Caracterização de Classes via Otimização em Redes Complexas

Lilian Berton[1], Liang Zhao[2]

[1,2]Instituto de Ciências Matemáticas e de Computação (ICMC) – Universidade de São Paulo (USP)
Caixa Postal 668 – 13560-970 – São Carlos – SP – Brasil

{lberton,zhao}@icmc.usp.br

Abstract. *Complex networks have emerged as an important way of data representation and abstraction able of capturing the topological structure presented in databases. This work proposes a method for building a network from a vector based dataset. It is based on the optimization of an energy function that considers purity and extension measures of the network. The constructed network was used to characterize mixing level among data classes in classification problem. Class characterization is an important issue, but it is not well studied. Therefore, we consider this work a contribution in this direction.*

Resumo. *As redes complexas surgiram como uma importante maneira de representação e abstração de dados capaz de capturar as relações topológicas presentes em bases de dados. Este trabalho propõe um método para construção de rede a partir de uma base de dados vetorial, o qual é baseado na otimização de uma função de energia que considera medidas de pureza e extensão da rede. A rede construída foi utilizada para caracterizar mistura entre classes de dados em problema de classificação de dados. A caracterização de classes é uma questão importante na classificação de dados, porém ainda é pouco estudada. Desta forma, consideramos este trabalho uma contribuição nesta direção.*

1. Introdução

Nos últimos anos as redes complexas têm se destacado fazendo com que o interesse nesta área aumentasse. As redes complexas são grafos de grande escala que possuem padrões de conexão não trivial. Redes como a Internet [Faloutsos et al. 1999], a *World Wide Web* [Albert et al. 1999], redes de interações sociais entre indivíduos [Scott 2000], redes neurais biológicas [Sponrs 2002], entre outras, têm sido estudadas sob essa abordagem. As redes complexas têm se mostrado uma maneira interessante para representação e abstração de dados, mudando a maneira com que vários sistemas interconectados são modelados [Bornholdt e Schuster 2003].

Tradicionalmente as redes complexas eram descritas de acordo com o modelo de Erdös e Rényi (1959), conhecido como Redes Randômicas. No entanto, Watts e Strogatz (1998) descobriram que a média de caminhos mais curtos em uma rede pode ser reduzida por alteração aleatória de poucas ligações em uma rede regular. A rede resultante foi denominada Rede de Pequeno Mundo. Posteriormente, Barabási e Albert (1999) descobriram que muitas redes reais têm distribuição de grau dos nós que obedece a lei de potência: $P(k) \sim k^{-\gamma}$, onde k é o número de conexões de um nó escolhido

aleatoriamente e γ é o expoente de escala, o que significa que existe um pequeno conjunto de nós que possui um grande número de ligações (*hubs*) e um grande número de nós com poucas ligações. Estas redes são denominadas Redes Livre de Escala.

Métodos de agrupamento de dados baseados em redes complexas, também conhecidos como detecção de comunidades, têm sido extensivamente explorados na literatura. Tais comunidades podem ser definidas como grupos de vértices da rede densamente conectados, enquanto que conexões entre vértices de grupos diferentes são esparsas [Newman e Girvan 2004]. Muitas técnicas foram desenvolvidas baseadas em diversas ideias para detecção de comunidades. Alguns exemplos podem ser encontrados em: [Zhou 2003a; Zhou 2003b; Newman 2004; Newman e Girvan 2004; Reichardt e Bornholdt 2004; Boccaletti et al. 2007; Quiles et al. 2008]. De maneira geral, a estrutura em comunidades revela similaridade por meio de conexões entre os vértices pertencentes a um mesmo grupo. Essas similaridades, por sua vez, podem revelar grupos nos dados em problemas de agrupamento e, de maneira análoga, podem evidenciar classes em problemas de classificação. Desse modo, as redes complexas também estão sendo usadas com sucesso na classificação de dados.

A classificação de dados lida com a detecção automática de padrões em conjuntos de dados. Por padrões entendem-se relações ou estruturas inerentes a alguns conjuntos de dados [Mitchell 1997]. Com a detecção de padrões significantes nos dados disponíveis (conjunto de treinamento), espera-se que um classificador possa realizar predições de classes em novos dados de entrada. Existem muitos problemas importantes que podem ser resolvidos utilizando-se dessa abordagem, abrangendo diversas áreas como bioinformática [Baldi e Brunak 1998], mineração de dados [Cook e Holder 2000], reconhecimento de escrita [LeCun et al. 1989], entre outras.

O desempenho de um classificador depende de diferentes fatores. Um fator muito importante se refere às características dos dados a serem classificados. Na prática nenhum classificador é o melhor em todos os problemas dados, fenômeno que pode ser explicado pelo teorema *no free lunch* [Worlpert e Macready 1997]. Como nem sempre se tem conhecimento prévio sobre a distribuição de um conjunto de dados ou sobre as características das classes, neste trabalho propõe-se um método que auxilie na caracterização de classes de dados. Para isso, uma rede é construída a partir de uma base de dados vetorial, sendo a mesma o resultado do processo de otimização de uma função de energia que considera medidas de pureza e extensão da rede. Para validar a proposta, executou-se o algoritmo em alguns cenários de dados artificiais e reais, buscando analisar a mistura nas classes de dados.

