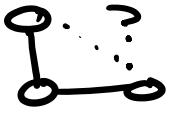
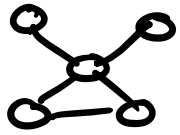
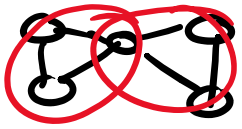
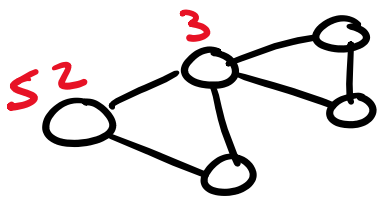
Graph Mining classic topics

1. Link prediction 

2. Centrality  ← which node is most importance?

3. clustering/CO 

4. Vertex(/edge) classification



← predict age

Vertex Labeling Problem

Given graph $G = \{V, E, W, Y\}$

V = vertex set

E = edge set

W = weights (edges)

Y = vertex labels

Y = vertex labels

Ex: social network

V = people

E = friendships / relationships

W = strength of relationships

Y = demographic information

The problem:

Given $G = \{V, E, W, Y_l\}$

(Y_l ← existing labels)

Predict Y_u ← unlabeled data

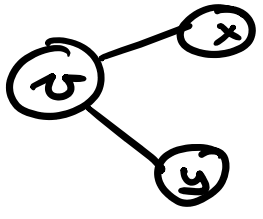
Approach: iterative classification

↳ we update our predictions using our current predictions

Features / labels (Y_l)

Age of v

Gender



Gender 

Politics

etc.

Average age $N(v)$

Proportion of M/F/NB/etc. of $N(v)$

Generally: metadata of v, $N(v)$

Basic classification problem:

- We have Y_ℓ, pieces of data
- We construct features
- Train classifier using Y_ℓ/features
- Predict Y_u, unlabeled data

Formalize:

Construct Φ_ℓ, Φ_u from G
(features)

Train f on Φ_ℓ $\left(\min_{v_\ell \in V} \| f(\Phi_\ell) - Y_\ell \|\right)$
(classifier)

For some # iterations:

Predict $Y_u = f(\phi_u)$

update ϕ_u (or ϕ_e)

(potentially retrain f)

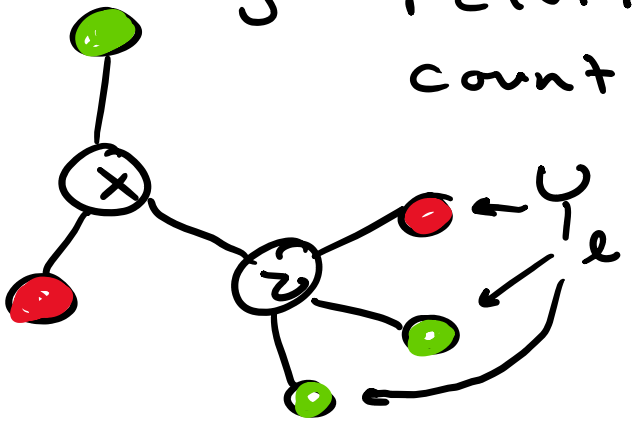
Return final Y_u

Classifier = Label propagation

Y_e = ground truth labels

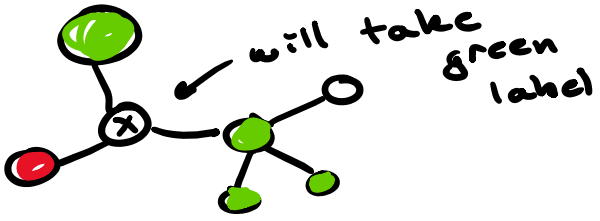
Q_v = labels in $N(v)$

f = return label with max count in $N(v) \mid \phi$



$\phi_v = \{\bullet = 2, \bullet = 1\}$

$f(\phi_v) = \{\bullet\}$



(same as for v)

