Universidade de São Paulo
Instituto de Matemática e Estatística
Bachalerado em Ciência da Computação

Tiago Madeira

Geração uniforme de *k-trees* para aprendizado de redes bayesianas

Supervisor: Prof. Dr. Denis Deratani Mauá

São Paulo
Novembro de 2016

Resumo

O resumo ainda não foi escrito.

Palavras-chave: sem, resumo, por, enquanto.

Abstract

The abstract has not been written yet.

Keywords: no, abstract, yet.

Sumário

Capítulo 1

Introdução

Em teoria dos grafos, k-*trees* são consideradas uma generalização de árvores. Há interesse considerável em desenvolver ferramentas eficientes para manipular essa classe de grafos, porque todo grafo com *treewidth* k é um subgrafo de uma k-*tree* e muitos problemas NP-completos podem ser resolvidos em tempo polinomial quando restritos a grafos com *treewidth* limitada.

Com efeito, o artigo de Arnborg e Proskurowski[1] apresenta algoritmos para resolver em tempo linear problemas como, dado um grafo com *treewidth* limitada:

- Encontrar o tamanho máximo dos seus conjuntos independentes;
- Computar o tamanho mínimo dos seus conjuntos dominantes;
- Calcular seu número cromático; e
- Determinar se ele tem um ciclo hamiltoniano.

O problema que desperta nosso interesse em k-*trees* é a inferência em redes bayesianas.

Uma rede bayesiana é um modelo probabilístico em grafo usado para raciocinar e tomar decisões em situações com incerteza através de técnicas de inteligência artificial e aprendizagem computacional. Ela representa uma distribuição de probabilidade multivariada num DAG (grafo acíclico dirigido) no qual os vértices correspondem às variáveis aleatórias do domínio e as arestas correspondem, intuitivamente, a influência de um vértice sobre outro.

Segundo Koller e Friedman[6], a inferência em redes bayesianas em geral é NP-difícil; porém, se seu DAG possui *treewidth* limitado, a inferência pode ser realizada em tempo polinomial. Daí a importância de aprender redes bayesianas que tenham *treewidth* limitada.

A partir dessa motivação, este trabalho de conclusão de curso consistiu em estudar os conceitos de teoria dos grafos relacionados a k-*trees* e implementar um algoritmo para gerar k-*trees* de forma uniforme que possam ser usadas no aprendizado de redes bayesianas.

A continuar.

Capítulo 2

Fundamentos

Neste capítulo, apresentamos definições fundamentais de teoria dos grafos, teoria da probabilidade e redes bayesianas que o leitor deve conhecer para compreender o trabalho.

Outras definições mais específicas, como as utilizadas para construir o algoritmo para codificar e decodificar k-*trees* estão localizadas nos capítulos subsequentes.

Partimos do pressuposto de que o leitor conhece notações básicas de conjuntos.

2.1 Grafos

Nesta seção apresentamos de forma breve apenas os conceitos de teoria dos grafos necessários para a compreensão deste trabalho. Mais detalhes podem ser encontrados no livro de Bondy e Murty[3], que foi utilizado como referência.

Definição 1 (grafo). Um grafo é um par ordenado $G = (V, E)$. Os elementos de V são chamados de vértices de G. Os elementos de E são chamados de arestas de G e consistem em pares (não-ordenados) de vértices distintos[1]. Dados $u, v \in V$, se $(u, v) \in E$ dizemos que u e v são adjacentes em G.

Definição 2 (grafo dirigido). Um grafo $G = (V, E)$ é dito dirigido se E consiste em pares *ordenados* de vértices.

Definição 3 (grafo completo). Um grafo $G = (V, E)$ é dito completo se $(u, v) \in E$ para todo $u, v \in V, u \neq v$.

Definição 4 (subgrafo). Um grafo $F = (V_F, E_F)$ é chamado de subgrafo de $G = (V_G, E_G)$ se $V_F \subseteq V_G$ e $E_F \subseteq E_G$.

Definição 5 (subgrafo induzido). Dado um grafo $G = (V, E)$ e um subconjunto V' de V, o subgrafo de G induzido por V', $G' = (V', E')$, é o grafo formado pelos vértices $V' \subseteq V$ e arestas que só contém elementos de V', ou seja, $E' = \{(u, v) \in E \mid u, v \in V'\}$.

Definição 6 (caminho). Dado um grafo $G = (V, E)$, um caminho em G é um subgrafo de G cujos vértices podem ser arranjados numa sequência linear de forma que dois vértices são adjacentes se eles são consecutivos na sequência e não-adjacentes caso contrário. Se $u, v \in V$ pertencem a um caminho P, dizemos que eles estão conectados pelo caminho P.

Definição 7 (distância). Dado um grafo $G = (V, E)$ e dois vértices $(u, v) \in V$, a distância entre u e v é o número de arestas num menor caminho que os conecte.

Definição 8 (ciclo). Dado um grafo $G = (V, E)$, um ciclo em G é um subgrafo de G cujos vértices podem ser arranjados numa sequência cíclica de

[1] A rigor, por causa da palavra "distintos", essa é a definição de *grafo simples*. Tal definição é utilizada porque neste trabalho não temos interesse em grafos que possuam arestas (u, v) com $u = v$.

forma que dois vértices são adjacentes se eles são consecutivos na sequência e não-adjacentes caso contrário.

Definição 9 (DAG). Um grafo $G = (V, E)$ é chamado de DAG (do inglês *directed acyclic graph*: grafo dirigido acíclico) se ele é dirigido e não possui ciclos.