A seguir, na Seção 2 são apresentados alguns trabalhos relacionados. Na Seção 3 é descrito o método proposto, apresentando a função de energia utilizada, as medidas de extensão e pureza e o algoritmo com detalhes da implementação. Na Seção 4 são mostrados os resultados obtidos a partir da aplicação do método em redes artificiais e reais, bem como análise dos mesmos. Por fim, na Seção 5 são apresentadas as principais conclusões do trabalho e perspectivas de trabalhos futuros.

2. Trabalhos relacionados

Esta Seção descreve alguns dos métodos utilizados para construção de redes a partir de dados, com enfoque para as redes *K*-associados e sobre um processo de otimização que inspirou o desenvolvimento deste trabalho.

Construção de redes a partir de dados no formato atributo-valor

A construção de redes a partir de dados é uma fase importante no aprendizado baseado em grafos, as abordagens mais conhecidas são baseadas em relações de vizinhança, como grafos *KNN* [Zhu 2005] e ϵ [Belkin e Niyougi 2003]. Nos grafos *KNN* um parâmetro *K* é usado para controlar o número de vértices vizinhos que serão utilizados nas conexões. Nos grafos ϵ, as conexões são baseadas num raio ϵ, de modo que um vértice v_i será conectado com um vértice v_j, se a distância $d(v_i, v_j) < \epsilon$. Outras abordagens incluem relações de vizinhança usando uma função gaussiana $f(x_i, x_j) = \exp\left(-\|x_i - x_j\|^2 / 2\sigma^2\right)$ na qual x_i representa um dado de entrada e σ controla o tamanho da vizinhança [Von Luxburg 2007].

Outra abordagem proposta por Lopes et al. (2009) pode ser considerada como um *KNN* adaptativo. A seguir esta técnica é descrita com mais detalhes.

Seja $k_{nn}(v_i)$ o conjunto dos *K*-vizinhos mais próximos do vértice v_i por uma medida de similaridade dada, observe que vértices em $k_{nn}(v_i)$ podem ter diferentes rótulos de classes. A vizinhança N_{K_i} para um vértice v_i é definida como seus *K*-vizinhos conectados como segue: $N_{K_i} = \{v_j \mid e_{ij} \in E \text{ e } v_j \in \text{classe}(v_i) \text{ e } v_j \in k_{nn}(v_i)\}$. O grau g_i é definido como o número de arestas para seus vizinhos N_{K_i} e isto é no máximo $2K_i$ (*K* arestas de v_i para seus vizinhos e *K* de seus vizinhos para v_i).

Seja g_i o grau do vértice v_i, *N* o número de dados no conjunto de treinamento (número de vértices), *K* um parâmetro para controlar o número de vizinhos usados na construção da rede. A fração $g_i/2K$ corresponde à conexão entre o vértice v_i e os vértices em seu próprio componente, note que cada classe pode possuir um ou mais componentes da rede e cada componente é um conjunto de vértices conectados. Essa proporção varia entre 0 e 1, inclusive. Em seguida, o total de conexões entre N_c vértices num componente *C* é dado pela Equação (1).

$$|E_c| = \frac{1}{2}\sum_{i=1}^{N_c} g_i = \frac{N_c}{2}\sum_{i=1}^{N_c}\frac{g_i}{N_c} = \frac{N_c}{2}\langle G_c\rangle \quad (1)$$

Onde $\langle G_c\rangle$ corresponde ao grau médio do componente *C*. O número máximo de arestas entre N_c vértices é KN_c desde que $K < N_c$. Com isso, a probabilidade de arestas entre vértices no mesmo componente *C* (componentes inter conectados) é dada pela Equação (2).

$$P_i = \frac{\frac{N_c\langle G_c\rangle}{2}}{KN_c} = \frac{\langle G_c\rangle}{2K} \quad (2)$$

Na Equação (2) $P_i = 1$ quando há somente vértices com o mesmo rótulo na *K*-vizinhança de todo componente de v_i. Desse modo, $\langle G_c\rangle/2K$ pode ser visto como uma medida de *pureza* na região do componente *C*.

No processo descrito até agora, cada *K* gera uma rede e certamente algumas redes terão componentes melhores do que outras, de acordo com a noção de pureza. Raramente uma rede obtida de um único *K* terá os melhores componentes entre todos os componentes em todas as *K* redes possíveis. A partir disso, o objetivo é obter uma rede com a melhor organização de todos os dados nos componentes independentemente de se ter um único *K*. Fazendo isto, a ideia é variar *K* mantendo os melhores componentes encontrados. Esse processo irá resultar em uma rede chamada rede ótima, com

componentes formados por valores distintos de *K*. A rede ótima é a estrutura final obtida por meio desse processo. Essa rede pode ser vista como o resultado do processo de aprendizagem supervisionado e será usada no classificador *K*-associado não-paramétrico proposto para classificar novos dados [Lopes et al. 2009].

