Ranking and labeling in graphs: Analysis of links and node attributes

Soumen Chakrabarti
IIT Bombay
http://www.cse.iitb.ac.in/∼soumen

Course plan: Ranking (2 hours)

- Feature vectors
  - Basics of discriminative and max-margin ranking
- Nodes in a graph
  - HITS and Pagerank
  - Personalized Pagerank and variations
  - Maximum entropy flows
  - Learning edge conductance

Course plan: Labeling (1.5 hours)

- Feature vectors
  - Discriminative loss minimization
  - Probabilistic and conditional models
  - Structured prediction problems
- Nodes in a graph
  - Directed Bayesian models, relaxation labeling
  - Undirected models, some easy graphs
  - Inference using LP and QP relaxations

Ranking feature vectors

- Suppose $x \in X$ are instances and $\phi : X \to \mathbb{R}^d$ a feature vector generator
- E.g., x may be a document and ϕ maps x to the "vector space model" with one axis for each word
- The score of instance x is $\beta'\phi(x)$ where $\beta \in \mathbb{R}^d$ is a weight vector
- For simplicity of notation assume x is already a feature vector and drop ϕ
- We wish to learn β from training data $\prec$: "$i \prec j$" means the score of x_i should be less than the score of x_j, i.e.,

$$\beta' x_i \leq \beta' x_j$$

Soft constraints

- In practice, there may be no feasible β satisfying all preferences $\prec$
- For constraint $i \prec j$, introduce slack variable $s_{ij} \geq 0$

$$\beta' x_i \leq \beta' x_j + s_{ij}$$

- Charge a penalty for using $s_{ij} > 0$

$$\min_{s_{ij} \geq 0; \beta} \sum_{i \prec j} s_{ij} \quad \text{subject to}$$
$$\beta' x_i \leq \beta' x_j + s_{ij} \quad \text{for all } i \prec j$$

A max-margin formulation

- Achieve "confident" separation of loser and winner:

$$\beta' x_i + 1 \le \beta' x_j + s_{ij}$$

- Problem: Can achieve this by scaling β arbitrarily; must be prevented by penalizing $\|\beta\|$

$$\min_{s_{ij} \ge 0; \beta} \frac{1}{2}\beta'\beta + B \sum_{i \prec j} s_{ij} \quad \text{subject to}$$

$$\beta' x_i + 1 \le \beta' x_j + s_{ij} \quad \text{for all } i \prec j$$

- B is a magic parameter that balances violations against model strength

Solving the optimization

- $\beta' x_i + 1 \leq \beta' x_j + s_{ij}$ and $s_{ij} \geq 0$ together mean $s_{ij} = \max\{0, \beta' x_i - \beta' x_j + 1\}$ ("hinge loss")
- The optimization can be rewritten without using s_{ij}

$$\min_{\beta} \frac{1}{2}\beta'\beta + B \sum_{i \prec j} \max\{0, \beta' x_i - \beta' x_j + 1\}$$

- $\max\{0, t\}$ can be approximated by a number of smooth functions
  - e^t – growth at $t > 0$ too severe
  - $\log(1 + e^t)$ – much better, asymptotes to $y = 0$ as $t \to -\infty$ and to $y = t$ as $t \to \infty$

Approximating with smooth objective

- Simple unconstrained optimization, can be solved by Newton method

$$\min_{\beta \in \mathbb{R}^d} \frac{1}{2}\beta'\beta + B \sum_{i \prec j} \log(1 + \exp(\beta' x_i - \beta' x_j + 1))$$

- If $\beta' x_i - \beta' x_j + 1 \ll 0$, i.e., $\beta' x_i \ll \beta' x_j$, then pay little penalty
- If $\beta' x_i - \beta' x_j + 1 \gg 0$, i.e., $\beta' x_i \gg \beta' x_j$, then pay large penalty

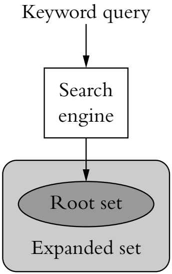
Ranking nodes in graphs

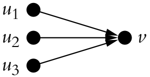
- Instances no longer feature vectors sampled from some distribution
- Instances are (also) nodes in a graph
- Instance should score highly if high-scoring instances link to it
- Two instantiations of this intuition

  Hyperlink-induced topic search (HITS): Nodes have two roles: hubs (fans) and authorities (celebrities)

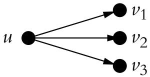
  Pagerank: Nodes have only one role: endorse other nodes

Quick HITS overview





$a(v) = h(u_1) + h(u_2) + h(u_3)$



$h(u) = a(v_1) + a(v_2) + a(v_3)$

$\vec{a} \leftarrow (1, \ldots, 1)^T, \vec{h} \leftarrow (1, \ldots, 1)^T$

while $\vec{h}$ and $\vec{a}$ change "significantly" **do**

$\quad \vec{h} \leftarrow E\vec{a}$

$\quad \ell_h \leftarrow \left\| \vec{h} \right\|_1 = \Sigma_w h[w]$

$\quad h \leftarrow h/\ell_h$

$\quad \vec{a} \leftarrow E^T h_0 = E^T E \vec{a}_0$

$\quad \ell_a \leftarrow \left\| \vec{a} \right\|_1 = \Sigma_w a[w]$

$\quad \vec{a} \leftarrow \vec{a}/\ell_a$

end while

- Authority flows along cocitation links, e.g., $v_1 \rightarrow u \rightarrow v_2$
- Note, hub (authority) scores are copied, not divided among authority (hub) nodes—important distinction from Pagerank and related approaches

Detour: Translation models

- Long-standing goal of Information Retrieval: return documents with words *related to* query words, without damaging precision
- Retrieval using language models: score document d wrt a query q (each interpreted as a set or multiset of words) by estimating $\Pr(q|d)$,
- If q_i ranges over query words and w ranges over all words in the corpus vocabulary, we can write

$$\Pr(q|d) = \prod_i \sum_w t(q_i|w) \Pr(w|d)$$

  assuming conditional independence between query words
- $t(q_i|w)$ is the probability that a corpus w gets "translated" into query word q_i (e.g., $q_i = \textit{random}$ and $w = \textit{probability}$)

Word-document random walks I

- Corpus as bipartite graph: word layer, document layer
- Document node d connects to word node w if w appears in d
- Random walk with absorption:
  1. Start the walk at node v initialized to w
  2. Repeat the following sub-steps: With probability $1 - \alpha$ terminate the walk at v, and with the remaining probability α execute these half-steps:
     - 2.1 From word node v, walk to a random document node d containing word v
     - 2.2 From document node d walk to a random word node $v' \in d$

