Ciência de Redes
Aprendizado Supervisionado Aplicado à Predição de Links

Celio Henrique Nogueira Larcher Junior

1. Introdução

A utilização de redes é alternativa promissora para modelagem de diversos problemas do mundo real, tendo especificamente a representação das conexões entre nós grande significância em vários destes contextos. Neste rumo, o problema de predição de links em uma rede passa a apresentar grande interesse para uma vasta gama destas aplicações, dado que em tais situações a possibilidade de identificar novas conexões é informação extremamente relevante. Exemplos de tais aplicações são sistemas de recomendação de compras, indicações para novas colaborações acadêmicas e análise de possíveis conexões em redes de contado de terroristas.

Simplificadamente o problema de predição de links pode ser descrito com o seguinte enunciado: dado a situação de um grafo dinâmico G=(V,E) em um instante de tempo qualquer, deseja-se saber quais conexões são mais prováveis de acontecer em algum momento no futuro.

Na literatura observam-se trabalhos que buscam associar informações da estrutura da rede a dados provenientes do domínio da aplicação, possibilitando fornecer um modelo acurado e de boa capacidade preditiva, mas limitando o escopo da abordagem desenvolvida [1, 6]. Em outro rumo, foram observadas abordagens generalistas que envolvem apenas a estrutura da rede, utilizando medidas como distribuição de grau, semelhança entre nós, cálculos de distâncias, entre outras [3]. Esta segunda opção será escolhida neste trabalho.

Relacionado as abordagens generalistas, um grande esforço é posto no desenvolvimento de técnicas voltadas a medidas baseadas em similaridade, sejam estas com escopo local, como análise de vizinhança, ou com escopo global, como análises de caminho mais curto, sendo que um estudo de diversas destas técnicas pode ser visto em [5].

Uma outra opção não tão abordada está na utilização de técnicas de aprendizado de supervisionado. A predição de links pode ser vista como um problema de classificação, onde, para cada par de vértices, estes podem estar ou não conectados. É possível então extrair atributos da rede de forma às informações relativas a cada par de nós se apresentarem como dados de entrada para um sistema de aprendizado. Seguindo este procedimento, as mais diversas técnicas de aprendizado supervisionado são passíveis de serem aplicadas, sendo necessário identificar quais atributos devem ser utilizados. Alguns exemplos de trabalhos seguindo esta linha são [1, 2, 3, 6].

Neste sentido, dadas as diversas métricas possíveis de serem extraídas da estrutura de uma rede, este trabalho espera verificar quais dentre estas são mais adequadas ao problema de predição de links via aprendizagem de máquina. De fato, é provável que técnicas diferentes de aprendizagem possam ter melhor desempenho com um conjunto diverso de atributos. Da mesma forma, diferentes instâncias possivelmente terão inclinação maior a alguns atributos em detrimento de outros, mas espera-se que, pelo menos, um subconjunto promissor possa ser identificado, além da possibilidade de se observar qual papel os atributos básicos da rede possuem neste problema.

Ainda é relevante comentar que, sendo esta uma abordagem generalista do problema, não serão considerados atributos específicos a qualquer domínio de aplicação, limitando-se a verificação apenas daqueles relacionados a estrutura da rede.

2. Algoritmos de aprendizado supervisionado

Algoritmos de aprendizado supervisionado são uma classe de algoritmos que tem como objetivo a previsão do valor de algum atributo especificado para uma amostra (atributo de saída), através dos valores presentes nos demais atributos desta mesma amostra (atributos de entrada). A descoberta das relações entre os atributos é feita através de um processo denominado treinamento, que consiste na apresentação de diversos exemplos com o correspondente valor esperado de saída, para que o algoritmo seja capaz de aprender e generalizar as relações existentes entre atributos. Espera-se que, após o treinamento, os modelos gerados pelos algoritmos sejam capaz de inferir valores para o atributo de saída em novas amostras não apresentadas anteriormente.

Seguindo este princípio, existe uma vasta gama de algoritmos de aprendizagem supervisionada, utilizando-se de diferentes abordagens para generalização dos dados. Neste sentido para uma avaliação mais acurada do comportamento dos atributos da rede, buscou-se selecionar um subconjunto dos algoritmos de aprendizagem que abrangesse diferentes tipos de abordagens, mas também guiando a escolha pela popularidade das técnicas. Desta forma, os seguintes algoritmos foram utilizados para avaliação das métricas das redes para a predição de *links*:

- Árvore de Decisão (J48);
- Random Forest;
- Suport Vector Machine (SMO);
- Naive Bayes;

A implementação destes algoritmos foi obtida da ferramenta *Weka* [9], popular *framework* de mineração de dados. Ainda, para as técnicas que necessitam de alguma forma de parametrização, foram utilizados os valores padrões da ferramenta.