Otimização em redes complexas

Cancho e Solé (2003) utilizam um algoritmo evolutivo envolvendo a minimização da quantidade de arestas e a média do menor caminho e encontram quatro principais tipos de redes: exponenciais, livres de escala, estrelas e altamente densas. A seguir o método é descrito com mais detalhes.

Em um tempo $t = 0$ tem-se um grafo aleatório com distribuição de grau de Poisson, no qual dois vértices são conectados com uma probabilidade p. A função de energia do algoritmo de otimização é definida como:

$$E(\lambda) = \lambda d + (1 - \lambda)\rho \tag{3}$$

na qual $0 \leq \lambda, d, \rho \leq 1$. λ é um parâmetro que controla a combinação linear de d e ρ. O número normalizado de arestas, ρ é definido em termos de a_{ij} como:

$$\rho = 1/\binom{n}{2} \sum_{i<j} a_{ij} \tag{4}$$

A distância vértice-vértice normalizada, d, é definida como $d = D/D^{linear}$, tal que D é a média da distância mínima entre os vértices e $D^{linear} = (n + 1)/3$ é o valor máximo de D que pode ser encontrado em uma rede conexa, considerando-se um grafo linear.

$$D = 1/\binom{n}{2} \sum_{i<j} D_{ij} \tag{5}$$

A minimização de $E(\lambda)$ envolve a minimização simultânea da distância e do número de arestas. Essas duas restrições incluem dois aspectos relevantes em uma rede, o custo das ligações físicas entre unidades e a velocidade de comunicação entre elas.

O algoritmo de minimização procede da seguinte maneira: No tempo $t = 0$, a rede é inicializada com uma densidade $\rho(0)$ seguindo a distribuição de grau de Poisson. Num tempo $t > 0$, o grafo é modificado alterando-se aleatoriamente a conexão entre alguns pares de vértices. Com uma probabilidade *v*, cada a_{ij} pode mudar de 0 para 1 ou de 1 para 0. A nova matriz de adjacência é aceita se $E(\lambda, t + 1) < E(\lambda, t)$. Caso contrário, diferentes alterações são realizadas e testadas novamente. O algoritmo é interrompido quando as modificações em $A(t)$ não são aceitas após *T* execuções. A minimização do algoritmo funciona como a técnica de *simulated annealing*.

O trabalho de Cancho e Solé (2003) mostra que a otimização tem um papel fundamental na formação e evolução das redes complexas. A Figura 1 sugere que quatro fases estão presentes no processo.

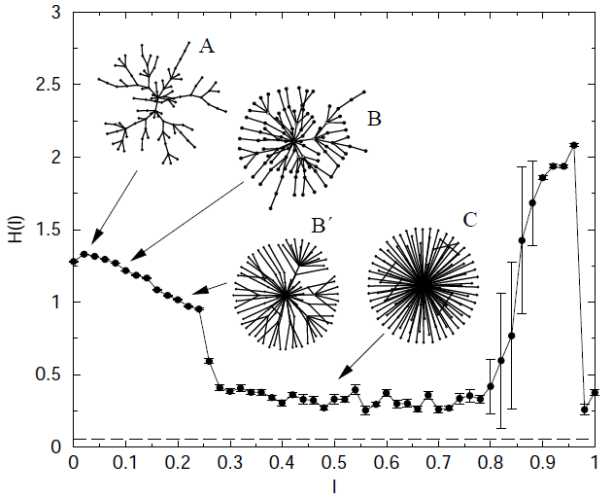


Figura 1: Redes ótimas para determinados valores de λ. Média sobre 50 redes otimizadas com n = 100, T = $\binom{n}{2}$***, v =*** $2/\binom{n}{2}$ ***e*** $\rho(0) = 0.2$***. A: uma rede exponencial com*** $\lambda = 0.01$***. B: uma rede livre de escala com*** $\lambda = 0.08$***. Hubs com múltiplas conexões e a dominância de vértices com apenas uma conexão podem ser vistos. C: uma rede estrela com*** $\lambda = 0.5$***. B': uma rede intermediária entre B e C com muitos hubs podem ser identificadas, (Cancho e Solé 2003).***

3. O Método Proposto

A motivação para este trabalho partiu das redes *K*-associados [Lopes et al. 2009]. Essas redes são construídas com base em uma medida de pureza e como esta medida não considera o tamanho dos componentes de maneira explícita, ela favorece a formação de muitos componentes pequenos quando se tem um nível alto de mistura nas classes. Uma boa classificação depende do equilíbrio entre esses dois fatores, por essa razão, buscou-se desenvolver uma técnica considerando não apenas o fator de *pureza*, mas também *extensões* de classes formadas.

No método proposto, uma rede é construída a partir de um conjunto de dados proposicionais representados no formato atributo-valor. A rede é construída com base nas relações de similaridade entre os vértices, de modo que cada vértice irá alterar suas conexões com um *k*-vizinho mais próximo que seja de sua mesma classe, somente se essa alteração maximize a *função de energia* aqui sugerida. A utilização dessa função de energia foi inspirada pelo trabalho de Cancho e Solé (2003), no qual se buscava a minimização da quantidade de arestas e da média do menor caminho.

