98% dos que contém esta *hashtag* possuem uma polaridade positiva, enquanto para a *hashtag* #door, isto é o caso em apenas 45% dos *tweets*, um sinal óbvio de que, apesar desta *hashtag* raramente estar associada a uma polaridade negativa, ainda assim o seu potencial positivo não é tão significante. Para o aprimoramento do conjunto inicial de *hashtags* classificadas, são consideradas aquelas *hashtags* que, a partir da fórmula utilizada, obtiveram uma probabilidade positiva ou negativa superior a 95%, além de conterem pelo menos 5 *tweets* subjetivos, ou seja, não-neutros. Relembramos que, nos grafos utilizados, todas as *hashtags* estão associadas a pelo menos 10 *tweets* no grafo grande, e 50 *tweets* no grafo pequeno, uma vez que este foi o *threshold* utilizado para as arestas dos respectivos grafos.

Utilizando as fórmulas em conjunto com as classificações iniciais baseadas em palavras-chave, observamos os resultados nas tabelas abaixo, onde a primeira linha representa os valores originais obtidos sem o aprimoramento do classificador de texto. Nota-se que a fórmula (5.1) classifica muito mais *hashtags*, como esperado, devido à sua estimativa desenfreada que desconsidera os *tweets* neutros, permitindo uma gama maior de palavras ultrapassarem o limiar de 95%, presumindo-se porém uma taxa de ruído mais elevada associada a isto.

Tabela 5.1 Conjunto inicial de *hashtags* classificadas no grafo de 5 mil nós.

Abordagem de aprimoramento	Classificações Positivas	Classificações Negativas	Total de *hashtags* classificadas
–	236	72	308 (6%)
Fórmula (5.1)	873	354	1227 (25%)
Fórmula (5.2)	269	144	413 (9%)

Para os resultados experimentais, as 3 abordagens serão portanto comparadas: Não utilizar estimativas iniciais explícitas, deixando a critério do algoritmo, sendo os nós temporariamente ignorados pelo Iterative Classification, e a distribuição das classes sendo usada como probabilidade pelo Relaxation Labeling; Utilizar a fórmula (5.1); E utilizar a fórmula (5.2).

Tabela 5.2 Conjunto inicial de *hashtags* classificadas no grafo de 20 mil nós.

Abordagem de aprimoramento	Classificações Positivas	Classificações Negativas	Total de *hashtags* classificadas
–	890	276	1166 (6%)
Fórmula (5.1)	7016	1407	8423 (41%)
Fórmula (5.2)	1227	789	2016 (10%)

Além disso, as 3 abordagens serão também comparadas para a definição do conjunto inicial de *hashtags* classificadas, considerando ambas as fórmulas para o aprimoramento do conjunto inicial, e o conjunto original sem o auxílio das *hashtags* estimadas pelo classificador de texto.

CAPÍTULO 6

Experimentação

Este capítulo descreverá a metodologia utilizada na realização dos experimentos, usando as ferramentas e técnicas previamente discutidas, e apresentará os respectivos resultados obtidos, assim como uma análise dos mesmos.

6.1 Metodologia

Para os testes, foram exploradas ao máximo as habilidades do Netkit, de forma a se criar uma interface para testes sistemáticos de forma automatizada. O Netkit possui algumas limitações que precisaram ser contornadas, como por exemplo, apesar de teoricamente prover uma opção interna para a realização de *cross validation* nos experimentos, a mesma somente é aplicável quando todos os nós da rede a ser classificada são originalmente conhecidos – algo muito longe de ser o caso para as redes avaliadas neste trabalho, onde, dependendo da configuração, a quantidade de nós conhecidos varia entre 6% e 41%.

Portanto, foi implementado um mecanismo de validação cruzada estratificada utilizado como envoltório para o Netkit. Para cada combinação de configurações a serem testadas, era realizada uma *cross validation* em cima do Netkit, alterando o conjunto inicial de nós conhecidos e os nós a serem avaliados, de forma proporcional à quantidade de iterações desejadas, de forma que cada instância fosse utilizada exatamente uma única vez como parte do conjunto de avaliação. Para manter os conjuntos sempre o mais balanceado possível, os exemplares positivos e negativos são tratados separadamente, mantendo sempre as proporções mais próximas possíveis. Isto é alcançável através da seguinte fórmula, que permite que quaisquer duas iterações distintas da validação cruzada se diferenciem por no máximo a quantidade de um único nó de cada classe:

$$x = p\left\lfloor\frac{x}{k}\right\rfloor + q\left\lceil\frac{x}{k}\right\rceil \tag{6.1}$$

Onde x é a quantidade de instâncias de uma classe, e k a quantidade de iterações desejadas na *cross validation*, de forma que todos os termos da equação são inteiros. Quaisquer par de valores inteiros (x,k) é resolvível pela equação acima, que resulta em p iterações com y elementos, e q (possivelmente zero) iterações com $y+1$ elementos, daí a garantia de duas iterações distintas diferenciarem em no máximo 1 elemento de cada classe. Com isso, uma validação cruzada com $k = 5$ terá, portanto, aproximadamente 80% dos nós fornecidos com a classificação conhecida, e 20% reservados para o conjunto de avaliação em cada iteração. Neste trabalho, foram testados os valores $k \in \{2, 3, 5, 10\}$.