Definição 10 (árvore). Dado um grafo $G = (V, E)$, dizemos que ele é uma árvore se cada dois vértices $u, v \in V$ são conectados por exatamente um caminho.

Definição 11 (k-clique). Seja $G = (V, E)$ um grafo. Um k-clique é um subconjunto dos vértices, $C \subseteq V$, tal que $(u, v) \in E\ \forall\ u, v \in C, u \neq v$ (ou seja, tal que o subgrafo induzido por C é completo).

2.1.1 k-*trees*

Definição 12 (k-*tree*). [5] Uma k-*tree* é definida da seguinte forma recursiva:

1. Um grafo induzido por um k-clique é uma k-*tree*.

2. Se $T'_k = (V, E)$ é uma k-*tree*, $K \subseteq V$ é um k-clique e $v \notin V$, então $T_k = (V \cup \{v\}, E \cup \{(v, x) \mid x \in K\})$ é uma k-*tree*.

Definição 13 (k-*tree* enraizada). [4] Uma k-*tree* enraizada é uma k-*tree* com um k-clique destacado $R = \{r_1, r_2, \cdots, r_k\}$ que é chamado de *raiz* da k-*tree* enraizada.

Definição 14 (*partial* k-*tree*). [2] Um subgrafo de uma k-*tree* é chamado de *partial* k-*tree*. Um grafo é uma *partial* k-*tree* se e só se ele tem *treewidth* menor ou igual a k.

2.2 Probabilidade

A escrever. [6]

2.3 Redes bayesianas

A escrever. [6]

Capítulo 3

Geração aleatória de k-*trees*

O problema de gerar k-*trees* está intimamente relacionado ao problema de codificá-las. De fato, se há um código bijetivo que associa k-*trees* à *bytes*, basta gerar *bytes* aleatórios para gerar k-*trees* aleatórias.

Neste capítulo, apresentamos o problema de codificar k-*trees*, discutimos a solução linear e bijetiva para codificar/decodificar k-*trees* proposta por Caminiti et al[4], explicamos como ela foi implementada neste trabalho para gerar k-*trees* aleatórias e mostramos os resultados obtidos.

3.1 Introdução à codificação de k-*trees*

A escrever.

3.2 A solução de Caminiti et al

A escrever.

Definição 15 (k-*tree* de Rényi). [7] Uma k-*tree* de Rényi R_k é uma k-*tree* enraizada com n vértices rotulados em $[1, n]$ e raiz $R = \{n - k + 1, n - k +$

$2, \cdots, n\}$.

Definição 16 (esqueleto de uma k-*tree* enraizada). [4] O esqueleto de uma k-*tree* enraizada T_k com raiz R, denotado por $S(T_k, R)$, é definido da seguinte forma recursiva:

1. Se T_k é apenas o k-clique R, seu esqueleto é uma árvore com um único vértice R.

2. Dada uma k-*tree* enraizada T_k com raiz R, obtida por T'_k enraizada em R através da adição de um novo vértice v conectado a um k-clique K (ver definição 12), seu esqueleto $S(T_k, R)$ é obtido adicionando a $S(T'_k, R)$ um novo vértice $X = \{v\} \cup K$ e uma nova aresta (X, Y), onde Y é o vértice de $S(T'_k, R)$ que contém K com uma distância mínima da raiz. Chamamos Y de pai de X.

Definição 17 (árvore característica). [4] A árvore característica $T(T_k, R)$ de uma k-*tree* enraizada T_k com raiz R é obtida rotulando os vértices e arestas de $S(T_k, R)$ da seguinte forma:

1. O vértice R é rotulado 0 e cada vértice $\{v\} \cup K$ é rotulado v;

2. Cada aresta do vértice $\{v\} \cup K$ ao seu pai $\{v'\} \cup K'$ é rotulada com o índice do vértice em K' (visualizando-o como um conjunto ordenado) que não aparece em K. Quando o pai é R a aresta é rotulada ε.

Note que a existência de um único vértice em $K' \setminus K$ é garantida pela definição 16. De fato, v' precisa aparecer em K, caso contrário $K' = K$ e o pai de $\{v'\} \cup K'$ contém K. Isso contradiz o fato de que cada vértice em $S(T_k, R)$ é ligado à distância mínima da raiz.

3.3 Experimentos e resultados

A escrever.

Capítulo 4

Aprendizado de redes bayesianas

A ser escrito.

Capítulo 5

Conclusão

Ainda não foi escrita.

Referências Bibliográficas

[1] Stefan Arnborg and Andrzej Proskurowski. Linear time algorithms for np-hard problems restricted to partial k-trees. *Discrete Applied Mathematics*, 23:11–24, 1989.

[2] Hans L. Bodlaender. Treewidth: Structure and algorithms. *Structural Information and Communication Complexity*, 4474:11–25, 2007.

[3] John A. Bondy and Uppaluri S. R. Murty. *Graph Theory*. Springer, 2008.

[4] Saverio Caminiti, Emanuele G. Fusco, and Rossella Petreschi. Bijective linear time coding and decoding for k-trees. *Theory of Computing Systems*, 46:284–300, 2010.

[5] Frank Harary and Edgar M. Palmer. On acyclic simplicial complexes. *Mathematika*, 15:115–122, 1968.

[6] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

[7] C. Rényi and A. Rényi. The prüfer code for k-trees. *Combinatorial Theory and its Applications*, pages 945–971, 1970.