$\phi_x = \{\bullet = 2, \bullet = 1\}$

$f(\phi_x) = \{\bullet\}$

Naive Bayes Classifier

$\bar{x}$ = features

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

$\bar{x}_v = \{x_{v_1}, x_{v_2}, \ldots x_{v_k}\}$

(probabilities)

To classify some v as class C_i

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

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

Two things first:

Bayes' Theorem → $P(A \mid B) = \frac{P(A)P(B \mid A)}{P(B)}$

Chain rule → $P(A \cap B) = P(A)\,P(B \mid A)$

for 3+ "events"

$$P(A_1 \cap A_2 \cap \ldots \cap A_k) = P(A_1)\,A_2 \cap \ldots \cap A_k) * P(A_2 \cap \ldots \cap A_k)$$

Classification:

trivially to calc. given labels

what we use to learn

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

(Bayes)

$P(\bar{x}_v)$ → constant

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

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

chain rule again

$$= P(x_{v_1} \mid x_{v_2} \dots x_{v_k} \cap C_i) * P(x_{v_2} \dots x_{v_k} \cap C_i)$$

↑ recursive chain rule

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

Naive Bayes' assumption

↳ all x_j are independent

$$P(x_{v_1} \mid C_i)\,P(x_{v_2} \mid C_i) \dots P(C_i)$$

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

$$Y_u(\bar{x}_u) = \max_{i \in C} P(C_i) \prod_{j=k}^{k} P(x_{v_j} | C_i)$$

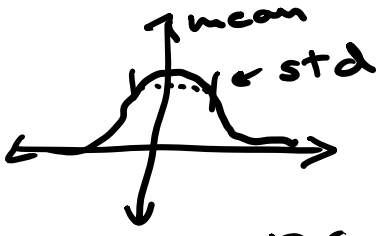
$P(C_i)$ = ratio of class labels

$$P(C_i) = \frac{|C_i|}{|V_l|} \leftarrow \text{\# verts in class } i$$

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

↳ discrete = easy

Continuous = need to assume a given distribution



$P(x_{v_j} | C_i)$ = the probability that the feature value x_{v_j} is sampled from same distribution (mean, var.) calculated from known

calculated from known
labeled x_{uj} features
$u \in V_l$

Random Walks

Idea: the prob. of some $v \in V_u$
assuming some C_i is the
prob. of a random walk from
v ends on some $u \in V_l$ with C_i

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

P = transitional prob. matrix

D = diagonal degree matrix (sum of weights)

W = weighted adj matrix

P_{ij} = prob. of a random walk from $i \to j$

$P = D^{-1}W \to$ a single step

$\lim_{t \to \infty} P^t$ = steady state solution

However: we only care about stopping on labeled vertices

We can modify our P

$P_i = e_i$ (identity) if $i \in V_\ell$

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

we label vertices from V_ℓ first and then V_u second

$$P = \begin{pmatrix} P_{\ell\ell} & P_{\ell u} \\ P_{u\ell} & P_{uu} \end{pmatrix} = \begin{pmatrix} I & 0 \\ P_{u\ell} & P_{uu} \end{pmatrix}$$

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

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

P_{uu}^∞ goes to zero as we don't stop at a V_u vertex

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

$$\Gamma = \left((I - P_{uu})^{-1} P_{ul} \; \bar{O} \right)$$

Y_l = prob. distribution over same class label

Vertex in labeled set

$v_i \in V_l, \; Y_{l_i} = [0 \ldots 1 \ldots 0]$ (1 in i^{th} colum)

unlabeled

$v_j \in V_u, \; Y_{u_j} = [0.1 \ldots 0.2 \ldots 0.05 \ldots 0.0001]$
(arbitrary example)

Prediction matrix

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

we want to find the max. prob. for our prediction

$$\underset{c_i \in C}{\operatorname{argmax}} \; Y_{u_i}$$

Iteratively (to calculate)

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

CODE MODE

Discogs → musical artists/album database

we can construct bipartite graphs

artist → genre (what we'll classify as singular)

artist → style (multiple)

↳ used to construct graph

We'll construct an overlapping community graph

artists → overlapping communities based on style

⇒ Predict genre