Em uma camada superior a este mecanismo, as configurações do Netkit são alteradas conforme as combinações pré-determinadas a serem testadas, para então a ferramenta ser executada, e os resultados extraídos. As "combinações" dizem respeito aos diferentes classificadores locais, relacionais e coletivos utilizados, assim como a quantidade de iterações da validação cruzada, que afeta a proporção de nós conhecidos durante o teste, e o critério de se utilizar apenas os nós classificados através de palavras-chaves, ou o conjunto incrementado por alguma estratégia do classificador de texto do capítulo 5. Os testes foram ainda realizados em ambos os grafos descritos no capítulo 4.

Para os classificadores locais, relacionais, e de inferência coletiva, foram realizados testes preliminares para selecionar aqueles que obtivessem os melhores resultados, a serem utilizados nos testes completos:

Para o módulo de inferência coletiva, foram testados o Relaxation Labeling e o Iterative Classification, ambos baseados na heurística iterativa que busca a convergência. No Netkit, o Relaxation Labeling utiliza também um limite de iterações, e parâmetros baseados em *Simulated Annealing* para "esfriarem" progressivamente o impacto das mudanças ao longo das iterações. Foram utilizados os parâmetros padrões propostos pelos autores.

Para o classificador relacional, foram utilizados o *Weighted Vote Relational Neighbor* (wvRN) [14], *Network-only Bayes* (n.o.Bayes) [7], *Class Distributional Relational Neighbor* (cdRN) [7], *Network-only Link-based Classifier* (n.o.LB) [15], e *Logistic Regression* (Logistic) [7]. As versões "Network-only" são as que foram adaptadas no Netkit para trabalharem de forma univariada levando em consideração apenas as classes (ou suas respectivas probabilidades) dos vizinhos de um nó.

Para o classificador local, foram utilizados as estimativas pré-classificadas pelo SVM em conjunto com ambos Relaxation Labeling e Iterative Classification, e como segunda variante, o Iterative Classification é combinado com o classificador trivial que atribui *null* aos nós desconhecidos, e o Relaxation Labeling é combinado com o classificador trivial que utiliza a distribuição de probabilidade das classes existentes no conjunto conhecido para ser aplicada inicialmente aos nós desconhecidos. Para a estimativa do SVM, ambos resultados da fórmula (5.1) e (5.2) se situavam muito próximos na precisão obtida através dos testes preliminares, sendo a escolha virtualmente indiferente, e a fórmula (5.2) utilizada para os testes completos.

Totalizaram, portanto, 2 classificadores locais, 5 relacionais, 2 coletivos, 3 conjuntos iniciais de *hashtags* conhecidas, e 4 proporções de conjuntos de treinamento/avaliação, resultando em 240 combinações de testes para cada um dos grafos. Foi avaliado ainda, também, o desempenho obtido pelo classificador de texto de forma isolada, para ser utilizado como patamar inicial na comparação com os resultados obtidos pelas técnicas de classificação coletiva. Dado que as fórmulas (5.1) e (5.2) sempre classificam o nó como sendo da classe que obteve a maior probabilidade, independente

deste valor ser 51% ou 100%, ambas resultam na exata mesma polaridade para todos os nós, e por isso obtém o mesmo resultado quando utilizadas de forma isolada para classificar as *hashtags*.

Por fim, foram também calculados e avaliados outros aspectos dos grafos utilizados no experimento, como a assortatividade e a modularidade dos mesmos, como serão apresentados abaixo.

6.2 Resultados

Os resultados comparados nesta seção, a menos que especificados de outra forma, são provenientes dos resultados agregados obtidos utilizando-se cada algoritmo ou abordagem. Isto é, na comparação entre os algoritmos para classificação coletiva, por exemplo, as combinações de cada algoritmo de inferência coletiva com todos os classificadores relacionais, locais, e etc, são considerados, e a média obtida é apresentada. Isto tem por objetivo comparar o desempenho de cada módulo de forma geral nesta rede de co-ocorrência de *hashtags*, porém análises mais específicas são também consideradas e apresentadas.

O primeiro desempenho a se observar é o resultante do classificador de texto em isolamento, sem nenhuma abordagem relacional ou coletiva, similar às abordagens tradicionais de aprendizagem de máquina. Como explicado na seção anterior, ambas as fórmulas (5.1) e (5.2) obtém exatamente o mesmo resultado nesta situação, que é de 78.90% de acerto no grafo pequeno, e 73.19% no grafo grande – ambos muito superiores aos resultados de aproximadamente 55% relatados em estudo similar [1]. Estes valores serão usados para comparação com as técnicas coletivas que obtiverem os melhores desempenhos.

A precisão relativamente alta obtida no grafo pequeno, se dá, em parte, devido à quantidade limitada e seleta de *hashtags* a serem classificadas: apenas 308, as quais todas possuem no mínimo 50 *tweets* relacionados, e em média uma ordem de grandeza a mais que isso. A informação disponível é portanto abundante e precisa – o que não é necessariamente o caso enfrentado na rede grande, onde existem 1166 *hashtags*, das quais podemos inferir com certeza que 858, que não estão presentes no grafo pequeno, possuem entre 10 e 49 *tweets* relacionados, muitos dos quais nem serão subjetivos. É, portanto, perfeitamente esperado que na medida em que consideramos mais vértices, e com menos informações, o nível de ruído irá aumentar de forma a prejudicar significativamente a precisão do classificador de texto, que irá se deparar com muito mais nós e muito menos informações, onde os nós com pouca informação, por serem majoritários, acabam tendo um peso muito maior na precisão final agregada.