     Now set $v \leftarrow v'$ and loop.
- Let there be m words and n documents

Word-document random walks II

- Starting with the m-node word layer, walking over to the n-node document layer can be expressed with a $m \times n$ matrix A, where $A_{wd} = \Pr(d|w)$
- Each row of A adds up to 1 by design
- Once we are at the document layer, the transition back to the word layer can be represented with a $n \times m$ matrix B, where $B_{dw} = \Pr(w|d)$
- Each row of B adds up to 1 by design
- In general $B \neq A'$
- The overall transition from words back to words is then represented by the matrix product $C = AB$, where C is $m \times m$
- Rows of C add up to one as well

Word-document random walks III

- Starting from word w, the probability that the process stops at word q after k steps is given by

$$(1-\alpha)\alpha^k (C^k)_{wq}$$

  where $(C^k)_{wq}$ is the (w,q)-entry of the matrix C^k
- Summing over all possible non-negative k, we get

$$\begin{aligned} t(q|w) &= (1-\alpha)(\mathbb{I} + \alpha C + \cdots + \alpha^k C^k + \cdots)_{wq} \\ &= (1-\alpha)(\mathbb{I} - \alpha C)^{-1}_{wq} \end{aligned}$$

▸ HW

- For $0 < \alpha < 1$, because rows of C add up to 1, $(\mathbb{I} - \alpha C)^{-1}$ will always exist
- Parameter $\alpha \in (0,1)$ controls the amount of diffusion

Word-document random walks IV

$w =$ *ebolavirus*, Web corpus: virus, ebola, hoax, viruses, outbreak, fever, disease, haemorrhagic, gabon, infected, aids, security, monkeys, hiv, zaire

$w =$ *starwars*, Web corpus: star, wars, rpg, trek, starwars, movie, episode, movies, war, character, tv, film, fan, reviews, jedi

$w =$ *starwars*, TREC corpus: star, wars, soviet, weapons, photo, army, armed, film, show, nations, strategic, tv, sunday, bush, series

- Starting at given w, top-scoring qs make eminent sense
- Depends on corpus, naturally

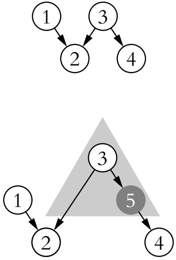
HITS-SVD connection I

- Let $A \in \{0,1\}^{m \times n}$ be a boolean matrix where A_{ij} is 1 if and only if word i ($1 \le i \le m$) occurs in document j ($1 \le j \le n$)
- This time let $B = A'$
- Do not bother with walk absorption and the parameter α
- Start from a mix of all words instead of one word, i.e., initialize $x = \mathbb{1}/m$
- After transition to documents the weight vector over documents is xA
- After transition back to words the weight vector over words is xAA'
- x, xAA', $x(AA')A$, $x(AA')(AA')$, $x(AA')^2A$, $\ldots$

HITS-SVD connection II

- Power iterations, converging to dominant eigenvector of $C = AA'$; C is a symmetric $m \times m$ matrix
- C has m eigenvectors; stack them vertically to get $U = u_{\cdot 1}, u_{\cdot 2}, \ldots, u_{\cdot m}$
- C satisfies $U'C = \Lambda U'$, where Λ is a diagonal matrix with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m \geq 0$
- Meanwhile suppose the SVD of A is $A_{m\times n} = U_{m\times m}\Sigma_{m\times n}V'_{n\times n}$ where $U'U = \mathbb{I}_{m\times m}$ and $V'V = \mathbb{I}_{n\times n}$
- $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_m)$ of singular values, with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \cdots \sigma_m = 0$, for some $0 < r \leq m$
- $C = AA' = U\Sigma \underline{V'V}\Sigma U' = U\Sigma\mathbb{I}\Sigma U' = U\Sigma^2 U'$, $\therefore CU = U\Sigma^2$, or $U'C = \Sigma^2 U'$

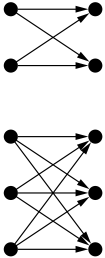
Topology sensitivity and winner takes all



$$E = \begin{vmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{vmatrix} \quad ; \; E^T E = \begin{vmatrix} 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{vmatrix}$$

$$E = \begin{vmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{vmatrix} \quad ; \; {}^T E = \begin{vmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{vmatrix}$$

(a)



(b)

- In (a, upper graph), $a_2 \leftarrow 2a_2 + a_4$ and $a_4 \leftarrow a_2 + a_4$ ▸ HW
- In (a, lower graph), $a_2 \leftarrow 2a_2 + a_4$, $a_4 \leftarrow a_4$, and $a_5 \leftarrow a_2 + a_5$ ▸ HW
- In (b), after k steps, $a_{\text{small}} = 2^{2i-1}$ and $a_{\text{large}} = 3^{2i-1}$ — ratio is $a_{\text{large}}/a_{\text{small}} = (3/2)^{2i-1}$ ▸ HW

HITS score stability I

- E is the node adjacency matrix
- Authority vector a is dominant eigenvector of $S = E'E$
- Perturb S to $\tilde{S}$, get $\tilde{a}$ in place of a
- Can S and $\tilde{S}$ be close yet a and $\tilde{a}$ far apart?
- Let $\lambda_1 > \lambda_2$ be the two largest eigenvalues of S
- Let $\delta = \lambda_1 - \lambda_2 > 0$
- S has a factorization

$$S = U \begin{bmatrix} \lambda_1 & 0 & \mathbf{0} \\ 0 & \lambda_2 & \mathbf{0} \\ 0 & 0 & \Lambda \end{bmatrix} U',$$

Each column of U an eigenvector of S having unit L_2 norm; Λ is a diagonal matrix of remaining eigenvalues

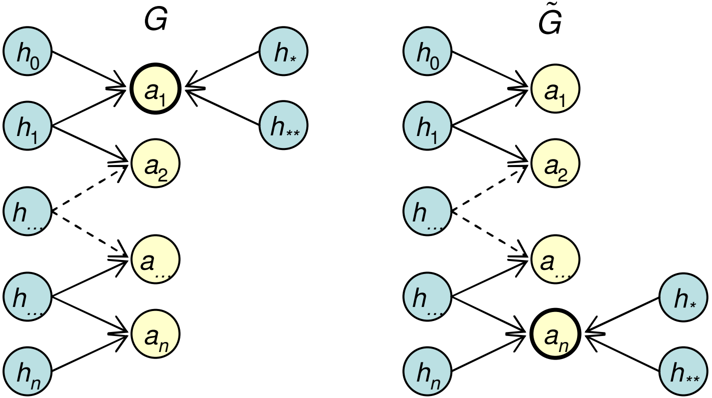
HITS score stability II

- Now we define

$$\tilde{S} = S + 2\delta U_{\cdot 2} U'_{\cdot 2} = U \begin{bmatrix} \lambda_1 & 0 & \mathbf{0} \\ 0 & \lambda_2 + 2\delta & \mathbf{0} \\ 0 & 0 & \Lambda \end{bmatrix} U'.$$