3. Predição de *links* e aprendizado supervisionado

O problema de predição de *links*, como comentado, busca encontrar possíveis pontos de conexão em um instante futuro, considerando uma rede variante no tempo. De maneira mais detalhada, tem-se a seguinte descrição: dado a situação de um grafo variante no tempo $G = (V, E, t)$ no instante de tempo t, espera-se realizar previsões para novas conexões que possam ocorrer no intervalo (t ,$t + \Delta t$), onde Δt é a janela de tempo investigada. Desta forma, espera-se que, baseado nas informações da rede no estado presente (t), sejam identificadas características que tornem possível esta inferência. Neste sentido, o problema se adequa de forma interessante ao aprendizado supervisionado, dado que os algoritmos tem como foco encontrar características dos dados que os tornem capaz de generalizar resultados para novas observações.

Seguindo este princípio, o problema de predição de *links* pode ser visto como um problema de classificação, onde a partir das informações relativas a um par de nós, existem duas classes possíveis: “uma aresta entre determinado par de nós surgirá em um intervalo de tempo futuro” e “não é esperado que surja uma aresta entre este par de nós neste intervalo de tempo”. Ainda, apresentando um conjunto de amostras com exemplos de pares de nós não conectados, com estas contendo tanto casos onde surgirão arestas no intervalo de tempo em questão, quanto casos em que as arestas se manterão ausentes, é possível buscar características que generalizem os dois estados de forma a possibilitar a previsão de novas arestas neste grafo em intervalos de tempo subsequentes.

Apesar desta adaptação natural, uma dificuldade na utilização desta abordagem está em, a princípio, as redes não apresentam uma estrutura que torne viável o treinamento de técnicas de aprendizado, dado que as informações, especialmente relacionadas a estrutura da rede, não estão

explicitas para verificação. Desta forma, para criação do conjunto de treinamento e posterior submissão de novas amostras, é necessário o pré-processamento do grafo, quando serão recolhidas as informações julgadas pertinentes para cada par de nós.

Ainda, este é um procedimento que pode se tornar proibitivo a depender do conjunto de atributos verificados e a dimensão da rede. Apesar disto, em métricas de conhecimento local ou em redes de dimensão pouco elevada, este fator não se torna um impeditivo podendo os algoritmos possuírem um bom desempenho.

4. Treinamento no problema de predição de *links*

Como comentado, numa fase anterior ao treinamento dos algoritmos é necessário o processamento do grafo para construção da base de treinamento. Para isto, foram selecionados pares de nós correspondendo a uma fração do número de arestas (10%), sendo estes nós desconectados no momento presente. Em cada par selecionado foi realizado o cômputo dos atributos definidos para o treinamento, além de ser adicionada a informação relativa a existência de uma futura aresta ou se esta permanece ausente, identificando a que classe o par em questão pertence. Buscou-se ainda que a construção deste conjunto o torne balanceado, com o número de amostras em cada uma das classes semelhante. Como atributos foram utilizados:

- caminho mais curto entre os vértices do par (*short_path*)
- número de vizinhos em comum entre os vértices do par (*common_neighbors*)
- grau de ambos os vértices (*degree_H*, *degree_L*)
- centralidade por intermediação de ambos os vértices (*betweenness_centrality_H*, *betweenness_centrality_L*)
- centralidade por proximidade de ambos os vértices (*closeness_centrality_H*, *closeness_centrality_L*)
- centralidade por auto-vetor de ambos os vértices (*eigenCentrality_H*, *eigenCentrality_L*)
- coeficiente de clusterização de ambos os vértices (*clustering_H*, *clustering_L*)
- excentricidade de ambos os vértices (*eccentricity_H*, *eccentricity_L*)

Os atributos relativos a características do nó (portanto aparecendo em pares) foram discriminados de forma ao de maior valor se apresentar sempre em um dos campos do conjunto de treinamento (com sufixo *_H*), enquanto o de menor valor é atribuído ao campo restante (com sufixo *_L*), fato que ocorre independentemente a que nó os valores pertençam. Exemplificando esta prática, tomando um par de nós qualquer (α,β), onde o nó α tem grau 5 e coeficiente de clusterização 0.3 e o nó β tem grau 7 e coeficiente de clusterização 0.1, ao realizar a indexação deste par a seguinte relação de atributos e campos seria estabelecida: *degree_H*=7, *degree_L*=5, *clustering_H*=0.3 e *clustering_L*=0.1. Esta prática é adotada de forma a diminuir o espaço de busca e homogeneizar os dados utilizados para treinamento e avaliação.

Um ponto a ser comentado, neste trabalho não foram utilizados grafos variantes no tempo, escopo do problema abordado. Em seu lugar, grafos estáticos foram alterados de forma a simular o comportamento variante no tempo, procedimento apresentado na subseção a seguir.

4.1. Criação do conjunto de treinamento com grafos estáticos

De maneira simples, a conversão de grafos estáticos em "grafos variantes no tempo" foi feita alterando a rede através da remoção de algumas arestas, sendo estas consideradas as arestas a serem adicionadas no intervalo de tempo futuro analisado.