Na figura 6.1, abaixo, observamos o desempenho do Relaxation Labeling (RL), quando comparado ao Iterative Classification (ICA). Notamos que, fora a única situação em

que eles atingem desempenho semelhante, o Iterative Classification segue o mesmo padrão de curva, porém sempre um pouco inferior ao Relaxation Labeling. A situação em que eles obtém resultados praticamente idênticos, é quando existem relativamente poucos nós, dos quais a maioria já é conhecida, o que a princípio representa o cenário mais fácil. A variação na porcentagem de nós pré-classificados é atrelada às diferentes quantidades de iterações na *cross validation*, de forma que a proporção 90/10 representa a média dos resultados obtidos em 10 iterações.

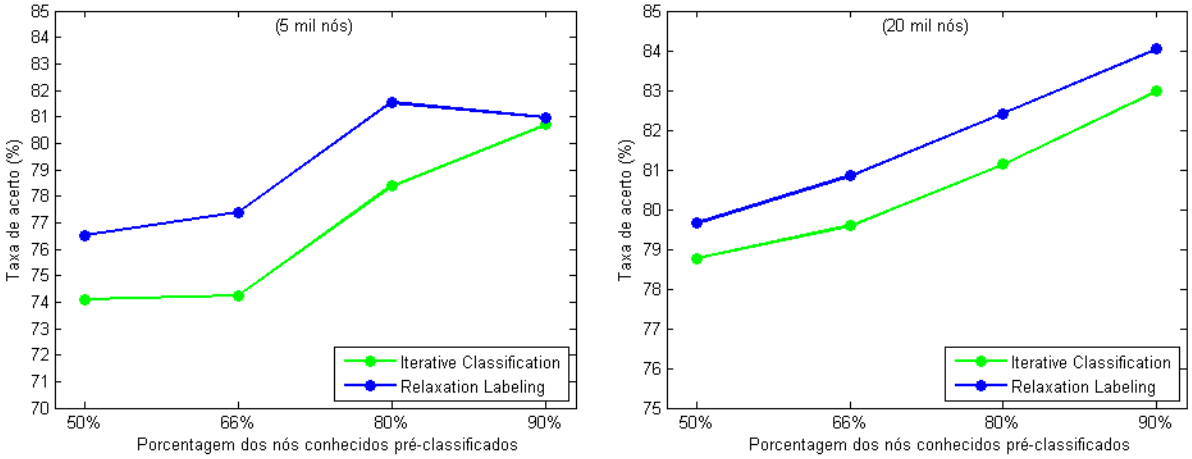

Figura 6.1 Precisão agregada das combinações de algoritmos que diferem apenas no uso do Relaxation Labeling ou Iterative Classification para inferência coletiva.

Em estudos mais antigos, o RL geralmente atingia resultados similares ou até inferiores ao ICA [4] [6] [7], porém isto era sempre atribuído ao fato de o RL originalmente calcular a correlação entre as classes apenas uma única vez no início do algoritmo, e manter este resultado congelado ao longo das iterações, utilizando-o nas suas fórmulas intermediárias. Por isto, em estudos mais recentes, incluindo no Netkit, a implementação utilizada é uma adaptação que refaz estes cálculos a cada iteração intermediária, não se prendendo tanto à informação inicial, fazendo com que esta tenha menos influência na classificação final, algo que tem se demonstrado vantajoso [4] [7].

Na figura 6.2, são comparados os testes onde inicialmente foram utilizadas as estimativas provenientes do classificador de texto, ou então não definir estimativas iniciais, delegando esta responsabilidade para um classificador local trivial – no caso do Relaxation Labeling, utilizar a distribuição de probabilidade das classes dos nós conhecidos inicialmente, e no caso do Iterative Classification, ignorar os nós ainda não conhecidos, até que recebam uma classificação através dos seus vizinhos em uma iteração futura. Esta abordagem para o ICA é mais adequada, pois em ambos os grafos existe apenas um componente conexo, e os nós positivos são majoritários no conjunto de treinamento, o que resultaria em o ICA classificar *todos* os nós desconhecidos como sendo positivos na etapa

inicial, o que atrapalharia o desempenho do mesmo, uma vez que ele considera apenas atribuições binárias baseadas na maior probabilidade de classe de um nó.