Buscamos definir uma função de energia que considere a relação entre a pureza e a extensão da rede, visto que os dois fatores são opostos, pois o aumento da pureza tende a diminuir a extensão e vice-versa. A função de energia é representada pela Equação (6):

$$E = \lambda d + (1 - \lambda)p \tag{6}$$

na qual, d representa a extensão e p representa a pureza, tal que p e $d \in [0,1]$. A partir das medidas de pureza e extensão, investigamos o comportamento delas na rede alterando gradativamente o peso de cada uma na função de energia, para tal um parâmetro $\lambda \in [0,1]$ foi introduzido.

Para a proposta considerada, a medida de extensão é computada do seguinte modo: o caminho mínimo é calculado para *N* vértices de cada componente formado na rede. O valor máximo encontrado para cada componente *C* representa o diâmetro do mesmo. Podemos obter assim a média de diâmetro dos componentes:

$$\langle d \rangle = \left(\sum_{i=1}^{C} d_i\right)/C \tag{7}$$

Para que $\langle d \rangle$ permaneça no intervalo entre 0 e 1, é normalizado novamente pelo maior diâmetro possível de ser encontrado quando consideramos a maior classe como um grafo linear. Obtemos assim *d,* que representa a extensão da rede.

$$d = \langle d \rangle / d_{max} \tag{8}$$

A pureza de um vértice *i* se refere à relação entre o total de ligações que um vértice estabeleceu e o número máximo que ele poderia ter estabelecido. Seja g_i o grau de saída do vértice v_i, k um parâmetro para controlar o número de vizinhos usados na construção da rede. A Equação (9) corresponde a relação entre o número de conexões entre o vértice v_i e os vértices em seu próprio componente.

$$p_i = g_i/k \tag{9}$$

A medida *p* se refere a pureza de toda a rede, sendo a média de pureza de todos os vértices.

$$p = \left(\sum_{i=1}^{N} p_i\right)/N \tag{10}$$

Algoritmo 1

Entrada: Conjunto de vértices: $V = v_1, \ldots, v_n$
Conjunto de classes: $L = classe(v_1), \ldots, classe(v_n)$
Parâmetro $\lambda \in [0,1]$

Saída: Rede gerada: R

1) **Para** cada vértice v_i de V
Para cada vértice v_j de V
S = Calcula_similaridade (v_i, v_j);
S = Ordena_crescente (S);
R = V;

2) p = Calcula_pureza (R);
d = Calcula_extensão (R);
E = Calcula_Energia (p, d, λ);

3) **Para** *cont* = 1 até N
R_Δ = Modifica_rede (R, S, V, L);
p_Δ = Calcula_pureza (R_Δ);
d_Δ = Calcula_extensão (R_Δ);
E_Δ = Calcula_Energia $(p_\Delta, d_\Delta, \lambda)$;
Se $E_\Delta > E$
$E = E_\Delta$;
$R = R_\Delta$;

3.1) Modifica_rede (R, S, V, L)
v_i = *random*();
k = *random*();
Se $(k < n)$
v_k = Busca_vertice (S);
Se $\left(classe(v_i) == classe(v_k)\right)$
Se (!Existe_aresta(v_i, v_k))
$R_\Delta \leftarrow R_\Delta \cup$ Insere_aresta (V);
Senão
$R_\Delta \leftarrow R_\Delta -$ Remove_aresta (V);

4) **Retorna** R.

O Algoritmo 1 descreve com detalhes a implementação do método. A entrada do algoritmo é um conjunto de dados representados na forma atributo-valor, com a respectiva classe associada e um parâmetro λ.

No passo 1 o algoritmo calcula a partir dos dados de entrada a matriz de similaridade, nesse caso foi calculada a distância euclidiana entre todos os pares de vértices. Posteriormente cada linha dessa matriz é ordenada em ordem crescente. A rede R é inicializada, de modo que cada objeto de dado é representado em um vértice.

No passo 2 são calculadas as medidas de extensão da rede, Equação (8), de pureza, Equação (10), e a partir destas é realizado o cálculo da energia, Equação (6), para determinado valor de λ.

O passo 3 é repetido N vezes. Outra opção que poderia ser utilizada como critério de parada, seria executar o algoritmo até que N vezes o valor de E não sofra alteração.

O passo 3.1 *Modifica-rede* é executado retornando uma nova rede com modificações nas ligações para um vértice i. A partir da nova rede, as medidas de pureza e extensão são recalculadas e a função de energia é reavaliada para estas novas medidas. Caso o valor da nova função de energia seja maior que o anterior, a rede antiga é substituída pela nova rede que sofreu modificações.

No passo 3.1 um vértice i é escolhido aleatoriamente para sofrer alteração nas ligações. Também é escolhido aleatoriamente um vértice k, que esteja dentro do conjunto dos n-vizinhos mais próximos do vértice i, para se conectar/desconectar com este. Nos testes realizados, o vértice k é sorteado num intervalo entre 1 e 5.