Seguindo esta ideia, o procedimento consistiu em selecionar aleatoriamente um subconjunto de arestas do grafo e removê-las, mantendo apenas a informação da existência das mesmas. A partir deste estado, o grafo passa a ser considerado um grafo variante no tempo, tendo o conjunto de arestas removidas do grafo, inseridas em uma lista de pares de nós a entrar na base de treinamento e sendo estes marcados como pares de nós que receberão no futuro uma conexão. À mesma lista se juntam pares de nós sem aresta (marcados como pares de nós que não receberão conexão) até que esta possua como tamanho a porcentagem determinada do número de arestas inicial (10%), com cada classe tendo aproximadamente o mesmo número de amostras. Por fim, em posse da lista de pares de vértices que irão compor a base de treinamento, é feito o cômputo das métricas verificadas, dando origem à base de treinamento.

Um fator a ser comentado, apesar da remoção ocorrer aleatoriamente, busca-se preservar a conectividade do grafo no processo de construção destes grafos variantes no tempo. Desta forma, em grafos que se apresentam inicialmente conexos, repete-se o sorteio para cada escolha de aresta que resulte na quebra da componente conexa.

5. Experimentos

Para realização dos experimentos foram utilizados inicialmente grafos artificiais criados segundo os modelos Gnp, SmallWorld e Preferential Attachment. A parametrização utilizada para cada modelo pode ser vista a seguir:

- GNP: 10.000 vértices e probabilidade de conexão 0.001;
- Small World: 10.000 vértices, 10 conexões entre vizinhos e probabilidade de reconexão 0.1;
- Preferential Attachment: 10.000 vértices e 5 arestas por inserção de vértice

Os grafos resultantes destes modelos são conexos, com 10.000 vértices e aproximadamente 50.000 arestas. Os parâmetros dos modelos Small World e Preferential Attachment foram definidos buscando que os grafos tenham um número de arestas semelhante ao obtido no Gnp. Ainda, a probabilidade de reconexão do modelo Small World tem como objetivo acentuar suas características, via maximização do índice *smallworldness* [4].

Estes grafos são utilizados para identificação do subconjunto de atributos de destaque na tarefa de predição de *links*, tendo em vista que cada um destes modelos apresenta características diversas e, neste sentido, a análise destes modelos pode representar uma generalização razoável do comportamento esperado para outros grafos.

5.1. Analise dos atributos em grafos artificiais

Para avaliação de cada atributo, segundo os algoritmos de classificação, foi utilizado o módulo *classifierBasedAttributeSelection* presente na ferramenta Weka, com avaliação realizada via quantificação do impacto individual dos atributos. De maneira mais detalhada, a avaliação do conjunto de atributos ocorre com as técnica de aprendizado sendo submetidas as etapas de treinamento e avaliação com a utilização de apenas um atributo por vez, procedimento repetido para cada um dos atributos. De posse dos resultados desta sequência de experimentos, o ganho atribuído a cada atributo é dada pela margem de performance da classificação superior a um *random guest*

(classificação por sorteio), sendo a área sobre a curva (AUC) utilizada como métrica de desempenho na classificação e tendo como metodologia de verificação dos resultados a validação cruzada via modelo 5-fold.

A Tabela 1 apresenta o somatório do ganho dos atributos em cada uma das técnicas de aprendizado, categorizado por modelo de grafo, com a coluna Total mostrando o somatório do desempenho nos três modelos. A tabela é ordenada pelo ganho apresentado na coluna Total, estando em negrito os 5 atributos de maior valor em cada modelo.

Tabela 1: Ganho dos atributos na classificação por modelo de grafo

Atributos	GNP (Ganho)	Small World (Ganho)	Prefferential Attachment (Ganho)	Total
short_path	0.009500	**1.791648**	0.368998	**2.170146**
common_neighbors	-0.000242	**1.799350**	0.122241	**1.921349**
degree_H	**0.033249**	**0.832725**	**0.706378**	**1.572353**
degree_L	0.016190	**0.794484**	**0.609749**	**1.420423**
eigenCentrality_H	**0.018104**	**0.696613**	**0.499912**	**1.214629**
closeness_centrality_H	0.009479	0.248539	**0.592475**	0.850493
betweenness_centrality_H	0.009790	0.096285	**0.511478**	0.617552
clustering_H	-0.005562	0.380655	0.229543	0.604635
eigenCentrality_L	**0.030412**	0.322793	0.186595	0.539801
eccentricity_L	0.000281	0.101960	0.425873	0.528114
clustering_L	-0.002117	0.297300	0.091714	0.386897
betweenness_centrality_L	0.012696	0.147593	0.149654	0.309943
closeness_centrality_L	**0.022215**	0.017742	0.227030	0.266988
eccentricity_H	**0.017172**	0.016663	0.023986	0.057821

Observando os dados apresentados na Tabela 1 nota-se que no modelo Gnp as contribuições estão pulverizadas em diversos atributos sem destaque para nenhum fator específico. Por outro lado, no modelo Small World tem-se um conjunto bem definido de atributos com destaque, podendo ser citados *short_path* e *common_neighbors* com boa contribuição em todas as técnicas de classificação, além dos atributos envolvendo grau *degree_H* e *degree_L* e do atributo *eigenCentrality_H*. Da mesma forma, o modelo Prefferential Attachment apresenta alguns atributos com maior destaque, podendo ser citados principalmente aqueles relacionados ao grau, *degree_H* e *degree_L*, mas também os atributos de centralidade, *closeness_centrality_H*, *betweenness_centrality_H* e *eigencentrality_H*, que possuem um bom desempenho.