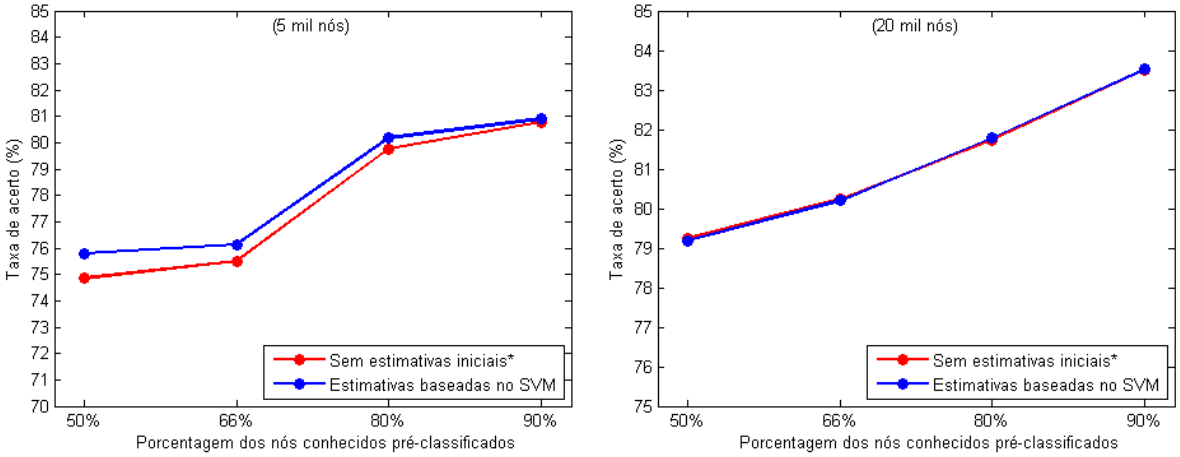

Figura 6.2 Comparação da precisão agregada nas abordagens utilizando o SVM para gerar as estimativas iniciais, ou um classificador local trivial.

Para os classificadores relacionais, o cenário se torna um pouco mais complicado, ao menos na rede pequena. A figura 6.3 mostra esta comparação, onde observa-se um empate do *Link-Based* e do *Logistic Regression* na liderança absoluta na rede grande, enquanto na pequena ambos apresentam resultados medíocres quando uma pequena porcentagem dos nós é inicialmente conhecida, sendo então dominados pelo cdRN, que por sua vez permanece estagnado e é ultrapassado nos cenários onde a proporção de *hashtags* a serem classificadas é pequena. Neste caso, com base apenas nesta informação, não há uma única técnica que seja melhor para todas as situações, o que faz com que critérios adicionais precisem ser levados em conta na escolha do algoritmo empregado, seja facilidade de implementação, ou desempenho mais eficiente em termos de custo computacional.

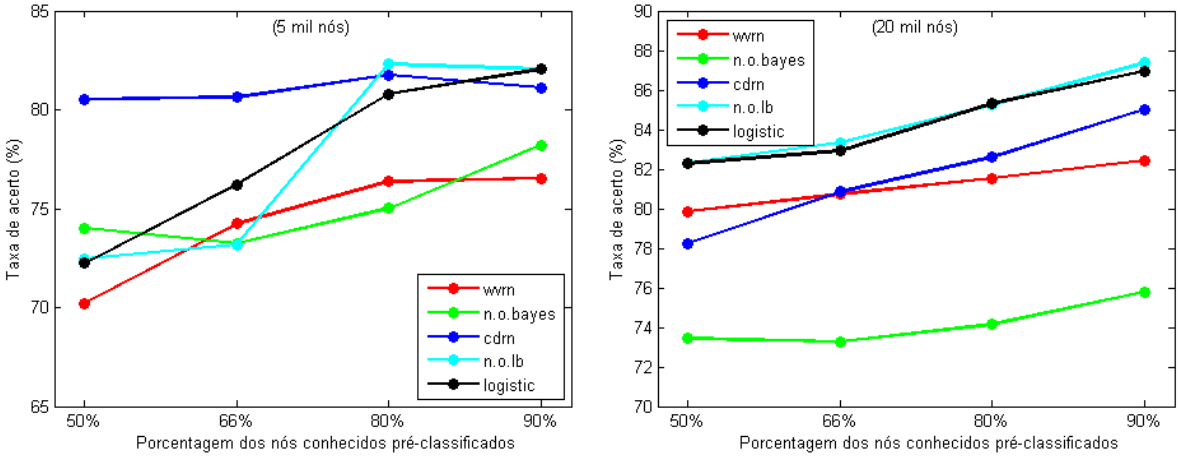

Figura 6.3 Precisão agregada das abordagens utilizando cada um dos cinco classificadores relacionais.

Com relação às abordagens de incrementar o conjunto de *hashtags* inicialmente classificadas, o impacto foi muito mais significativo, como pode ser visto na figura 6.4. Utilizando-se a fórmula (5.2), apesar de se obter uma quantidade muito pequena de classificações adicionais, as mesmas são muito mais precisas, o que é refletido na taxa de acerto quase unanimemente superior ao uso da fórmula (5.1). De fato, a abordagem de não fazer uso dos *tweets* neutros na estimativa das *hashtags*, possui uma taxa de ruído tão elevada, que no cenário onde poucos nós totais são inicialmente conhecidos, o desempenho chega a ser significativamente pior do que utilizar apenas o conjunto original, sem classificações adicionais.

Para manter a consistência dos resultados, em todos os 3 casos, as avaliações são realizadas somente com base no conjunto de *hashtags* originalmente classificadas através da técnica de palavras-chave, ficando as *hashtags* adicionais, provenientes da classificação de texto com alta estimativa, sendo usadas apenas para auxiliar a classificação, e não como parte do conjunto de avaliação.