  Because $\|U_{\cdot 2}\|_2 = 1$, the L_2 norm of the perturbation, $\|\tilde{S} - S\|_2$, is 2δ.
- Given $\tilde{S}$ instead of S, how will λ_1 and λ_2 change to $\tilde{\lambda}_1$ and $\tilde{\lambda}_2$?
- By construction $\tilde{\lambda}_1 = \lambda_1$ while $\tilde{\lambda}_2 = \lambda_2 + 2\delta > \lambda_2 + \delta = \lambda_1 = \tilde{\lambda}_1$
- Therefore, $\tilde{\lambda}_1$ and $\tilde{\lambda}_2$ have switched roles and $\tilde{\lambda}_2$ is now the *largest* eigenvalue
- Old $a = U_{\cdot 1}$; new $\tilde{a} = U_{\cdot 2}$
- $\|a - \tilde{a}\|_2 = \|U_{\cdot 1} - U_{\cdot 2}\| = \sqrt{2}$

HITS rank stability, adversarial



- Number of edges changed is $O(1)$
- $\Omega(n^2)$ node pairs swapped in authority order ▸ HW

HITS rank stability in practice

1	Genetic algorithms in search optimization	Goldberg	1	3	1	1	1
2	Adaptation in natural and artificial systems	Holland	2	5	3	3	2
3	Genetic programming: On the programming of...	Koza	3	12	6	6	3
4	Analysis of the behavior of a class of genetic...	De Jong	4	52	20	23	4
5	Uniform crossover in genetic algorithms	Syswerda	5	171	119	99	5
6	Artificial intelligence through simulated...	Fogel	6	135	56	40	8
7	A survey of evolution strategies	Back+	10	179	159	100	7
8	Optimization of control parameters for genetic...	Grefenstette	8	316	141	170	6
9	The GENITOR algorithm and selection pressure	Whitley	9	257	107	72	9
10	Genetic algorithms + Data Structures = ...	Michalewicz	13	170	80	69	18
11	Genetic programming II: Automatic discovey...	Koza	7	-	-	-	10
2060	Learning internal representations by error...	Rumelhart+	-	1	2	2	-
2061	Learning to predict by the method of temporal...	Sutton	-	9	4	5	-
2063	Some studies in machine learning using checkers	Samuel	-	-	10	10	-
2065	Neuronlike elements that can solve difficult...	Barto+Sutton	-	-	8	-	-
2066	Practical issues in TD learning	Tesauro	-	-	9	9	-
2071	Pattern classification and scene analysis	Duda+Hart	-	4	7	7	-
2075	Classification and regression trees	Breiman+	-	2	5	4	-
2117	UCI repository of machine learning databases	Murphy+Aha	-	7	-	8	-
2174	Irrelevant features and the subset selection...	John+	-	8	-	-	-
2184	The CN2 induction algorithm	Clark+Niblett	-	6	-	-	-
2222	Probabilistic reasoning in intelligent systems	Pearl	-	10	-	-	-

- Random erasure of 30% of the nodes
- Fairly serious instability
- Is random erasure the right model?

Pagerank

> *. . . we are involved in an "infinite regress": [an actor's status] is a function of the status of those who choose him; and their [status] is a function of those who choose them, and so ad infinitum.*
>
> *Seeley, 1949*

- Random surfer roams around graph $G = (V, E)$
- Probability of walking from node i to j is $\Pr(j|i) = C(j, i)$
- C is a $|V| \times |V|$ nonnegative matrix; each column sums to 1 (what about dead-end nodes?)
- Steady-state probability of visiting node i is its prestige

Ways to handle dead-end nodes

Amputation: Remove dead-ends, may cause other nodes to become dead-ends, keep removing

- How to assign scores to the removed nodes?

Self-loop: Each dead-end node i links to itself

- Still trapped at i; need to escape/restart

Sink node: Dead-end nodes link to a sink node, which links to itself

- Reasonable, but probability of visiting sink node means nothing

Makes significant difference to node ranks (scilab demo)

Steady state probabilities

Long after the walk gets under way, at any time step, the probability that the random surfer is at a given node

Need two conditions for well-defined steady-state probabilities of being in each state/node

- E must be irreducible: should be able to reach any v starting from any u
- E must be aperiodic: There must exist some ℓ_0 such that for every $\ell \geq \ell_0$, G contains a cycle of length ℓ

Teleport

- Simple way to satisfy these conditions: all-to-all transitions

$$\tilde{C} = \alpha C + (1-\alpha)\frac{1}{|V|}\mathbb{1}_{|V|\times|V|}$$

  $\mathbb{1}_{|V|\times|V|}$ is a matrix filled with 1s; $\tilde{C}$ also has columns summing to 1
- Random surfer walks with probability α, jumps with probability $1-\alpha$
- What is the "right" value of α?
- Is α a device to make E irreducible and aperiodic, or does it serve other purposes?

Solving the recurrence

- Solve $p = \alpha C p + (1-\alpha)\mathbb{1}_{|V|\times 1}$ for steady-state visit probability $p \in \mathbb{R}^{|V|\times 1}$, with $p_i \geq 0$, $\|p\|_1 = \sum_i p_i = 1$
- Consider

$$\hat{C} = \begin{bmatrix} \alpha C_{|V|\times|V|} & \frac{\mathbb{1}_{|V|\times 1}}{|V|} \\ (1-\alpha)\mathbb{1}_{1\times|V|} & 0 \end{bmatrix}$$

- Dummy node d outside V
- Transition from every node $v \in V$ to d
- And a transition from d back to every node $v \in V$
- Recurrence can now be written as $\hat{p} = \hat{C}\hat{p}$
- What is the relation between p and $\hat{p}$? ▸ HW

Pagerank score stability

- V kept fixed
- Nodes in $P \subset V$ get incident links changed in any way (additions and deletions)
- Thus G perturbed to $\tilde{G}$
- Let the random surfer visit (random) node sequence $X_0, X_1, \ldots$ in G, and $Y_0, Y_1, \ldots$ in $\tilde{G}$
- Coupling argument: instead of two random walks, we will design one joint walk on (X_i, Y_i) such that the marginals apply to G and $\tilde{G}$

Coupled random walks on G and $\tilde{G}$

- Pick $X_0 = Y_0 \sim \text{Multi}(r)$
- At any step t, with probability $1 - \alpha$, reset both chains to a common node using teleport r: $X_t = Y_t \in_r V$
- With the remaining probability of α
  - If $x_{t-1} = y_{t-1} = u$, say, and u remained unperturbed from G to $\tilde{G}$, then pick one out-neighbor v of u uniformly at random from all out-neighbors of u, and set $X_t = Y_t = v$.
  - Otherwise, i.e., if $x_{t-1} \neq y_{t-1}$ or x_{t-1} was perturbed from G to $\tilde{G}$, pick out-neighbors X_t and Y_t independently for the two walks.