Em uma análise contextualizada, é interessante notar a intersecção dos atributos em destaque e a motivação na construção de cada um dos modelos. Para o GNP, de fato era esperado certa dificuldade na identificação de algum padrão na rede, dado caráter aleatório das conexões presentes no grafo, o que se reflete em nenhum atributo possuir um valor de ganho relevante, estando todos abaixo de 0.05.

No modelo Small World, os atributos relacionados a identificação de elementos pertencentes ao mesmo cluster se destacam, nominalmente *common_neighbors* e *short_path* (que provavelmente identifica a existência de caminhos mais curtos que a média em pares pertencentes ao mesmo cluster), dado que os vértices internos a um mesmo cluster têm, neste modelo, maiores chances de realizar conexões entre si num instante futuro. Por outro lado, o relativo destaque dos atributos envolvendo grau, *degree_H* e *degree_L* não é imediatamente explicável dado que o grau atribuído aos vértices se aproxima de uma distribuição normal, mas pode indicar que um número

menor de conexões em um vértice é visto como um forte indicativo de ausência de alguma conexão esperada do vértice em questão. Em relação ao atributo *eigenCentrality_H* a interpretação parece ser semelhante, há uma baixa variação nos valores de centralidade e os nós que, eventualmente, apresentem um número menor de conexões pela ausência de alguma conexão futura esperada, acabam compondo pares de provável conexão.

Para o modelo Prefferential Attachment a identificação do grau é um fator importante, coincidindo com os atributos de maior destaque, *degree_H* e *degree_L*. De fato, vértices com maior grau têm um maior número de conexões no modelo, sendo esperado que novas conexões venham a surgir justamente entre estes vértices e os demais, especialmente se estes "demais" também apresentarem grau elevado. O mesmo pode ser dito quando se observa a contribuição das medidas de centralidade, *closeness_centrality_H*, *betweenness_centrality_H* e *eigencentrality_H*, dado que vértices mais centrais tendem a ser vértices de grau mais elevado neste modelo.

Ainda, ao verificar os destaque de cada modelo e comparando-os com a sequência dos atributos de maior ganho na coluna Total, verifica-se uma boa representatividade dos modelos Small World e Prefferential Attachment, tendo os dois melhores atributos de ambos, *common_neighbors* e *short_path* para o Small World e *degree_H* e *degree_L* para o Prefferential Attachment, fazendo parte dos 5 de maior somatório de ganho. Além destes, o outro atributo que se destaca é o *eigenCentrality_H*, que se apresenta entre os de maior destaque em todos os três modelos.

Em um segundo experimento, foi verificada a acurácia da tarefa de classificação em cada uma das técnicas de aprendizado, considerando todos os atributos, apenas os 5 melhores identificados em cada grafo (denominado Top-5 do grafo) e os 5 melhores verificados no somatório dos três modelos (denominado Top-5 gerais). Novamente a avaliação foi feita através de validação cruzada via modelo 5-fold.

As Tabelas 2, 3 e 4 apresentam a acurácia obtida na tarefa de classificação ao se utilizar cada uma das técnicas de aprendizado, com respectivamente todos os atributos, apenas os Top-5 do grafo e os Top-5 gerais.

Tabela 2: Acurácia do processo de classificação utilizando todos os atributos

Classificadores	Gnp (%)	Small World (%)	Preferential Attachment (%)
SVM	50.54	95.14	64.52
Naive Bayes	50.90	94.90	61.34
Árvore de Decisão	50.30	95.12	70.50
Random Forest	50.80	95.12	69.92

Tabela 3: Acurácia do processo de classificação utilizando os melhores atributos identificados para o grafo

Classificadores	Gnp (%)	Small World (%)	Preferential Attachment (%)
SVM	50.92	94.98	65.84
Naive Bayes	51.50	94.94	60.94
Árvore de Decisão	50.54	95.12	72.02
Random Forest	50.70	93.52	68.54

Tabela 4: Acurácia do processo de classificação utilizando os melhores atributos dentre todos os modelos

Classificadores	Gnp (%)	Small World (%)	Preferential Attachment (%)
SVM	50.40	94.98	63.24
Naive Bayes	51.52	94.94	60.72
Árvore de Decisão	49.78	95.12	72.08
Random Forest	50.04	93.52	67.62