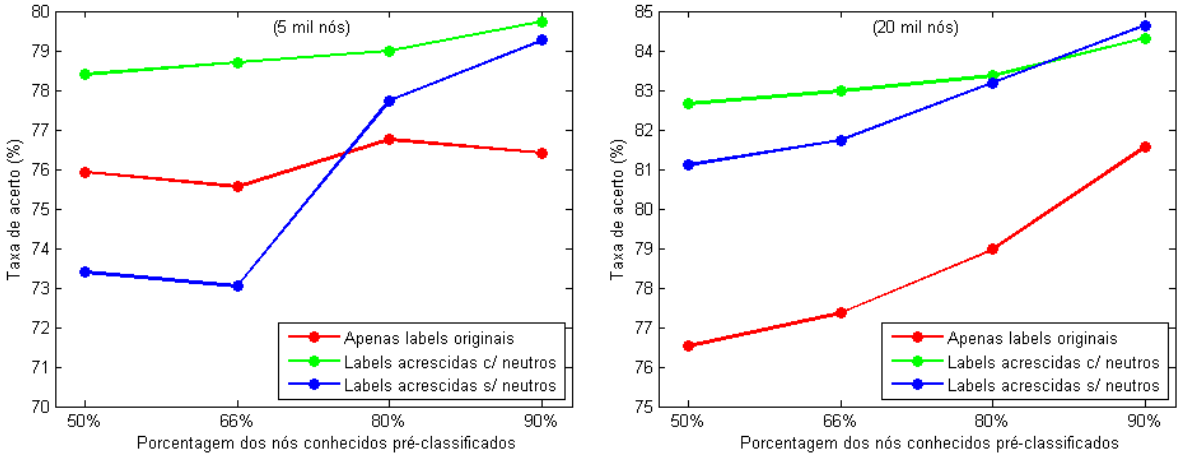

Figura 6.4 Precisão agregada das abordagens de incrementação do conjunto de *hashtags* inicialmente classificadas. "c/ neutros" diz respeito à aplicação da fórmula (5.2) no classificador de texto, onde se considera os *tweets* neutros; "s/ neutros" é referente ao uso da fórmula (5.1).

A partir destes resultados, podemos determinar quais técnicas e módulos obtiveram os melhores desempenhos, e focar a análise de forma mais detalhada nas combinações específicas que mais se destacaram, que seriam, portanto, as candidatas a serem utilizadas em um sistema final. Ficou claro que o Relaxation Labeling se demonstrou superior ao Iterative Classification para esta rede de co-ocorrência de *hashtags*, assim como utilizar as estimativas do SVM como distribuição de probabilidade inicial das classes de cada *hashtag* desconhecida também não trouxe prejuízos. Considerando também o uso de classificações adicionais para as *hashtags* iniciais com base nas que obtiveram alta probabilidade a partir

das estimativas do SVM, e utilizando os classificadores relacionais que demonstraram maior desempenho, obtemos as 3 melhores combinações, apresentadas na figura 6.5 abaixo.

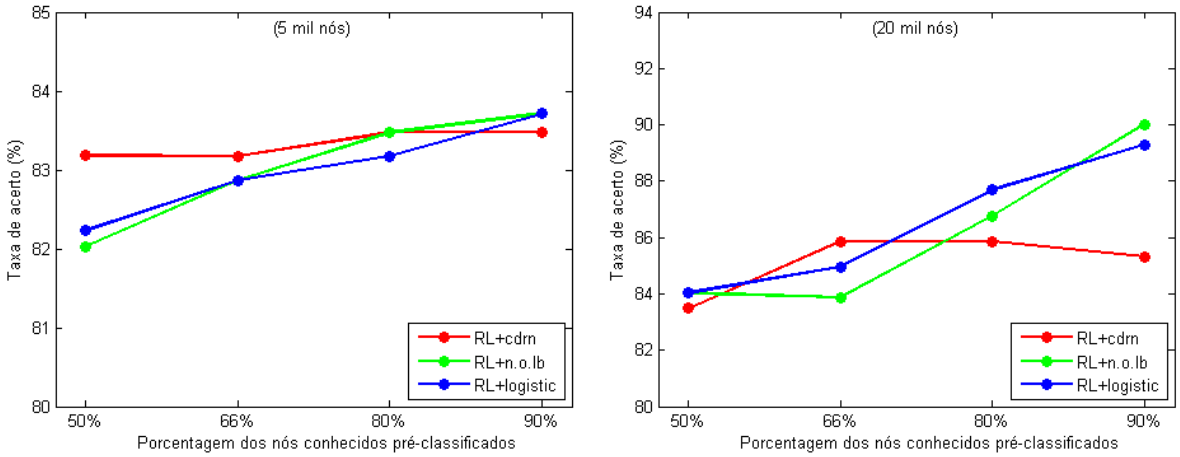

Figura 6.5 Precisão individual obtida pelas três combinações de melhor desempenho.

Nota-se mais uma vez o desempenho praticamente equivalente entre os classificadores relacionais *Link-Based* e *Logistic Regression*, que na rede grande parecem ser definitivamente a melhor opção, chegando a atingir taxas de até 90% de acerto quando existem muitas *hashtags* conhecidas, e poucas a se classificar. Isto faz com que eles sejam ideais por exemplo em aplicações onde a rede estivesse evoluindo com o tempo, onde o cenário sempre seria poucos novos nós precisando ser classificados. Já o cdRN demonstrou não se beneficiar de uma proporção maior de nós conhecidos, obtendo um melhor resultado apenas nos casos em que ambas a rede e a quantidade de *hashtags* conhecidas eram baixas. Para aplicações que se beneficiariam de um método assim, [7] cita alguns bons exemplos, como cenários antiterrorismo e detecção de fraudes, onde existem poucos dados disponíveis, e ainda menos dos quais se tem certeza da sua natureza.