Destaca-se que o vértice k é selecionado aleatoriamente, porém o 1-vizinho tem uma probabilidade maior de ser escolhido, que o 2-3...n-vizinho respectivamente. Do mesmo modo, o 2-vizinho tem uma probabilidade menor que o 1 de ser selecionado, porém sua probabilidade é maior que o 3-4...n-vizinho respectivamente. E assim por diante. O objetivo disto é fazer com que os vértices mais similares de um vértice i tenham preferência nas conexões.

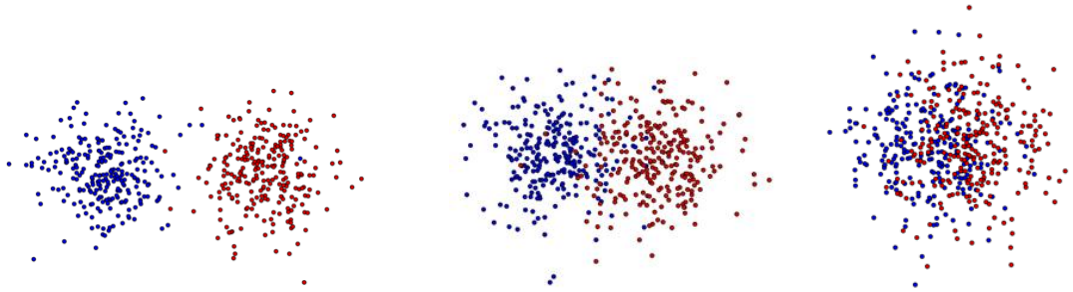
Caso o vértice k escolhido não esteja ligado com o vértice i, uma ligação entre eles será estabelecida se ambos pertencerem a mesma classe. Além disso, o vértice i irá estabelecer ligações com todos os k-vizinhos menores que k. Porém, se o vértice k já estiver conectado com o vértice i, esta conexão é removida, bem como todas as conexões existentes entre o vértice i e os vértices maiores que k.

Essa nova rede irá conter alterações nas ligações para um dado vértice i e será retornada para o Passo 3. Por fim, o passo 4 retorna a rede final, a qual apresenta a maximização da função de energia, para um dado valor de λ.

4. Simulações em Redes Artificiais e Reais

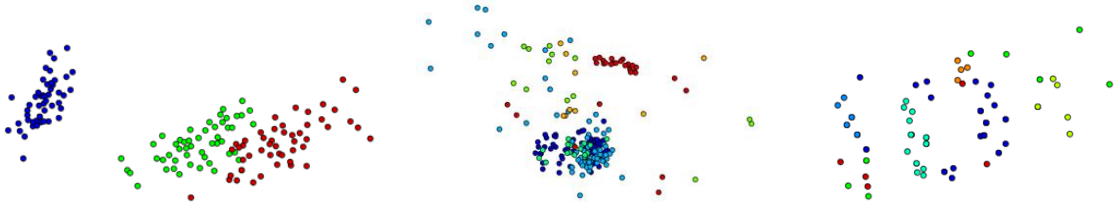
O algoritmo foi executado para um conjunto de dados artificiais e em três bases de dados reais a fim de analisar seu comportamento nestas bases. Testou-se para um conjunto de dados com distribuição gaussiana (Figura 2) de modo que, para cada conjunto de dados aumentava-se gradativamente a mistura dos elementos e nos

conjuntos de dados Iris, Glass e Zoo, obtidos no repositório UCI[1]. Para a geração das figuras foi utilizado o software PEx[2], com o método de *Sammon´s Mapping* para redução de dimensionalidade.



Gaussianas 1 *Gaussianas 2* *Gaussianas 3*

Figura 2: Base de dados *Gaussianas 1-2-3*. Cada figura representa dois conjuntos de dados com 250 elementos e distribuição gaussiana, a cor vermelha representa uma classe e a cor azul outra.



Iris *Glass* Zoo

Figura 3: Base de dados Iris, Glass e Zoo. Cada cor nos conjuntos de dados representa uma classe.

Na Figura 4 são mostradas as redes finais formadas para as bases *Gaussianas 1-2-3*, após a execução do *Algoritmo 1* para os valores de λ igual a 0 e 1. Nota-se que quando $\lambda = 0$ apenas a pureza está sendo levada em conta e quando $\lambda = 1$ apenas a extensão é considerada.

As redes *A* e *B* foram formadas a partir da base de dados *Gaussianas 1*. Nota-se que ambas as redes formaram poucos componentes, porém na rede *A* há mais conexões entre os vértices e na rede *B* o diâmetro dos componentes é maior que na rede *A*. As redes *C* e *D* foram formadas a partir da base de dados *Gaussianas 2*. Como o nível de mistura foi aumentado um número maior de componentes é formado, porém a rede C apresenta mais componentes que a rede *D*. As rede *E* e *F* foram formadas a partir da base de dados *Gaussianas 3* cujo nível de mistura está maior que das bases *Gaussianas 1* e *2*. Com isso, o número de componentes formados é bem alto quando $\lambda = 0$ (rede *E*), pois como cada vértice apresenta muitos vizinhos com classe diferente da sua, estabelece ligações com pouco deles para permanecer como uma pureza alta. Quando $\lambda = 1$, o número de componentes formados na rede *F* diminui com relação a rede *E*.