Analisando as tabelas anteriores, é interessante notar que a utilização de subconjuntos de atributos, tanto relacionados aos melhores para o grafo, quanto melhores em geral, não diminuiu em grande medida a acurácia do processo de classificação, o que vai de encontro ao objetivo proposto neste trabalho. Além disto, a qualidade na tarefa de classificação para cada um dos grafos se apresenta em linha com o que poderia ser previsto: para o GNP, o desempenho não é melhor ao de um *random guest* dado a falta de padrão nas conexões do grafo; no Small World, a qualidade de classificação é alta (superior a 90%), o que é esperado até certo ponto dado as características do modelo com as conexões padronizadas, mas que também, muito provavelmente, se deve ao procedimento de construção do grafo, originado de uma látice regular; e no modelo Prefferential Attachment o desempenho é regular, atingindo até 70%, o que corrobora com a premissa do grafo apresentar certa estrutura, apesar do caráter aleatório no processo de estabelecimento das conexões.

Em relação ao desempenho das técnicas, não era objetivo verificar qual delas tem um melhor desempenho para o problema. Apesar disso, é interessante notar que a Árvore de Decisão, teoricamente o algoritmo mais simples, foi o que apresentou a melhor qualidade no processo de classificação. Uma hipótese para o ocorrido é que, sendo a Árvore de Decisão uma técnica mais simples, esta pode ser menos sensível a uma má escolha de parâmetros quando comparada aos demais algoritmos de aprendizado, o que a faria possuir certa vantagem na utilização dos parâmetros padrão da ferramenta.

5.2. Acurácia da classificação em grafos reais

Em outra sequência de experimentos, visando validar os resultados anteriores, foi repetida a análise em grafos vindos de aplicações reais. Neste sentido, foram utilizados o grafo *newmovies* [7], relativo a um conjunto de filmes conectados pela participação de atores em comum, e o grafo *coauthor* [8], relativo a um conjunto de autores conectados pelos relações de coautoria em diferentes artigos. Ambos os grafos são desconexos, tendo a base *newmovies* com 26.851 vértices e 122.195 arestas e a base *coauthor* com 39.532 vértices 117.801 arestas. Ainda, verificando as características topológicas pode-se observar em ambos uma intersecção das características dos modelos Prefferential Attachment e Small World, com a distribuição de grau seguindo uma lei de potência e o valor de *smallworldness* elevado (*newmovies* com valor 1800 e *coauthor* com valor 3800).

Seguindo o mesmo procedimento, foi feita inicialmente a verificação do ganho de cada atributo no processo de classificação destes grafos, de forma a se construir um Top-5. Como resultado, os 5 melhores atributos obtidos para o grafo *newmovies* foram: *degree_H*, *closeness_centrality_H*, *common_neighbors*, *degree_L* e *closeness_centrality_L*, enquanto para a base *coauthor* identificou-se os atributos: *common_neighbors*, *short_path*, *closeness_centrality_L*, *degree_H* e *degree_L*, com ambas as listagens ordenados pelo ganho.

É interessante notar a sobreposição entre os Top-5 atributos destes grafos e o Top-5 geral determinado anteriormente, tendo o grafo *newmovies* uma similaridade via coeficiente de Jaccard de valor 0.43 ($J = \frac{|A \cap B|}{|A \cup B|}$) com seu Top-5 e o grafo *coauthor* similaridade de 0.66. Ainda, tendo em vista as características observadas nos grafos tem-se os atributos encontrados em linha com o que poderia ser esperado, aparecendo em ambos os grafos as métricas envolvendo grau, fator em

destaque no modelo Prefferential Attachment, e o atributo *common_neighbors*, um dos destaques no modelo Small World.

As Tabelas 5, 6 e 7 apresentam a acurácia obtida na tarefa de classificação ao se utilizar cada uma das técnicas de aprendizado, com respectivamente todos os atributos, apenas o Top-5 do grafo em questão e o Top-5 geral identificado anteriormente.

Tabela 5: Acurácia do processo de classificação utilizando todos os atributos - Grafos de aplicações reais

Classificadores	*newmovies* (%)	*coauthor* (%)
SVM	79.54	97.08
Naive Bayes	72.04	90.11
Árvore de Decisão	83.72	98.64
Random Forest	84.56	98.79

Tabela 6: Acurácia do processo de classificação utilizando os melhores atributos identificados para o grafo - Grafos de aplicações reais

Classificadores	*newmovies* (%)	*coauthor* (%)
SVM	79.60	95.71
Naive Bayes	79.68	96.37
Árvore de Decisão	80.80	98.58
Random Forest	79.53	98.54

Tabela 7: Acurácia do processo de classificação utilizando os melhores atributos dentre todos os modelos - Grafos de aplicações reais

Classificadores	*newmovies* (%)	*coauthor* (%)
SVM	79.39	94.78
Naive Bayes	79.02	97.29
Árvore de Decisão	84.05	98.66
Random Forest	80.82	98.57

Verificando o desempenho obtido, tem-se um cenário semelhante ao encontrado nos experimentos anteriores, sem grande perda na qualidade de classificação quando se utiliza apenas os subconjuntos de atributos (Top-5 do grafo e Top-5 geral).