Analysis of coupled walks I

Let $\delta_t = \Pr(X_t \neq Y_t)$; by design, $\delta_0 = 0$.

$$\begin{aligned}
\delta_{t+1} = {} & \Pr(\text{reset at } t+1)\, \Pr(X_{t+1} \neq Y_{t+1} | \text{reset at } t+1) + \\
& \qquad \Pr(\text{no reset at } t+1)\, \Pr(X_{t+1} \neq Y_{t+1} | \text{no reset at } t+1) \\
= {} & \Pr(\text{reset at } t+1)\, 0 + \alpha \Pr(X_t \neq Y_t | \text{no reset at } t+1) \\
= {} & \alpha \big(\Pr(\underline{X_{t+1} \neq Y_{t+1}}, X_t \neq Y_t | \text{no reset at } t+1) + \\
& \qquad \Pr(X_{t+1} \neq Y_{t+1}, X_t = Y_t | \text{no reset at } t+1) \big)
\end{aligned}$$

The event $X_{t+1} \neq Y_{t+1}, X_t = Y_t$ can happen only if $X_t \in P$. Therefore we can continue the above derivation as follows:

Analysis of coupled walks II

$$\begin{aligned}
\delta_{t+1} &= \dots \\
&\leq \alpha\big(\Pr(X_t \neq Y_t | \underline{\text{no reset at } t+1}) + \\
&\qquad \Pr(X_{t+1} \neq Y_{t+1}, X_t = Y_t, \underline{X_t \in P} | \text{no reset at } t+1)\big) \\
&= \alpha\big(\Pr(X_t \neq Y_t) + \\
&\quad \Pr(X_{t+1} \neq Y_{t+1}, X_t = Y_t, \underline{X_t \in P} | \text{no reset at } t+1)\big) \\
&\leq \alpha\big(\Pr(X_t \neq Y_t) + \Pr(X_t \in P)\big) \\
&= \alpha\left(\delta_t + \textstyle\sum_{u \in P} p_u\right),
\end{aligned}$$

(using $\Pr(H, J|K) \leq \Pr(H|K)$, and that events at time t are independent of a potential reset at time $t+1$)

Unrolling the recursion,

$\delta_\infty = \lim_{t \to \infty} \delta_t \leq \left(\sum_{u \in P} p_u\right) / (1 - \alpha)$ HW

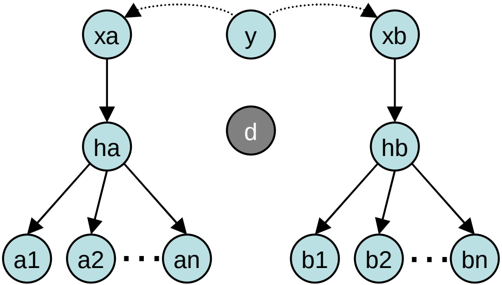
Analysis of coupled walks III

- Standard result: If the probability of a state disagreement between the two walks is bounded, then their Pagerank vectors must also have small L_1 distance to each other. In particular,

$$\|p - \tilde{p}\|_1 \leq \frac{2 \sum_{u \in P} p_u}{1 - \alpha}$$

- Lower the value of α, the more the random surfer teleports and more stable is the system
- Gives no direct guidance why α should not be set to exactly zero! (WAW talk)

Pagerank rank stability: adversarial



- G formed by connecting y to x_a, $\tilde{G}$ by connecting y to x_b
- $\Omega(n^2)$ node pairs flip Pagerank order ▸ HW
- I.e., L_1 score stability does not guarantee rank stability
- Can "natural" social networks lead often to such tie-breaking?

Pagerank rank stability: In practice

1	Genetic Algorithms in Search, Optimization and...	Goldberg	1	1	1	1	1
2	Learning internal representations by error...	Rumelhart+	2	2	2	2	2
3	Adaptation in Natural and Artificial Systems	Holland	3	5	6	4	5
4	Classification and Regression Trees	Breiman+	4	3	5	5	4
5	Probabilistic Reasoning in Intelligent Systems	Pearl	5	6	3	6	3
6	Genetic Programming: On the Programming of...	Koza	6	4	4	3	6
7	Learning to Predict by the Methods of Temporal...	Sutton	7	7	7	7	7
8	Pattern classification and scene analysis	Duda+Hart	8	8	8	8	9
9	Maximum likelihood from incomplete data via...	Dempster+	10	9	9	11	8
10	UCI repository of machine learning databases	Murphy+Aha	9	11	10	9	10
11	Parallel Distributed Processing	Rumelhart+	-	-	-	10	-
12	Introduction to the Theory of Neural Computation	Hertz+	-	10	-	-	-

- Quite stable, nowhere near adversarial
- Random 30% erasure hits many unpopular nodes, $\sum_{u \in P} p_u$ small
- Is random erasure a good assumption?

Other nonstandard path decay functions

- Standard Pagerank can be written as

$$p(\alpha) = (1-\alpha)\sum_{t\geq 0}\alpha^t r P^t = (1-\alpha)(\mathbb{I}-\alpha P)^{-1}\frac{\mathbb{1}}{|V|}$$

  where P is the row-normalized node adjacency matrix
- For path $\pi = (x_1, \ldots, x_k)$, let

$$\text{branching}(\pi) = \frac{1}{d_1\, d_2\, \cdots d_{k-1}}$$

- Equivalent Pagerank expression is

$$p_i(\alpha) = \sum_{\pi \in \text{path}(\cdot,i)} (1-\alpha)\alpha^{|\pi|}\,\text{branching}(\pi)/|V|$$

- Can generalize to

$$p_i = \sum_{\pi \in \text{path}(\cdot,i)} \text{damping}(|\pi|)\,\text{branching}(\pi)/|V|$$

- Important application: fighting link spam

Probabilistic HITS variants

In the analysis thus far, Pagerank's stability over HITS seems to come from two features:

- Pagerank *divides* among out-neighbors; hub score *copies* (which is why in HITS continual rescaling is needed)
- Pagerank uses teleport; HITS does not

Consider this authority-to-authority transition, starting at u

- Walk back to an in-neighbor of u, say w, chosen uniformly at random from all in-neighbors of v
- From w walk forward to an out-neighbor of w, chosen uniformly at random from all out-neighbors of w

No teleport yet, but dividing rather than copying

SALSA

- Combining the two half-steps, transition probability from authority v to authority w is

$$\Pr(w|v) = \frac{1}{\text{InDegree}(v)} \sum_{(u,v),(u,w)\in E} \frac{1}{\text{OutDegree}(u)}$$

- Suppose all pairs of authority nodes are connected to each other through alternating hub-authority paths
- Then $\pi_v \propto \text{InDegree}(v)$ is a fixpoint of the authority-to-authority transition process ▸ HW
- Overkill? Prevents any cocitation-based reinforcement!

HITS with teleport I

- Let the given graph be $G = (V, E)$. Remove any isolated nodes from G where no edge is incident.
- From G construct a bipartite graph $G_2 = (L, R, E_2)$, with $L = R = V$ and for each $(u, v) \in E$ connect the node corresponding to u in L to the node corresponding to v in R. By construction every node in L has some outlink and every node in R has some inlink.
- Write down the $(2|V|) \times (2|V|)$ node adjacency matrix for G_2.
- Write down the row-normalized node-adjacency matrix, which we will call E_2^{row}. Each row corresponding node $u \in L$ will add up to 1, and the rows for $v \in R$ will be all zeros.

HITS with teleport II

- Write down the column-normalized node-adjacency matrix, which we will call $E_2'^{\text{col}}$. Each row corresponding to node $v \in R$ will add up to 1, and the rows for $u \in L$ will be all zeros.
- Initialize an authority vector $a^{(0)}$ to be nonzero only for $v \in R$, with value $1/|R|$, and zero for all $u \in L$. Let $\mathbb{1}_h$ represent the uniform teleport vector distributed only over nodes in L, and $\mathbb{1}_a$ represent the uniform teleport vector

HITS with teleport III

distributed only over nodes in R. Compute the following iteratively:

$$
\begin{aligned}
h^{(1)} &= \alpha a^{(0)} E_2'^{\text{col}} + (1-\alpha)\mathbb{1}_h \\
a^{(1)} &= \alpha h^{(1)} E_2^{\text{row}} + (1-\alpha)\mathbb{1}_a \\
\cdots & \quad \cdots \\
h^{(k)} &= \alpha a^{(k-1)} E_2'^{\text{col}} + (1-\alpha)\mathbb{1}_h \\
a^{(k)} &= \alpha h^{(k)} E_2^{\text{row}} + (1-\alpha)\mathbb{1}_a
\end{aligned}
$$

etc. until convergence

HITS with teleport: Experience

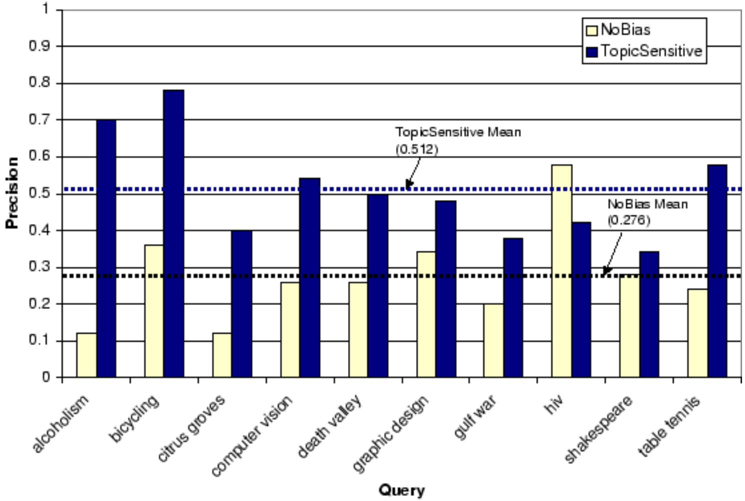
1	Learning internal representations by error...	Rumelhart+	1	3	3	2	1
2	Probabilistic Reasoning in Intelligent Systems	Pearl	4	1	1	1	2
3	Classification and Regression Trees	Breiman+	2	2	2	3	4
4	Pattern classification and scene analysis	Duda+Hart	3	4	4	4	3
5	Maximum likelihood from incomplete data via...	Dempster+	5	6	6	6	5
6	A robust layered control system for a mobile robot	Brook+	6	5	5	5	6
7	Numerical Recipes in C	Press+al	7	7	7	7	7
8	Learning to Predict by the Method of Temporal...	Sutton	8	8	8	8	8
9	STRIPS: A New Approach to ... Theorem Proving	Fikes+	9	10	10	10	15
10	Introduction To The Theory Of Neural Computation	Hertz+	11	11	9	9	9
11	Stochastic relaxation, gibbs distributions, ...	Geman+	10	9	-	-	-
12	Introduction to Algorithms	Cormen+	-	-	-	-	10

- Clearly much more rank-stable than HITS
- Is α all there is to stability?
- How to set α taking both content and links into account? (WAW talk)

Personalized Pagerank

- Recall we were solving $p = \alpha C p + (1-\alpha)\frac{\mathbb{1}_{|V|\times 1}}{|V|}$
- Can replace $\frac{\mathbb{1}_{|V|\times 1}}{|V|}$ with arbitrary teleport vector r, $r_i \geq 0$, $\sum_i r_i = 1$, examples:
  - $r_i > 0$ for pages i that you have bookmarked, 0 for other pages
  - $r_i > 0$ for pages about topic “Java programming”, 0 for other pages
- Extreme case of r: $r_i = 1$ for some specific node, 0 for all others — r called x_i in that case (“basis vector”)
- p is a function of r (and C) — write as p_r

Topic-sensitive Pagerank



- Details of how query is "projected" to topic space
- Clear improvement in precision

