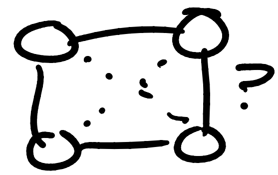
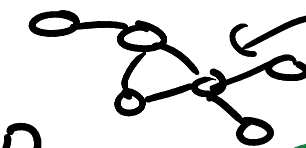
Lecture 17 - Vertex Classification

Thursday, March 31, 2022 7:56 AM

Graph mining:

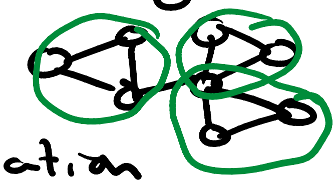


1. Link prediction
2. Centrality

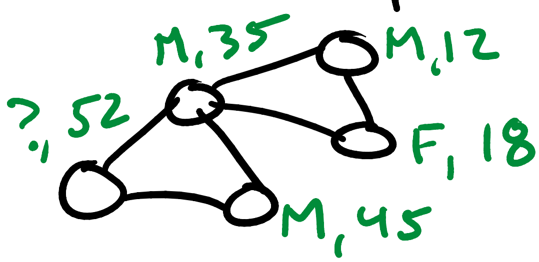


which vertex is most important

3. clustering / CD



4. vertex classification and label prediction



Vertex labeling problem

Given graph $G = (V, E, w, Y)$

V = vertex set

E = edge set

w = edge weights

Y = vertex labels

Example: social network

V = people

E = relationships

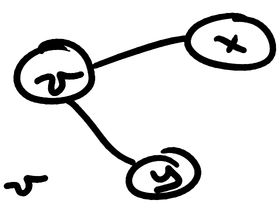
w = strength of relationship

W = strength of relationship
Y = demographic information

The problem:

Given $G = (V, E, W, Y_e)$
Predict Y_u
where Y_e = labeled data
Y_u = unlabeled data

Our approch: iterative classification



<u>Features</u>
Age of v
Gender of v
Politics of v
Avg. age of $N(v)$
proportions M|F|etc. of $N(v)$
proportions L|R|C of $N(v)$

features of v
- metadata of v
- metadata of $N(v)$

Basic classification problem:
- We have pieces of data
- we construct features from it
- train a classifier on known classes
- use to predict unknown classes

- train a classifier on ...
- use to predict unknown classes

Our iterative classification algorithm:

$\leftarrow$ features for labeled/unlabeled

construct ϕ_ℓ, ϕ_u from $G = (V, E, w, Y_\ell)$

$\leftarrow$ classifier

train f from ϕ_ℓ $\left(\min_{v_\ell \in V} \| f(\phi_\ell) - Y_\ell \|\right)$

min error output from classifier relative to ground truth

for some # iterations

predict $Y_u = f(\phi_u)$

update ϕ_u $\leftarrow$ updating features

return Y_u

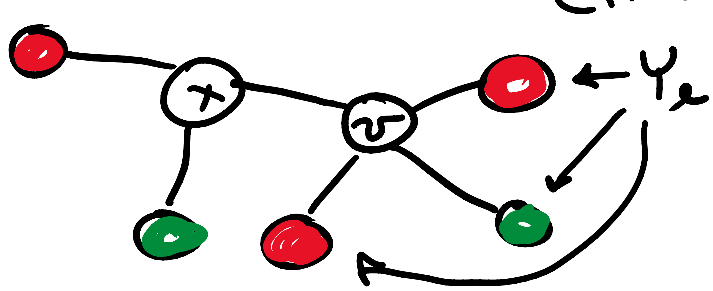
Classifiers

$\rightarrow$ recall label propagation

Y_ℓ = ground truth for $v_\ell \in V$

ϕ = labels of neighbors

f = return label w/ max count (ties broken randomly)



ϕ_v = counts for each label in $N(v)$

$\phi_v = \{ ● : 2, ● : 1, \ldots \}$

$f(\phi_v) =$ ●

$f(\phi_v) =$

Naive Bayes Classifier

$\bar{x}$ = features

$\bar{x}_v$ = features for vertex v

$\bar{x}_v = (x_{v_1}, x_{v_2}, x_{v_3} \dots x_{v_n})$

To classify some v as class C_i

$$\max_{i \in C} P(C_i \mid \bar{x}_v)$$

highest probability of C_i given features $\bar{x}_v$ of v

Two things first:

Bayes' Theorem $\rightarrow P(A \mid B) = \dfrac{P(A)P(B \mid A)}{P(B)}$

chain rule $\rightarrow P(A \cap B) = P(A)P(B \mid A)$

for more than two events

$$P(A_1 \cap A_2 \dots A_n) = P(A_1 \mid A_2 \dots A_n)P(A_2 \dots A_n)$$

$$P(C_i \mid \bar{x}_v) = \frac{P(C_i)P(\bar{x}_v \mid C_i)}{P\cancel{(\bar{x}_v)}}$$