Em relação a percentual de acertos, pode-se considerar um número relativamente alto, especialmente na base *coauthor*, o que é razoável considerando que as conexões em aplicações reais são muito mais regulares do que poderia ser encontrado em modelos de conexões aleatórios, se aproximando do conceito apresentado do Small World, o que inclusive se reflete no índice *smallworldness* verificado.

6. Conclusão e Trabalhos Futuros

Ao longo deste trabalho foi apresentada uma análise de algumas métricas topológicas básicas de grafos, quando aplicadas ao problema de predição de *links* via técnicas de aprendizados supervisionado.

Neste sentido, através de uma metodologia de avaliação de desempenho destas métricas e considerando um grupo de algoritmos de aprendizado, foi possível estabelecer um subconjunto de atributos suficientemente significativo (denominado Top-5 geral) capaz de manter a capacidade de predição quando comparado à classificação utilizando o conjunto total de atributos, com testes realizados em diferentes tipos de grafos.

Como análise adicional, foi possível observar nos grafos artificiais as intersecções entre os fatores motivadores de criação de cada modelo e os atributos identificados com maior contribuição na tarefa de classificação, o que corrobora a analise apresentada. Ainda, como poderia ser esperado, foi notado que os padrões de conexão dos grafos têm forte papel na qualidade de classificação que será obtida, tendo para uma geração completamente aleatória, por exemplo, capacidade mínima de predição das novas conexões.

A análise ainda foi corroborada pelo experimento realizado em grafos reais, tendo a classificação bom desempenho no subconjunto de atributos definido se aproximando do obtido com a utilização de todos os atributos. Ainda, foi observado que os atributos de maior destaque nestes grafos se assemelham ao Top-5 geral, o que reforça a escolha de tais atributos.

Como trabalhos futuros, uma possibilidade é a analise de outros métricas desenvolvidas especificamente para a predição de *links*, verificando o ganho na classificação que estas eventualmente possam proporcionar. Outra possibilidade é verificar se tais observações se mantêm em grafos que tenham, de fato, comportamento variante no tempo, permitindo, inclusive, validar a aproximação realizada via grafos estáticos.

Referências

[1] Al Hasan, M., Chaoji, V., Salem, S., Zaki, M. (2006). Link prediction using supervised learning. *In Proc. of SDM 06 workshop on Link Analysis, Counterterrorism and Security.*

[2] Benchettara, N., Kanawati, R., Rouveirol, C. (2010). Supervised machine learning applied to link prediction in bipartite social networks. *Proceedings - 2010 International Conference on Advances in Social Network Analysis and Mining, ASONAM 2010*

[3] Cukierski, W., Hamner, B., Yang, B. (2011). Graph-based features for supervised link prediction. *Proceedings of the International Joint Conference on Neural Networks*

[4] Humphries M. D., Gurney K. (2008). Network "Small-World-Ness": A Quantitative Method for Determining Canonical Network Equivalence. *Sporns O, ed. PLoS ONE*

[5] Martinez, V., Berzal, F., Cubero, J. (2016). A Survey of Link Prediction in Complex Networks. *ACM Computing Surveys*

[6] de Sa, H. R., Prudencio, R. B. C. (2011). Supervised link prediction in weighted networks. *In The 2011 International Joint Conference on Neural Networks*

[7] Tang, J., Sun, J., Wang, C., Yang, Z. (2009). Social Influence Analysis in Large-scale Networks. *In Proceedings of the Fifteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*

[8] Wang, L., Lou, T., Tang, J., Hopcroft, J. (2011). Detecting Community Kernels in Large Social Networks. *Proceedings of 2011 IEEE International Conference on Data Mining.*

[9] Witten, I. H., Frank, E. (2005). Data Mining: Practical Machine Learning Tools and Techniques. *San Francisco, CA, USA: Morgan Kaufmann Publishers Inc*

Apêndice

As Tabelas 8, 9 e 10 contém o resultado integral dos experimentos nos grafos relativos aos modelos GNP, Small World e Prefferential Attachment, respectivamente, discriminados pelo ganho apresentado em cada uma das técnicas de aprendizagem de máquina. Estes mesmos dados podem ser vistos para os grafos de aplicações reais *newmovies* e *coauthor* nas Tabelas 11 e 12.