Analisando os resultados obtidos, nota-se que conforme a mistura entre dados de diferentes classes aumenta o número de componentes formados na rede também aumenta, tanto para $\lambda = 0$, como para $\lambda = 1$. Porém para $\lambda = 1$ a tendência é de formar menos componentes que para $\lambda = 0$.

[1] http://archive.ics.uci.edu/ml/
[2] http://infoserver.lcad.icmc.usp.br/infovis2/PExDownload

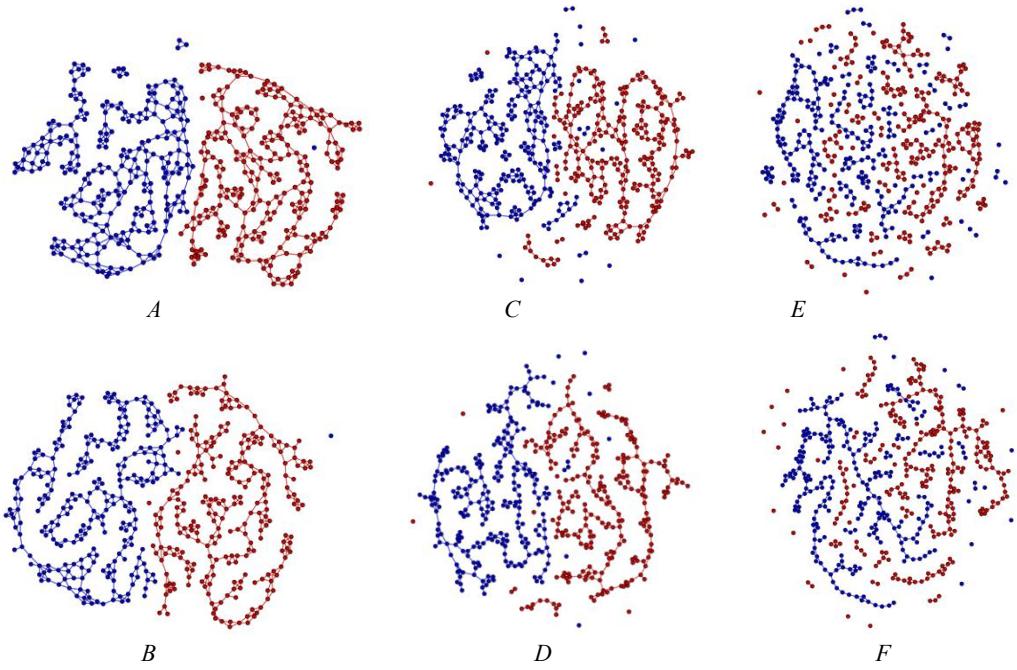


Figura 4: *A* e *B*: redes finais formadas para a base de dados *Gaussianas 1*. *C* e *D*: redes finais formadas para a base de dados *Gaussianas 2*. *E* e *F*: redes finais formadas para a base de dados *Gaussianas 3*. As redes *A*, *C* e *E* foram construídas com λ igual a 0 e as redes *B*, *D* e *F* foram construídas com λ igual a 1. Foram considerados *n* = 5 e *N* = 10000 no *Algoritmo 1*.

Na Figura 5 são mostradas as redes finais formadas para as bases de dados Iris, Glass e Zoo, após a execução do *Algoritmo 1* para λ igual a 1. Nota-se que na base Iris os componentes ficaram bem conexos, já que esta base apresenta menos mistura entre as classes, ao contrário da rede Glass e Zoo.

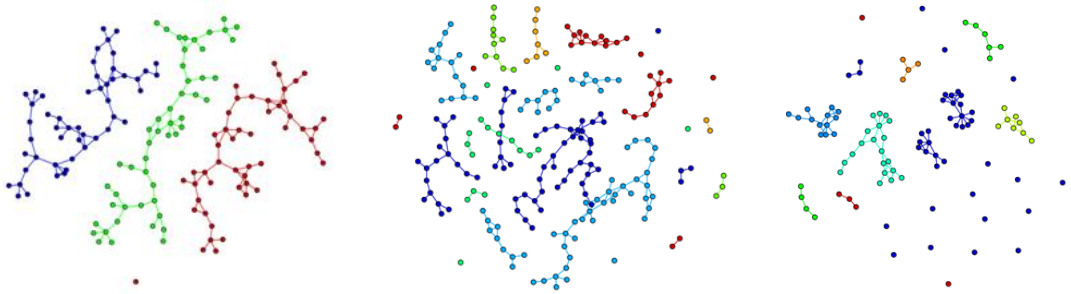


Figura 5: Redes finais formadas para as bases de dados Iris, Glass e Zoo. As redes foram construídas com λ igual a 1. Foram considerados *n* = 5 e *N* = 10000 no *Algoritmo 1*.

A Figura 6 mostra os valores obtidos para a pureza, extensão e energia das redes finais geradas a partir da base de dados *Gaussianas 1-2-3*, quando a função de energia é maximizada em alguns valores de λ.