Comparando estes desempenhos obtidos pelas melhores combinações, com os obtidos pelo classificador de texto de forma isolada, podemos notar uma melhoria significativa: No grafo com 5 mil nós, o SVM por conta própria obteve uma precisão de 78.9%, contra 83.7% da classificação coletiva; E no grafo com 20 mil nós, a diferença foi ainda mais considerável, de 73.2% do SVM para até 90% com a classificação coletiva.

Detalhando ainda mais os resultados, podemos analisar as taxas de precisão e cobertura especificamente relativas às instâncias positivas e negativas. Aqui, é considerado que quando um classificador estima uma instância como sendo exatamente 50% positiva e 50% negativa, isto conta como uma abstenção, que não contribui para a piora da precisão, apenas para a piora da cobertura, uma vez que a classificação não foi nem positiva e nem negativa. Isto é diferente das taxas de precisão total apresentadas anteriormente nos

gráficos acima, onde foi considerado o total de acertos em relação ao total de instâncias sendo classificadas.

Nas tabelas 6.1 e 6.2 abaixo, são utilizadas as mesmas configurações que na figura 6.5, exceto que é considerado apenas o resultado relativo à proporção 90/10 na validação cruzada. É considerado o Relaxation Labeling, as estimativas iniciais com base no SVM, e os nós iniciais incrementados pelas estimativas do SVM.

Pode-se notar que, em ambas as redes, o cdRN se destaca principalmente na sua cobertura dos nós positivos, e a precisão na classificação das instâncias negativas. No entanto, a quantidade de falsos positivos é muito maior do que nos outros dois algoritmos, assim como a abstenção na classificação de algumas instâncias negativas.

Por outro lado, tanto o *Link-Based* quanto o *Logistic Regression* conseguem melhorar significativamente os seus desempenhos na rede maior, com o *Link-Based* apresentando uma ligeira vantagem, além de ambos os algoritmos obterem uma cobertura muito melhor para as instâncias negativas, apesar de um ruído também muito mais elevado para os falsos positivos.

Em nossa rede, ambos os tipos de erro, falsos negativos e falsos positivos, possuem a mesma implicação. No entanto, em sistemas onde um tipo de erro seja mais prejudicial que outro, ou a ausência de classificação não seja aplicável ou tolerável, caberia então levar em consideração este comportamento mais detalhado, para determinar qual algoritmo seria mais confiável para o uso no sistema, mesmo que não apresentasse a melhor taxa de acerto global.

Tabela 6.1 Precisão e Cobertura das melhores combinações na rede com 5 mil nós, considerando as instâncias positivas e negativas.

Classificador Relacional	Precisão Positiva	Cobertura Positiva	Precisão Negativa	Cobertura Negativa
cdRN	84.57%	**95.72%**	**97.24%**	38.11%
Link-Based	**92.71%**	81.17%	59.52%	**78.41%**
Logistic	91.94%	82.98%	62.44%	73.05%

Tabela 6.2 Precisão e Cobertura das melhores combinações na rede com 20 mil nós, considerando as instâncias positivas e negativas.

Classificador Relacional	Precisão Positiva	Cobertura Positiva	Precisão Negativa	Cobertura Negativa
cdRN	85.04%	**96.66%**	**98.58%**	40.54%
Link-Based	**93.33%**	90.23%	77.03%	**79.50%**
Logistic	91.15%	90.87%	76.91%	73.76%

Analisando a estrutura da rede, confirmamos ainda a presença da assortatividade: Na rede pequena, quaisquer duas *hashtags* vizinhas escolhidas possuem 91.2% de chance de apresentarem a mesma polaridade, contra 53.1% de chance caso fossem escolhidas de forma aleatória. Já na rede grande, a probabilidade de *hashtags* vizinhas possuírem a mesma polaridade é de 78%, contra 51.1% no caso da escolha aleatória. Estes valores se tornam ainda mais dramático caso seja levado em consideração a frequência das arestas: 96% para a rede pequena, e 94% para a rede grande.

Esta característica, refletida diretamente nos links presentes entre os nós, pode ainda ser explorada por outras técnicas de mineração de links. Na figura 6.6 observamos um *close-up* de um pequeno sub-grafo das *hashtags*, sobre o qual foi executado o algoritmo de detecção de comunidades da ferramenta Gephi.

