Clusterização em grafos

Jacques Wainer

IC – Unicamp

Dezembro 2015

Dados relacionais ou em grafo

- Dados não tem componentes ou atributos - são atômicos
- O que é importante é a relação entre os dados (relacionais)
- ou dados são nós de um grafo (sem estrutura interna) e o importante são as arestas que ligam estes nós

Tipos de dados em grafos

- Bipartido - há 2 tipos de dados e so há relação entre um tipo e outro (e não entre nos do mesmo tipo). Pessoas e produtos, e uma aresta entre pessoas e produtos se a pessoa comprou o produto ou nao. Não trataremos de grafos bipartidos
- monopartido - só há um tipo de dado e relações entre eles. Pessoas, e uma aresta se elas ja se encontraram fisicamente.
- simétrico vs assimétrico: paginas web, se uma aponta para outra (assimétrico), pessoas se elas ja se encontram (simétrico)
- 0/1 ou valorado: a aresta existe ou nao existe (pagina web) ou tem um valor associado (0 se nao existe). Exemplo numero de emails que uma pessoa mandou para outra).

Similaridade vs distancia ou dissimilaridade

- para arestas valorados, a medida pode significar uma noção de **similaridade** - quanto maior o número mais parecidos/juntos os dados. Ex: numero de emails enviados
- ou pode significar uma medida de **distancia** ou **dissimilaridade** - quanto maior o número, menos parecidos menos juntos devem ser os dados.
- Alguns algoritmos assumem uma medida de similaridade outros de distancia - é preciso ficar atento a isso.
- normalmente se os dados são "originais" as medidas são de similaridade (emails trocados, estrelas na avaliação do produto, etc).
- grafos 0/1 é uma medida de similaridade
- para converter: $sim = \frac{1}{dist}$ ou $sim = \max(dist) - dist$ ou $sim = e^{\frac{-dist}{\sigma}}$

Convertendo dados assimétricos em simétricos

- os dados originais podem ser assimétricos (uma pagina aponta para outra) mas o algoritmos pode pedir uma medida simétrica
- conversões usuais $s_{ij} = \frac{a_{ij}+a_{ji}}{2}$
- $s_{ij} = \sqrt{a_{ij} a_{ji}}$

Convertendo dados vetoriais em grafos

- Calcule as distancias entre todos os pares de pontos - um grafo simétrico, completo com medida de dissimilaridade
- determine para cada ponto os k vizinhos mais próximos e compute uma similaridade baseado na distancia ate este vizinho. Para os outros pontos (nao vizinhos) a similaridade é 0 (k-NN grafo)
- determine para cada ponto, os vizinhos que estão a uma distancia menor que ϵ e compute uma similaridade baseado na distancia ate este vizinho. Para os outros pontos (nao vizinhos) a similaridade é 0 (ϵ-neighborhood grafo)

Clusterização em grafos - já visto

- k-medoids funciona em grafos pois não "cria" centros no espaço. Usa dissimilaridade mas pode ser usado com similaridade.
- clusterização hierárquica só usa a distancia entre pontos e portanto pode ser usada em grafos.

Clusterização baseada em MST

- Arvore geradora minima (minimum spanning tree) é uma arvore de menor soma das distancias que passa por todos os nós do grafo ▸ link
- determine a MSF e remova os k maiores arestas da MST - sobram k clusters ▸ link
- variações usam a distribuição das distancias no MST para cortar apenas arestas com distancias "grandes" (outliers)

Clusterização espectral

- Baseado em similaridade simétrica.
- cortes no grafo que minimizam a soma da similaridade das arestas contadas
- $S(i,j)$ é a similaridade entre i e j $(= S(j,i)$.
- o custo de separar um conjunto de nós A é $\sum S(i,j)$ onde $i \in A$ e $k \notin A$
- o volume de um conjunto de nos A é $vol(A) = \sum S(i,j)$ i e $j \in A$.

Clusterização espectral

- Duas definições de clusterização espectral:
- mincut: minimuze $cut(A_1, A_2, \ldots A_k) = \frac{1}{2} \sum_i^k custo(A_i)$
- mincut pode gerar clusters de tamanho muito diferentes
- normalized cut = minimize $\sum_i^k \frac{custo(A_i)}{vol(A_i)}$
- normalized cut gera clusters de tamanho mais balanceado (tamanho medido por $vol()$
- ha uma terceira definição

Clusterização espectral

- Mincut r Ncut são problema s difíceis de resolver.
- mas há aproximação a solução usando “autovetores do Laplaciano do grafo”
- diferente Laplacianos se relacionam com diferentes problemas: Mincut e Ncut

Clusterização espectral em R

- pacote `kknn` função `specClust` ▸ link
- baseado no k-nn grafo (parametro `nn` da função - nao sei como a função gera similaridade da distancia
- `method =` controla como calcular o Laplaciano - 3 opções. Não sei que opção corresponde a que problema.
- `centers` corresponde ao k numero de clusters. Sem `centers` a função determina o melhor numero de clusters (Como??)

Clusterização espectral em R

```
library(kknn)
data("iris")
ii=iris[,-5]
iris2d=prcomp(iris[,-5])$x[,1:2]
cl=specClust(ii,centers = 5,nn=7)
cl
plot(iris2d,col=cl$cluster)
cl2=specClust(ii,nn=7)
plot(iris2d,col=cl2$cluster)
cl3=specClust(ii,nn=15)
plot(iris2d,col=cl3$cluster)
```

Clusterização em grafos - outros

- o pacote `igraph` contem muitas funções em grafos
- clusterização em grafos é também chamada de "community detection"
- há varios algoritmos de community detection no igraph ▸ link
- o algoritmo fast greedy é usado em grafos muito grandes.
- outro algoritmo é o `affinity propagation` implementado pelo pacote `APcluster` ▸ link

Tarefa

- use os dados dos questionários respondidos pelos alunos. tarefa 2 da aula 3 ▸ link
- remova os atributos e codifique os atributos do tipo "proficiência em" e "dominio em" como discutido na tarefa
- Em quantos clusters os alunos do curso podem ser divididos
- nao existe resposta certa. Explore as alternativas, mostre seu trabalho, e se convença de uma resposta. E me convença que ela é a certa.

Clusterização semi-supervisionada

- Todos os algoritmos até agora forma não supervisionado. O algoritmos não tinha nenhuma informação sobre quem pertencia a qual cluster.
- nos usamos informação de classe dos dados para as métricas externas.
- clusterização semi-supervisionada: alguma informação sobre alguns dados é fornecida para o algoritmo de clusterização;
- mias comumente: *must-link* pares de dados que devem estar no mesmo cluster
- *must-not-link* pares de dados que não devem estar no mesmo cluster.

  ▸ link
- por exemplo, numa clusterização hierárquica, cortar o dendograma apenas onde as restrições must link nao sao quebradas e as must-not-link sao (exemplo simples)

Clusterização semi-supervisionada

- Em vez da informação supervisionada ter que ser "obedecida" ela pode indicar "dicas" que podem ser desconsideradas "se valer a pena"
- mais facil de implementar em clusterização baseada em distribuição de probabilidades (GMM).
- pacotes `EMcluster` e `Rmixmod` em R fazem isso