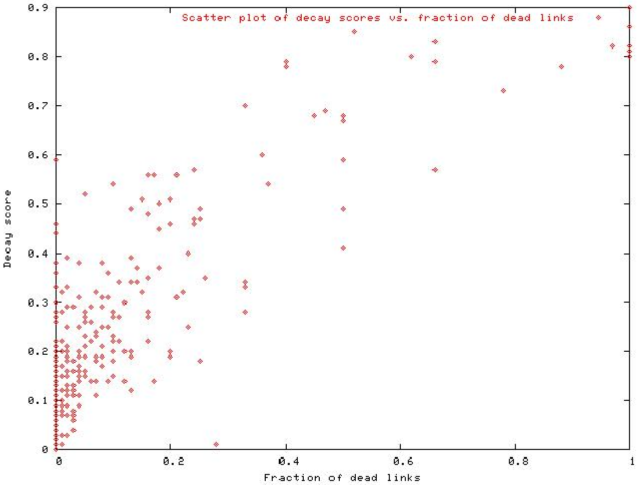
Page staleness

"A page is stale if it is inaccessible, or if it links to many stale pages"—to find how stale a page u is,

$v \leftarrow u$
for ever **do**
  if page v is inaccessible **then**
    return $s(u) = 1$
  toss a coin with head probability σ
  if head **then**
    return $s(u) = 0$ {with probability σ}
  else
    choose $w : (v, w) \in E$ with probability $\propto C(w, v)$
    $v \leftarrow w$

$$s(u) = \begin{cases} 1, & u \in D \\ (1-\sigma)\sum_v C(v,u)\, s(v), & \text{otherwise} \end{cases}$$

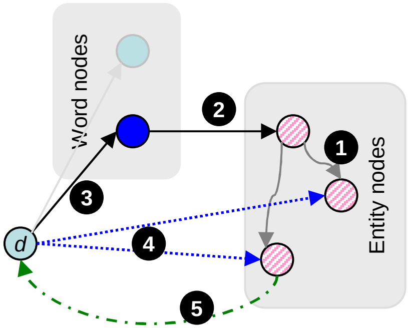
Page staleness: Experience



Staleness of a page is generally larger than the fraction of dead links on the page would have you believe

Biased walk for keyword search in graphs

- Teleport to query word nodes (3)
- Also teleport to entity nodes (4)
- Competition between relevance to query and query-independent prestige
- Each edge e has type $t(e)$ and weight $\beta(t(e))$



Effect of tuning edge weights

`transaction serializability`, $\beta(d \to \text{word})/\beta(d \to \text{entity}) = 1$	#cites
Graph based algorithms for boolean function manipulation	506
Scheduling algorithms for multiprogramming in a hard real time environment	413
A method for obtaining digital signatures and public key cryptosystems	312
Rewrite systems	265
Tcl and the Tk toolkit	242
`transaction serializability`, $\beta(d \to \mathit{word})/\beta(d \to \text{entity}) = 10^6$	#cites
On serializability of multidatabase transactions through forced local conflicts	38
Autonomous transaction execution with epsilon serializability	6
The serializability of concurrent database updates	104
Serializability a correctness criterion for global concurrency control in interbase	41
Using tickets to enforce the serializability of multidatabase transactions	12

- For small $\beta(d \to \mathit{word})$, query is essentially ignored
- Larger $\beta(d \to \mathit{word})$ gives better balance between query-independent prestige and query-dependent match
- Can learn $\beta(t)$s up to a scale factor from $\prec$ (WAW talk)

Personalization: Two key properties

- Cannot pre-compute p_r for all possible r
- Can we assemble Pageranks for an arbitrary r from Pageranks computed using "basis vectors"?

Linearity: If p_{r_1} is a solution to $p = \alpha Cp + (1-\alpha)r_1$ and p_{r_2} is a solution to $p = \alpha Cp + (1-\alpha)r_2$, then $p = \lambda p_1 + (1-\lambda)p_2$ is a solution to $p = \alpha Cp + (1-\alpha)(\lambda r_1 + (1-\lambda)r_2)$, where $0 \leq \lambda \leq 1$

Decomposition: If p_{x_u} is the Pagerank vector for $r = x_u$ and u has outlinks to neighbors v, then

$$p_{x_u} = \sum_{(u,v)\in E} \alpha C(v,u) p_{x_v} + (1-\alpha)x_u$$

HW

Learning r from $\prec$

- Recall $p = \alpha C p + (1-\alpha) r$, i.e., $(\mathbb{I} - \alpha C)p = (1-\alpha) r$, or $p = (1-\alpha)(\mathbb{I} - \alpha C)^{-1} r = Mr$, say
- $\prec$ can be encoded as matrix $\Pi \in \{-1, 0, 1\}^{|\prec| \times |V|}$ and written as $\Pi p \geq \mathbf{0}^{|\prec| \times 1}$ (each row expresses one pair preference)
- "Parsimonious teleport" is uniform $r_0 = \mathbb{1}_{|V| \times 1} / |V|$; that gives us standard Pagerank vector $p_0 = Mr_0$
- Want to deviate from p_0 as little as possible while satisfying $\prec$

$$\min_{r \in \mathbb{R}^{|V|}} (Mr - p_0)'(Mr - p_0) \quad \text{subject to}$$

$$\Pi M r \geq \mathbf{0}, \quad r \geq \mathbf{0}, \quad \mathbb{1}' r = 1$$

(quadratic objective with linear inequalities)

Pagerank as network flow

- Extend from learning r to learning "flow" of Pagerank on each edge $p_{uv} = p_u C(v, u) = p_u \Pr(v|u)$
- A valid flow satisfies

$$\sum_{(u,v)\in E'} p_{uv} = 1 \qquad \text{(Total)}$$

$$\forall v \in V' \quad \sum_{(u,v)\in E'} p_{uv} = \sum_{(v,w)\in E'} p_{vw} \qquad \text{(Balance)}$$

For all $v \in V_o \subseteq V$ having at least one outlink

$$(1-\alpha) \sum_{(v,w)\in E} p_{vw} = \alpha p_{vd} \qquad \text{(Teleport)}$$

- Pagerank satisfies these constraints, but so do many other flows

Maximum entropy flow

- Any principle to prefer one flow over another? Maximize entropy $\sum_{(u,v)\in E'} -p_{uv} \log p_{uv}$
- Or, stay close to a reference flow q by $\min_p \mathrm{KL}(p \| q)$
- The flows p_{uv} look like (β and τ unconstrained) ▸ HW

$$\begin{aligned}
\forall v \in V \quad & p_{dv} = (1/Z)\, q_{dv} \exp(\beta_v - \beta_d) \\
\forall v \in V_o \quad & p_{vd} = (1/Z)\, q_{vd} \exp(\beta_d - \beta_v + \alpha\tau_v) \\
\forall v \in V \setminus V_o \quad & p_{vd} = (1/Z)\, q_{vd} \exp(\beta_d - \beta_v) \\
\forall (u,v) \in E \quad & p_{uv} = (1/Z)\, q_{uv} \exp(\beta_v - \beta_u - (1-\alpha)\tau_u)
\end{aligned}$$