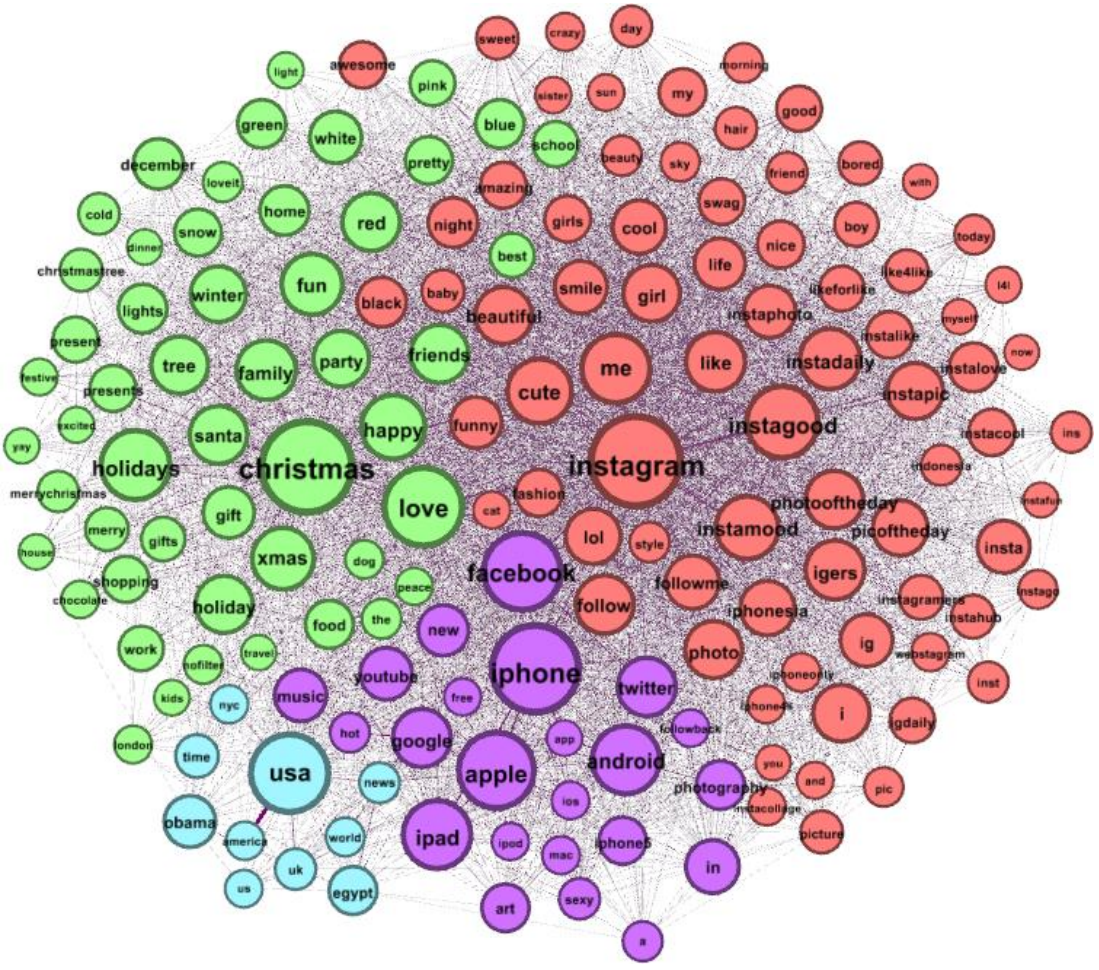

Figura 6.6 Visualização de um sub-grafo da rede quando dividido em 4 comunidades. O tamanho de cada nó é proporcional à sua intermediação no grafo.

É possível notar imediatamente a correspondência semântica entre nós de uma mesma comunidade. Ao se dividir em mais comunidades, observa-se, por exemplo, as cores

red/white/green acima associadas a Christmas, e *black* associado ao Instagram, formarem uma nova comunidade composta apenas por cores. Isso é baseado unicamente na ligação entre os nós, ou seja, na co-ocorrência das *hashtags* propriamente dita, sem levar em conta o restante das informações textuais presentes nos *tweets*. Isto abre caminho para um estudo mais a fundo de como este, e possivelmente outros algoritmos baseados na interligação de nós, poderiam também ser usados na mineração de informações em redes como o Twitter, complementando o que já se pode conseguir através da classificação coletiva.

CAPÍTULO 7

Conclusão

Com base nos experimentos, observamos como informações de links entre entidades são dados valiosos para tarefas de classificação, permitindo que os algoritmos de classificação coletiva façam um uso mais inteligente da informação de nós interligados mesmo que indiretamente. Assim, conseguem superar significativamente os algoritmos de estado-da-arte para análise de sentimentos que consideram os nós de forma independente, como o classificador de texto baseado no SVM, que atingiu uma taxa de acerto significativamente inferior aos algoritmos mais eficazes para classificação coletiva na rede estudada.

Mesmo dentre os classificadores coletivos, ainda existem distinções nos desempenhos obtidos em cada situação específica, não existindo, portanto, um classificador incondicionalmente melhor. É necessário avaliar as prioridades do sistema em questão, como o impacto causado por um falso negativo ou falso positivo, assim como a proporção de nós cuja classificação é inicialmente conhecida, para então poder se escolher o melhor classificador. De forma geral, na rede de co-ocorrência de *hashtags*, o uso do Relaxation Labeling para inferência coletiva e as estimativas iniciais baseadas em SVM se demonstraram superiores, assim como o aprimoramento do conjunto inicial de *hashtags* a partir da estratégia mais conservadora do classificador de texto, quando consideramos também os *tweets* neutros. Dentre os classificadores relacionais, o *Link-Based* e o *Logistic Regression* se demonstraram muito próximos e consideravelmente superiores aos outros, com exceção do cdRN que se destaca mais em alguns cenários específicos, e poderia ser mais apropriado para aplicações nestes cenários, como o de uma proporção pequena de classificações inicialmente conhecidas.