Tabela 8: Ganho na classificação por atributo: Modelo Gnp

SVM		Naive Bayes	
Atributo	Ganho	Atributo	Ganho
closeness_centrality_H	0.012490	closeness_centrality_H	0.012608
degree_H	0.010891	closeness_centrality_L	0.009512
betweenness_centrality_H	0.009792	eigenCentrality_L	0.009291
closeness_centrality_L	0.009492	degree_H	0.008954
eigenCentrality_L	0.009492	eigenCentrality_H	0.008235
short_path	0.009093	betweenness_centrality_L	0.006871
eigenCentrality_H	0.008193	degree_L	0.005479
betweenness_centrality_L	0.006894	eccentricity_H	0.004033
degree_L	0.006795	betweenness_centrality_H	0.003130
eccentricity_H	0.004796	clustering_H	-0.000085
clustering_H	0.001799	common_neighbors	-0.000315
eccentricity_L	0.000000	clustering_L	-0.000380
clustering_L	-0.000100	eccentricity_L	-0.000719
common_neighbors	-0.000100	short_path	-0.001606
Árvore de decisão		Random Forest	
Atributo	Ganho	Atributo	Ganho
eccentricity_H	0.004030	degree_H	0.011981
short_path	0.001451	eigenCentrality_L	0.011629
degree_H	0.001423	eccentricity_H	0.004313
betweenness_centrality_L	0.000799	degree_L	0.003696
degree_L	0.000221	closeness_centrality_L	0.003211
clustering_H	0.000000	eigenCentrality_H	0.001675
clustering_L	0.000000	eccentricity_L	0.000999
closeness_centrality_H	0.000000	short_path	0.000563
closeness_centrality_L	0.000000	common_neighbors	0.000473
eccentricity_L	0.000000	clustering_L	-0.001637
eigenCentrality_H	0.000000	betweenness_centrality_L	-0.001869
eigenCentrality_L	0.000000	betweenness_centrality_H	-0.002493
common_neighbors	-0.000300	clustering_H	-0.007276
betweenness_centrality_H	-0.000640	closeness_centrality_H	-0.015619

Tabela 9: Ganho na classificação por atributo: Modelo Small World

SVM		Naive Bayes	
Atributo	Ganho	Atributo	Ganho
common_neighbors	0.451200	common_neighbors	0.449608
short_path	0.446000	short_path	0.447826
degree_L	0.179800	degree_H	0.233032
degree_H	0.174840	eigenCentrality_H	0.230362
eigenCentrality_H	0.163800	degree_L	0.223617
eigenCentrality_L	0.078500	eigenCentrality_L	0.109183
closeness_centrality_H	0.066000	closeness_centrality_H	0.085119
betweenness_centrality_L	0.042100	betweenness_centrality_L	0.073227
betweenness_centrality_H	0.036400	betweenness_centrality_H	0.047358
eccentricity_L	0.028200	clustering_L	0.028522
closeness_centrality_L	0.012600	eccentricity_L	0.025631
clustering_L	0.000300	clustering_H	0.023449
eccentricity_H	-0.002200	closeness_centrality_L	0.015150
clustering_H	-0.002900	eccentricity_H	0.006474

Árvore de decisão		Random Forest	
Atributo	Ganho	Atributo	Ganho
short_path	0.448856	common_neighbors	0.449686
common_neighbors	0.448856	short_path	0.448967
degree_H	0.192898	degree_H	0.231955
eigenCentrality_H	0.175409	degree_L	0.220153
degree_L	0.170914	clustering_H	0.204237
clustering_H	0.155869	clustering_L	0.179905
clustering_L	0.088572	eigenCentrality_H	0.127041
eigenCentrality_L	0.085624	eigenCentrality_L	0.049486
closeness_centrality_H	0.062378	closeness_centrality_H	0.035042
eccentricity_L	0.022298	betweenness_centrality_L	0.029286
eccentricity_H	0.005185	eccentricity_L	0.025832
betweenness_centrality_H	0.004893	betweenness_centrality_H	0.007634
betweenness_centrality_L	0.002980	eccentricity_H	0.007204
closeness_centrality_L	0.000000	closeness_centrality_L	-0.010008

Tabela 10: Ganho na classificação por atributo: Modelo Prefferential Attachment

SVM		Naive Bayes	
Atributo	Ganho	Atributo	Ganho
closeness_centrality_H	0.141915	degree_L	0.186785
degree_H	0.126841	degree_H	0.174572
eccentricity_L	0.112036	closeness_centrality_H	0.173927
eigenCentrality_H	0.098333	eigenCentrality_H	0.141664
short_path	0.092972	betweenness_centrality_H	0.129032
betweenness_centrality_H	0.070726	eccentricity_L	0.104558
closeness_centrality_L	0.057182	short_path	0.096761
eigenCentrality_L	0.031807	closeness_centrality_L	0.079197
degree_L	0.028510	eigenCentrality_L	0.077881
common_neighbors	0.027567	betweenness_centrality_L	0.074862
betweenness_centrality_L	0.005502	common_neighbors	0.032371
clustering_L	-0.000280	clustering_L	0.017898
eccentricity_H	-0.007692	eccentricity_H	0.010347
clustering_H	-0.055170	clustering_H	-0.016179