Nota-se que o valor da pureza para a base *Gaussianas 3* é menor que da base *Gaussianas 2* e *Gaussianas 1* respectivamente. Isso porque a base *Gaussianas 3* apresenta alto nível de mistura entre os elementos de diferentes classes, com isso, componentes "menos puros" são formados. Ou seja, cada vértice estabelece ligações

com um número menor de vizinhos que estava sendo considerado no momento. Além disso, a pureza em cada base se mantém praticamente constante, diminuindo levemente conforme λ se aproxima de 1. Isso acontece, pois os vértices conseguem completar suas ligações tanto para um *k*-vizinho menor como para um *k*-vizinho maior, permanecendo com um valor alto de pureza.

Observa-se que o valor da extensão para a base *Gaussianas 3* é menor que da base *Gaussianas 2* e *Gaussianas 1* respectivamente, devido ao nível de mistura apresentado na base *Gaussianas 3*. Com isso, os vértices não conseguem formar um único componente para cada classe, diminuindo o valor para a extensão. Além disso, o valor de extensão aumenta conforme λ se aproxima de 1, já que esta medida passa a ter mais destaque no cômputo da função de energia.

A função de energia decai conforme λ se aproxima de 1 porque passa a dar mais peso para a extensão e essa medida obtém um valor menor que o da pureza para todas as redes.

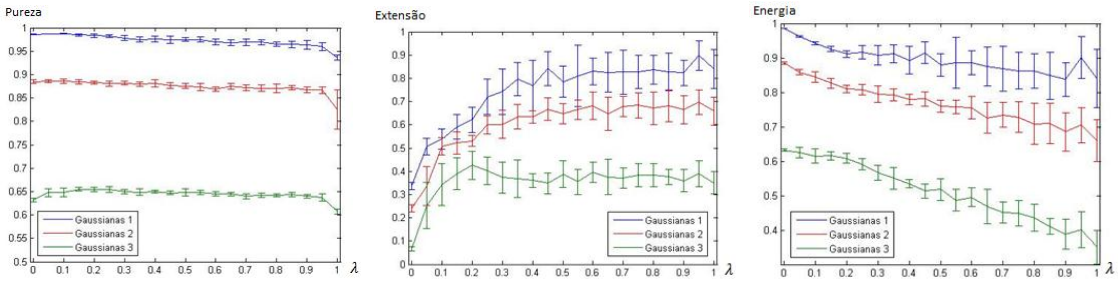


Figura 6: Representação da pureza, extensão e energia para as redes finais geradas a partir das bases de dados *Gaussianas 1-2-3*, com média sobre 30 execuções do algoritmo.

A Figura 7 mostra a pureza, extensão e energia das redes finais geradas quando a função de energia é maximizada em alguns valores de λ para as bases de dados Iris, Glass e Zoo. Nota-se que o valor da pureza para a base Iris assume valores mais próximos de 1, já que esta base apresenta menos mistura entre as classes, diferentemente das bases Glass e Zoo.

O valor da extensão para a base Iris assume valores maiores que as demais bases, pois consegue formar componentes conexos, além disso, o tamanho das três classes é igual. Já as bases Glass e Zoo apresentam além da mistura entre os componentes, classes com tamanhos variados, de modo que quando o cálculo da extensão é realizado, ele é ponderado pelo tamanho da maior classe, e com isso, o valor da extensão alcança valores mais baixos.

A energia apresenta um comportamento semelhante ao da base *Gaussianas*, ou seja, decai conforme λ se aproxima de 1 porque passa a dar mais peso para a extensão e essa medida obtém um valor menor que o da pureza para as redes testadas.

Analisando os resultados para pureza, extensão e energia, nota-se que as três medidas obtém valores mais baixos conforme o nível de mistura aumenta, de modo que estas medidas podem ser utilizadas para caracterizar a mistura nas classes dos dados. Além disso, observa-se que conforme a extensão das redes aumenta a pureza diminui levemente, indicando que seria possível utilizar uma rede formada para um valor maior de λ já que esta teria um número menor de componentes formados e poderia ser mais interessante para o processo de classificação.

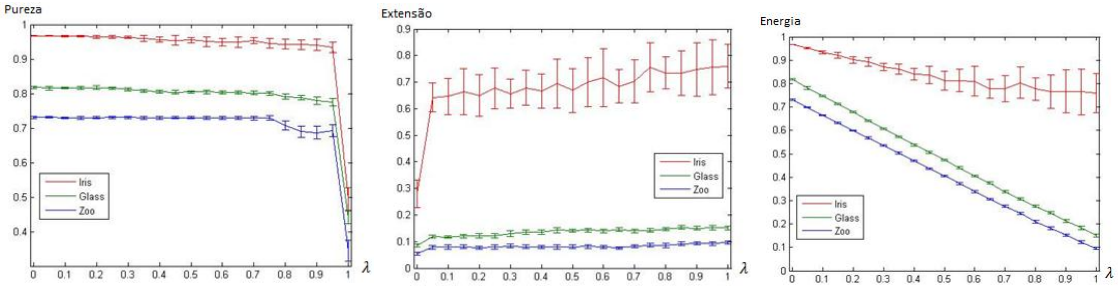


Figura 7: Representação da pureza, extensão e energia para as redes finais geradas a partir das bases de dados Iris, Glass e Zoo, com média sobre 30 execuções do algoritmo.