Vimos também que o conceito de assortatividade, presente nas mais diversas redes, é também observado nesta rede de co-ocorrência de *hashtags*, permitindo assim que algoritmos de detecção de comunidades consigam obter resultados expressivos mesmo sem utilizar nenhuma outra informação associada ao nó. Investigar esta relação mais a fundo pode ser crucial para uma possível melhoria, onde poderíamos, por exemplo, detalhar as *hashtags* em mais categorias do que apenas a sua polaridade “boa” e “ruim”, a exemplo das categorias implícitas pela detecção de comunidades. O sentimento associado a estas *hashtags*, por sua vez, também poderia ser alvo de um maior detalhamento, onde um sentimento não seria detectado como apenas “ruim”, mas sim como relativo a tristeza, raiva, ou frustração, etc. Estas são algumas das atuais limitações deste projeto, e que poderiam ser superadas possivelmente com este auxílio de outras estratégias baseadas em mineração de links. Outra limitação diz respeito à metodologia de testes, uma vez que a seleção de *hashtags* para o conjunto de treinamento e avaliação foi feita sem levar em conta

a estrutura da rede, ou seja, de forma aleatória, enquanto que em redes reais é possível que se conheça muitos nós de um *cluster*, e nenhum de outro, o que pode influenciar os resultados entre os algoritmos. Exemplos de redes assim são as de detecção de fraudes, e terrorismo, onde se sabe sobre vários integrantes de um mesmo grupo, porém nada de outro. Esta característica abre caminho ainda para um estudo sobre o custo de se classificar uma dada entidade, em conjunto com o benefício que aquela classificação traria através dos resultados de uma classificação coletiva, de forma a se obter uma métrica de custo-benefício que auxiliaria no direcionamento de investigações criminais, por exemplo.

Outro possível trabalho futuro seria manter uma análise contínua com relação a *hashtags* específicas, para se acompanhar em tempo real a mudança na sua polaridade sob a percepção dos usuários ao longo do tempo, uma vez que esta análise de sentimentos tem o caráter temporal de refletir apenas o que foi expressado nos *tweets* em uma dada época, sendo assim altamente influenciado por eventos de grande impacto, como um anúncio bom ou ruim por parte de uma empresa.

Por fim, ressaltamos o quão bem as ferramentas de análise de sentimentos têm se saído no Twitter, com aplicações nas mais diversas áreas, desde financeira, a exemplo da previsão de tendências de ações de empresas, até política, como no acompanhamento do apoio aos partidos políticos durante eventos feito os debates presidenciais. Com o uso de técnicas de mineração de links, como a classificação coletiva, predição de links e detecção de comunidades, estas aplicações têm o potencial de se expandirem cada vez mais, tanto em abrangência quanto em desempenho, e novas aplicações deverão surgir, cada vez mais aumentando a nossa capacidade de transformar em conhecimento útil esta informação quase ilimitada a que temos acesso na internet.

Referências Bibliográficas

[1] X. Wang, F. Wei, X. Liu, M. Zhou, and M. Zhang, "Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach," *in Proceedings of the 20th ACM Conference on Information and Knowledge Management*, pp. 1031–1040, 2011.

[2] L. Getoor and C. P. Diehl, "Link mining: a survey," *ACM SIGKDD Explorations Newsletter*, vol. 7, no. 2, pp. 3–12, 2005.

[3] J. Rabelo, R. Prudêncio and F. Barros, "Collective Classification for Sentiment Analysis in Social Networks", *In Proceedings of the 24th IEEE International Conference on Tools with Artificial Intelligence*, 2012.

[4] R. Angelova and G. Weikum, "Graph-based text classification: learn from your neighbors". *In ACM SIGIR*, pp. 485–492, 2006.

[5] J. Bollen, B. Gonçalves, G. Ruan, and H. Mao, "Happiness is assortative in online social networks," *Artificial Life*, vol. 17, no. 3, pp. 237–251, 2011.

[6] S. Chakrabarti, B. Dom, and P. Indyk, "Enhanced hypertext categorization using hyperlinks," in *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, pp. 307–318, 1998.

[7] S. A. Macskassy and F. Provost, "Classification in networked data: A toolkit and a univariate case study," *Journal of Machine Learning Research*, vol. 8, pp. 935–983, 2007.

[8] X. Meng, F. Wei, X. Liu, M. Zhou, S. Li, and H. Wang, "Entity-Centric Topic-Oriented Opinion Summarization in Twitter". *KDD'12*, 2012.

[9] M. Conover, B. Gonc¸alves, J. Ratkiewicz, A. Flammini, and F. Menczer, "Predicting the political alignment of twitter users," *in Proceedings of 3rd IEEE Conference on Social Computing (SocialCom)*, 2011.

[10] R. G. Cowell, A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter. "Probabilistic networks and expert systems". Springer, 1999.

[11] S. Goorha and L. Ungar, "Discovery of significant emerging trends," in *Proc. of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 57–64, 2010.

[12] L. Barbosa and J. Feng, "Robust sentiment detection on twitter from biased and noisy data". In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters, COLING '10*, Stroudsburg, PA, USA. Association for Computational Linguistics, pp. 36–44, 2010.

[13] D. Davidov, O. Tsur, and A. Rappoport, "Semi-supervised recognition of sarcastic sentences in twitter and amazon". In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning, CoNLL '10*, Stroudsburg, PA, USA. Association for Computational Linguistics, pp. 107–116, 2010.

[14] S. A. Macskassy and F. Provost, “A simple relational classifier”. In *Proceedings of the Multi-Relational Data MiningWorkshop (MRDM) at the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003.

[15] Q. Lu and L. Getoor, “Link-based classification”. In *Proceedings of the Twentieth InternationalConference on Machine Learning (ICML)*, pp. 496–503, 2003.

[16] D. Jensen, J. Neville, and B. Gallagher. “Why collective inference improves relational classification”. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 593–598, 2004.