Árvore de decisão		Random Forest	
Atributo	Ganho	Atributo	Ganho
degree_L	0.186749	degree_H	0.230376
degree_H	0.174589	degree_L	0.207705
betweenness_centrality_H	0.155192	betweenness_centrality_H	0.156528
clustering_H	0.147871	clustering_H	0.153020
closeness_centrality_H	0.140636	closeness_centrality_H	0.135996
eigenCentrality_H	0.134586	eigenCentrality_H	0.125329
eccentricity_L	0.104774	eccentricity_L	0.104505
short_path	0.085721	short_path	0.093544
closeness_centrality_L	0.059202	betweenness_centrality_L	0.039908
eigenCentrality_L	0.055411	clustering_L	0.038126
clustering_L	0.035970	common_neighbors	0.031461
common_neighbors	0.030842	closeness_centrality_L	0.031450
betweenness_centrality_L	0.029381	eigenCentrality_L	0.021497
eccentricity_H	0.009879	eccentricity_H	0.011451

Tabela 11: Ganho na classificação por atributo: Grafo *newmovies*

SVM		Naive Bayes	
Atributo	Ganho	Atributo	Ganho
closeness_centrality_H	0.210639	closeness_centrality_H	0.272882
common_neighbors	0.195853	closeness_centrality_L	0.261760
closeness_centrality_L	0.191981	degree_H	0.243986
degree_L	0.189718	common_neighbors	0.243893
degree_H	0.177640	degree_L	0.216327
eigenCentrality_L	0.152175	eigenCentrality_L	0.208455
eigenCentrality_H	0.125526	betweenness_centrality_H	0.194463
eccentricity_H	0.124239	eccentricity_H	0.190305
betweenness_centrality_L	0.088679	clustering_L	0.181371
eccentricity_L	0.085303	betweenness_centrality_L	0.164818
clustering_H	0.071682	clustering_H	0.153285
betweenness_centrality_H	0.018160	eigenCentrality_H	0.131151
short_path	0.016817	short_path	0.016256
clustering_L	-0.068597	eccentricity_L	0.009592
Árvore de decisão		Random Forest	
Atributo	Ganho	Atributo	Ganho
short_path	0.330821	short_path	0.385669
degree_H	0.256562	degree_H	0.318779
common_neighbors	0.240937	degree_L	0.277695
eigenCentrality_H	0.237087	common_neighbors	0.242344
closeness_centrality_H	0.232946	betweenness_centrality_H	0.231363
clustering_L	0.226616	clustering_L	0.230577
degree_L	0.223038	closeness_centrality_H	0.226603
closeness_centrality_L	0.202841	eigenCentrality_H	0.218933
clustering_H	0.191306	clustering_H	0.210598
eccentricity_L	0.173168	eigenCentrality_L	0.195458
betweenness_centrality_H	0.143945	eccentricity_H	0.193035
eigenCentrality_L	0.133190	closeness_centrality_L	0.192616
eccentricity_H	0.131841	eccentricity_L	0.183290
betweenness_centrality_L	0.028327	betweenness_centrality_L	0.177400

Tabela 12: Ganho na classificação por atributo: Grafo *coauthor*

SVM		Naive Bayes	
Atributo	Ganho	Atributo	Ganho
common_neighbors	0.484243	common_neighbors	0.484514
short_path	0.206661	closeness_centrality_L	0.246237
degree_H	0.171019	degree_H	0.217892
degree_L	0.170872	degree_L	0.210638
eccentricity_L	0.149058	short_path	0.206722
closeness_centrality_L	0.148760	clustering_H	0.156544
clustering_H	0.116346	betweenness_centrality_H	0.154756
betweenness_centrality_H	0.090306	closeness_centrality_H	0.148496
clustering_L	0.043471	eccentricity_L	0.131260
eccentricity_H	0.031139	clustering_L	0.122893
betweenness_centrality_L	0.013691	eccentricity_H	0.102408
eigenCentrality_L	0.003865	betweenness_centrality_L	0.095905
eigenCentrality_H	0.002750	eigenCentrality_L	0.002959
closeness_centrality_H	-0.030512	eigenCentrality_H	0.001730

Árvore de decisão		Random Forest	
Atributo	Ganho	Atributo	Ganho
short_path	0.485821	short_path	0.491786
common_neighbors	0.483355	common_neighbors	0.483339
closeness_centrality_L	0.219178	degree_H	0.235618
clustering_L	0.199306	degree_L	0.228272
degree_H	0.186917	closeness_centrality_L	0.223492
degree_L	0.176426	eccentricity_L	0.216982
clustering_H	0.149710	clustering_L	0.194938
eccentricity_L	0.145511	betweenness_centrality_H	0.163890
closeness_centrality_H	0.140661	clustering_H	0.163803
eccentricity_H	0.120889	closeness_centrality_H	0.157266
betweenness_centrality_H	0.088434	betweenness_centrality_L	0.147460
betweenness_centrality_L	0.020614	eccentricity_H	0.144746
eigenCentrality_L	0.002959	eigenCentrality_L	0.135167
eigenCentrality_H	0.001524	eigenCentrality_H	0.108922