4. Conclusões

Este artigo apresenta um método para construção de redes baseado nas relações de similaridade entre os vértices da mesma classe e em uma função de energia que pondera medidas de pureza e extensão da rede. A construção de redes a partir de dados é uma fase importante no aprendizado baseado em grafos, de modo que este trabalho pode ser visto como uma contribuição para esta etapa.

O método foi aplicado em alguns conjuntos de dados artificiais e reais para caracterizar mistura entre as classes de dados. Os resultados obtidos mostraram que conforme a mistura dos dados aumenta, a pureza, a extensão e a energia diminuem, indicando que estas medidas podem ser utilizadas para caracterizar mistura de classes. Observa-se também que conforme a extensão aumenta a pureza não diminui de maneira considerável, indicando que redes formadas com um valor maior de λ poderiam ser utilizadas para a classificação quando se tem muita mistura nos dados, pois desse modo, se consideraria redes com um número menor de componentes formados.

A técnica de caracterização de classes embora apresente resultados preliminares, é um assunto importante e novo. Pode-se considerar que o presente trabalho é uma das primeiras tentativas nesta direção, além de ser uma contribuição para a construção de grafos a partir de dados proposicionais.

Como trabalho futuro pode-se aplicar o método em outros dados reais a fim de estudar seu comportamento nestes dados e caracterizá-los. O método pode ser aprimorado, de modo que diferentes medidas podem ser propostas e exploradas, visando extrair outras características de dados com diferentes classes. Podem ser testadas ainda outras medidas de similaridade para composição da rede, além da distância euclidiana.

Referências

Albert, R., Jeong, H. e Barabási, A. L. (1999) "Internet: Diameter of the world wide web". Nature 401, p.130-131.

Baldi, P. e Brunak, S. (1998) Bioinformatics: the machine learning approach, MIT Press.

Barabási, A. L. e Albert, R. (1999) "Emergence of scaling in random networks", Science 286, p. 509-512.

Belkin, M. Niyogi, P. (2003) "Laplacian eigenmaps for dimensionality reduction and data representation", Neural Computation 15, p. 1373-1396.

Boccaletti, S., Ivanchenko, M., Latora, V., Pluchino, A. e Rapisarda, A. (2007) "Detecting complex network modularity by dynamical clustering", Physical Review E 75, p. 1-4.

Bornholdt, S. e Schuster, H. G. (2003) Handbook of Graphs and Networks: From the Genome to the Internet, Wiley-VCH.

Cancho, F. e Solé, R. V. (2003) "Optimization in complex networks. Statistical Mechanics of Complex Networks", Lecture Notes in Physics 625, p. 114-125.

Cook, D. J. e Holder, L. B. (2000) "Graph-based data mining", IEEE Intelligent Systems 15, p. 32-41.

Erdös, P. e Rényi, A. (1959) "On random graphs", Publicationes Mathematicae 6, p. 290-297.

Faloutsos, M., Faloutsos, P. e Faloutsos, C. (1999) "On power-law relationship of the internet topology", Computer Communication Review 29, p. 251-262.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. e Jackel, L. D. (1989) "Backpropagation applied to handwritten zip code recognition", Neural Computation 1, p. 541-551.

Lopes, A. A., Bertini, Jr. J. R., Motta, R. e Zhao, L. (2009) "Classification Based on the Optimal K-Associated Network", Proceedings of The First International Conference on Complex Sciences: Theory and Applications, p. 1-11.

Mitchell, T. M. (1997) Machine learning. McGraw-Hill Series in Computer Science, McGraw-Hill.

Newman, M. E. J. (2004) "Fast algorithm for detecting community structure in networks", Physical Review E 69 066133.

Newman, M. E. J. e Girvan, M. (2004) "Finding and evaluating community structure in networks", Physical Review E 69 026113.

Quiles, M. G., Zhao, L., Alonso, R. L. e Romero, R. A. F. (2008) "Particle competition for complex network community detection", Chaos 18, p. 1-10.

Reichardt, J. e Bornholdt, S. (2004) "Detecting fuzzy community structures in complex networks with a Potts model" Physical Review Letters 93, p. 1-4.

Scott, J. (2000) Social network analysis: a handbook, Sage.

Sponrs, O. (2002) "Networks analysis, complexity, and brain function", Complexity 8, p. 56-60.

Von Luxburg, U. (2007) "A tutorial on spectral clustering", Statistical Computation 17, p. 395-416.

Watts, D. J. e Strogatz, S. H. (1998) "Collective dynamics of 'small-world' networks", Nature 393, p. 440-442.

Wolpert, D. H. e Macready, W. G. (1997) No Free Lunch Theorems for Optimization, IEEE Transactions on Evolutionary Computation 1, p. 67.

Zhou, H. (2003a) "Network landscape from a Brownian particle´s perspective". Physical Review E 67 041908.

Zhou, H. (2003b) "Distance, dissimilarity index, and network community structure". Physical Review E 67 061901.

Zhu, X. (2005) "Semi-supervised learning literature review". Technical Reporter 1530, Computer-Science, University of Wisconsin-Madison.