- Dual objective is $\max_{\beta,\tau} -\log Z$, with Z such that $\sum_{(u,v)\in E'} p_{uv} = 1$
- Can now add constraints like (WAW talk)

$$\forall u \prec v: \sum_{(w,u)\in E'} p_{wu} - \sum_{(w,v)\in E'} p_{wv} \quad \leq 0 \quad \text{(Preference)}$$

Labeling feature vectors and graph nodes

Labeling feature vectors

Training data: (x_i, y_i), $i = 1, \ldots, n$, $x_i \in \mathcal{X}$ (often $\mathbb{R}^d$)
$y_i \in \mathcal{Y} = \{-1, +1\}$

Single test instance: Given x not seen before, want to predict Y

Batch of test instances: Given many xs in a batch, predict Y for each x

Transductive learning: Given training and test batch together

Predictor: A parameterized function $f : \mathcal{X} \to \mathcal{Y}$; parameters learnt from training data

Loss: For instance (x, y), 1 if $f(x) \neq y$, 0 otherwise

Training loss: $\sum_{i=1}^{n} [\![y_i \neq f(x_i)]\!]$

Supervised learning approaches

Discriminative learning: Directly minimize (regularized) training loss

Joint probabilistic learning: Build a model for $\Pr(x, y)$, use Bayes rule to get $\Pr(Y = y|x)$

Conditional probabilistic learning: Directly build a model for $\Pr(Y = y|x)$

Linear parameterization of f

- Let $f(x) = x\beta$, where $x \in \mathbb{R}^{1\times d}$ and $\beta \in \mathbb{R}^{d\times 1}$
- Training loss $\sum_{i=1}^{n} [\![y_i \neq f(x_i)]\!] = \sum_{i=1}^{n} [\![y_i x_i \beta < 0]\!]$
- As in ranking, we may insist on more than $y_i x_i \beta \geq 0$; say we want $y_i x_i \beta \geq 1$
- Training loss is $\sum_{i=1}^{n} [\![y_i x_i \beta < 1]\!] = \sum_i \text{step}(1 - y_i x_i \beta)$

$$\text{step}(z) = \begin{cases} 0, & z \leq 0 \\ 1, & z > 0 \end{cases}$$

- Step function has two problems wrt optimization of β
  - It is not differentiable everywhere
  - It is not convex
- Design surrogates for training loss so that we can search for β

Hinge loss

- $\max\{0, 1 - y_i x_i \beta\}$ is an upper bound on training loss

$$\min_{\beta,s} \frac{1}{2}\beta'\beta + \frac{B}{n}\sum_i s_i \quad \text{subject to}$$

$$\forall i \quad s_i \geq 1 - y_i x_i \beta, \quad s_i \geq 0$$

- Standard soft-margin primal SVM; dual is ▸ HW

$$\min_{\alpha \in \mathbb{R}^n} \frac{1}{2}\alpha' X' Y' Y X \alpha - \mathbb{1}'\alpha$$

$$\text{subject to} \quad \forall i : \quad 0 \leq \alpha_i \leq B \quad \text{and} \quad y'\alpha = \mathbf{0}$$

Here $y = (y_1, \ldots, y_n)'$ and $Y = \text{diag}(y)$.

Soft hinge loss

- "Soft hinge loss" $\ln(1 + \exp(1 - y_i x_i \beta))$ is a reasonable approximation for $\max\{0, 1 - y_i x_i \beta\}$
- (Primal) optimization becomes

$$\min_{\beta} \frac{1}{2}\beta'\beta + \frac{B}{n}\sum_i \ln(1 + \exp(1 - y_i x_i \beta))$$

- Compare with logistic regression with a Gaussian prior:

$$\begin{aligned}
&\max_{\beta} \sum_i \log \Pr(y_i|x_i) - \frac{\lambda}{2}\beta'\beta \\
&= \min_{\beta} \sum_i - \log \Pr(y_i|x_i) + \frac{\lambda}{2}\beta'\beta \\
&= \min_{\beta} \sum_i \ln(1 + \exp(-y_i x_i \beta)) + \frac{\lambda}{2}\beta'\beta
\end{aligned}$$

Classification for large $\mathcal{Y}$

Collective labeling of a large number of instances, whose labels cannot be assumed to be independent, e.g.,

- Assigning multiple topics from a topic tree/dag to a document
- Assigning parts of speech (pos) to a sequence of tokens in a sentence
- Matching tokens across an English and a Hindi sentence that say the same thing

A generic device: include x and y into a feature generator $\psi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$

- Given x, prediction is $\arg\max_{y \in \mathcal{Y}} \beta' \psi(x, y)$
- In training set, want $\beta' \psi(x_i, y_i)$ to beat $\beta' \psi(x_i, y)$ for all $y \neq y_i$

Large $\mathcal{Y}$ example: Markov chain

- For simplicity assume all sequences of length exactly T; x, y now sequences of length T
- Labels Σ (noun, verb, preposition, etc.); $\mathcal{Y} = \Sigma^T$, huge
- x_i^t (y_i^t) is the tth token (label) of the ith instance
- Suppose there are W word-based features, e.g., `hasCap`, `hasDigit` etc.
- $\psi(x, y) =\in \mathbb{R}^d$ where $d = W\,|\Sigma| + |\Sigma|\,|\Sigma|$

$$\psi(x, y) = \sum_{t=1}^{T} \psi(y^{t-1}, y^t, x, t),$$

$$\text{where} \quad \psi(y, y', x, t) = (\ \underbrace{\hat{\psi}(x, y')}_{W\,|\Sigma|,\text{emission}}\ ,\ \underbrace{\vec{\psi}(y, y')}_{|\Sigma|\,|\Sigma|,\text{transition}}\ )$$

- Corresponding model weights $\beta = (\hat{\beta}, \vec{\beta}) \in \mathbb{R}^d$

Max-margin training for large $\mathcal{Y}$

- Given (x_i, y_i), $i = 1, \ldots, n$, want to find β such that for each instance i,

$$\beta'\psi(x_i, y_i) \geq \beta'\psi(x_i, y) + \text{margin} \quad \forall y \in \mathcal{Y} \setminus \{y_i\}$$

- Leads to the following optimization problem:

$$\min_{\beta, s \geq \mathbf{0}} \frac{1}{2}\beta'\beta + \frac{B}{n}\sum_i s_i \quad \text{subject to}$$

$$\forall i, \forall y \neq y_i \qquad \beta'\delta\psi_i(y) \geq 1 - \frac{s_i}{\Delta(y_i, y)}$$