Note: this is constant

$$P(C_i)P(\bar{x}_v \mid C_i) = P(C_i \cap \bar{x}_v)$$

$$P(C_i \cap \bar{x}_v) = P(C_i \cap x_{v_1} \cap x_{v_2} \cdots x_{v_k})$$

$$= P(x_{v_1} \cap x_{v_2} \ldots x_{v_k} \cap C_i)$$

chain rule again

$$= P(x_{v_1} \mid x_{v_2} \ldots x_{v_k} \cap C_i)\underline{P(x_{v_2} \ldots x_{v_k} \cap C_i)}$$

apply chain rule recursively

$$= P(x_{v_1} \mid x_{v_2} \ldots C_i)P(x_{v_2} \mid x_{v_3} \ldots C_i) \ldots P(C_i)$$

Naive Bayes assumption:
→ all x_j are independent

$$= P(x_{v_1} \mid C_i)P(x_{v_2} \mid C_i) \ldots P(C_i)$$

$$= P(C_i)\prod_{j=1}^{k} P(x_{v_j} \mid C_i)$$

$$\boxed{Y_u(\bar{x}_v) = \max_{i \in C} P(C_i)\prod_{j=1}^{k} P(x_{v_j} \mid C_i)}$$

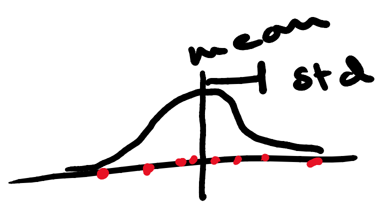
$P(C_i)$ = ratio of C_i labels

$P(C_i)$ = ratio of C_i labels

$$P(C_i) = \frac{|C_i|}{|V_\ell|}$$

$P(x_{v_j} | C_i)$ = over all C_i labels how often feature x_{v_j} shows up

→ we need to assume a distribution for numeric feature values



we can assume a normal distribution associated with each feature

what we're calculating for $P(x_{v_j} | C_i)$ is the probability that feature value x_{v_j} is sampled from the distribution (mean, variance) calculated from known labeled x_{u_j} features

Random Walks

Idea: the probability of some $v \in V_u$ assuming label C_i is the

$v \in V_u$ assuming label C_i is the probability of a random walk from v ends on some $u \in V_\ell$ with label C_i

$$\underbrace{P = D^{-1}W}$$

P = transition prob. matrix
D = diagonal degree matrix
W = weighted adjacency matrix

$P_{ij} \in P \rightarrow$ prob. of walk from $i \rightarrow j$

$\lim_{t \to \infty} P^t$ = gives us steady-state walk probability distribution

Note: PP gives us a matrix defining 2-hop distribution

Note x2: we want to stop our walk if we land on a labeled vert.

Let's consider such a modification to P

$P_i = e_i$ ← identity if $i \in V_\ell$

$P_j = (D^{-1}W)_i$ if $j \in V_u$

we can order vertices from labeled

We can order vertices from labeled first then unlabeled second

we stop on l ← don't go from $l \to u$

$$P = \begin{pmatrix} P_{ll} & P_{lu} \\ P_{ul} & P_{uu} \end{pmatrix} = \begin{pmatrix} I & 0 \\ P_{ul} & P_{uu} \end{pmatrix}$$

we still want $\lim_{t \to \infty} P^t$

$$P^\infty = \begin{pmatrix} I & 0 \\ (I - P_{uu})^{-1} P_{ul} & P_{uu}^\infty \end{pmatrix}$$

going to zero as we don't stop at and $v \in V_u$

$$P^\infty = \begin{pmatrix} I & 0 \\ (I - P_{uu})^{-1} P_{ul} & 0 \end{pmatrix}$$

Y_l = probability distribution over some class label

Vertex in labeled set

ith column

$$v_i \to Y_{l_i} = [0 \ldots 1 \ldots 0]$$

unlabeled

$$v_j \to Y_{u_j} = [0.1 \ldots 0.2 \ldots 0.05]$$

Prediction matrix

$$Y_u = (I - P_{uu})^{-1} P_{ul} Y_l$$

we want to find a max prob. for prediction

→ so to assign some label to v_i

$$\underset{c_i \in C}{\arg\max}\, Y_{u_j}$$

Iterative approach to calculate this

$$Y_u^{t+1} = P_{ul} Y_l + P_{uu} Y_u^t$$

via our original P matrix

Is this equivalent?

$$\boxed{Y_u^{t+1} = \sum_{i=1} P_{uu}^{i-1} P_{ul} Y_l + P_{uu}^{t+1} Y_u}$$

as $t \to \infty$

$$Y_u^{\infty} = (I - P_{uu})^{-1} P_{ul} Y_l$$

Yes