- $\Delta(y_i, y)$ is severity of mismatch
- $\delta\psi_i(y)$ is shorthand for $\psi(x_i, y_i) - \psi(x_i, y)$
- Exponential number of constraints in primal and variables α_{iy} in dual

Cutting plane algorithm to optimize dual

- Primal: $\min_x f(x)$ subject to $g(x) \leq \mathbf{0}$
- Dual: $\max_{x,z} z$ subject to $u \geq \mathbf{0}$, $z \leq f(x) + u'g(x)\ \forall x$
- Approximate finite dual: $\max z$ s.t. $z \leq f(x_j) + u'g(x_j)$ for $j = 1, \ldots, k-1$, $u \geq \mathbf{0}$
- "Master program": for $k = 1, 2, \ldots$
  - Let (z_k, u_k) be current solution
  - Solve $\min_x f(x) + u_k'g(x)$ to get x_k
  - If $z_k \leq f(x_k) + u_k'g(x_k) + \epsilon$ terminate
  - Add constraint $z \leq f(x_k) + u'g(x_k)$ to approximate dual
- Dual max objective is non-decreasing with k
- Strictly increasing if $\epsilon > 0$

SVM training for structured prediction

$S_i = \varnothing$ for $i = 1, \ldots, n$

repeat

  for $i = 1, \ldots, n$ **do**

    current $\beta = \sum_j \sum_{y' \in S_j} \alpha_{jy'} \delta\psi_j(y')$ (Representer Theorem)

    we want $\beta' \delta\psi_i(y) \geq 1 - s_i / \Delta(y_i, y)$ or $s_i \geq \Delta(y_i, y)(1 - \beta' \delta\psi_i(y)) = H(y)$, say

    $\hat{y}_i = \arg\max_{y \in \mathcal{Y}} H(y)$ {to look for violations}

    $\hat{s}_i = \max\{0, \max_{y \in S_i} H(y)\}$

    if $H(\hat{y}_i) > \hat{s}_i + \epsilon$ **then**

      add $\hat{y}$ to S_i {admit $\alpha_{i\hat{y}}$ into dual}

      $\alpha_S \leftarrow$ dual optimum for $S = \cup S_i$

until no S_i changes

Structured SVM: Analysis sketch

- Let $\bar{\Delta} = \max_{i,y} \Delta(y_i, y)$, $\bar{R} = \max_{i,y} \|\delta\psi_i(y)\|_2$
- After every inclusion, dual objective increases by

$$\min\left\{\frac{B\epsilon}{2n}, \frac{\epsilon^2}{8\bar{\Delta}^2\bar{R}^2}\right\}$$

- Dual objective upper bounded by min of primal which is at most $B\bar{\Delta}$
- Number of inclusion rounds is at most

$$\max\left\{\frac{2n\bar{\Delta}}{\epsilon}, \frac{8B\bar{\Delta}^3\bar{R}^2}{\epsilon^2}\right\}$$

- Need inference subroutine: $\max_y \Delta(y_i, y)(1 - \beta'\delta\psi_i(y))$
- Can do this for Markov chains in poly time ▸ HW

Directed probabilistic view of Markov network

Concrete setting:

- Hypertext graph $G(V, E)$
- Each node u is associated with observable text $x(u)$; text of node set A denoted $x(A)$
- Each node has unknown (topic) label y_u; labels of node set A denoted $y(A)$

Our goal is

$$\arg\max_{y(V)} \Pr(y(V)|E, x(V)) = \arg\max_{y(V)} \frac{\Pr(y(V))\Pr(E, x(V)|y(V))}{\Pr(E, x(V))}$$

$$\text{where} \quad \Pr(E, x(V)) = \sum_{y(V)} \Pr(y(V))\Pr(E, x(V)|y(V))$$

is a scaling factor (which we do not need to know for labeling).

Using the Markov assumption

- $V^K \subset V$ has known labels $y(V^K)$
- Fix node v with neighbors $N(v)$
- Known labels for $N^K(v)$, unknown labels for $N^U(v)$

$$\Pr(Y(v) = y | E, x(V), y(V^K))$$
$$= \sum_{y(N^U(v)) \in \Omega_v} \Pr(y, y(N^U(v)) | E, x(V), y(V^K))$$
$$= \sum_{y(N^U(v)) \in \Omega_v} \Pr(y(N^U(v)) | E, x(V), y(V^K)) \Pr(y | y(N^U(v)), E, x(V), y(V^K))$$

- Ω_v = label configurations of $N^U(v)$ (can be large)
- "Solve for" all $\Pr(Y(v) = y | \ldots)$ simultaneously

Relaxation labeling

- To ease computation, approximate as in naive Bayes

$$\Pr(y(N^U(v)) \mid E, x(V), y(V^K)) \approx \prod_{w \in N^U(v)} \Pr\left(y(w) \,\middle|\, E, x(V), y(V^K)\right)$$

- Estimated class probabilities in the rth round is $\Pr_{(r)}(y(v) \mid E, x(V), y(V^K))$.
- May use a text classifier for $r = 0$

Relaxation steps

- Update as follows

$$\underline{\Pr_{(r+1)}(y(v) \mid E, x(V), y(V^K))}$$

$$\approx \sum_{y(N^U(v)) \in \Omega_v} \left[\prod_{w \in N^U(v)} \Pr_{(r)}\left(y(w) \,\middle|\, E, x(V), y(V^K)\right) \right]$$

$$\Pr\left(y(v) \,\middle|\, y(N^U(v)), E, x(V), y(V^K)\right)$$

- More approximations

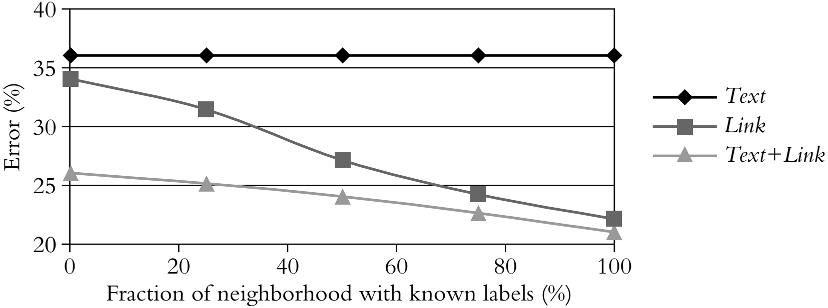
$$\Pr\left(y(v) \,\middle|\, y(N^U(v)), E, x(V), y(V^K)\right)$$

$$\approx \Pr\left(y(v) \,\middle|\, y(N^U(v)), E, x(V), y(N^K(v))\right)$$

$$\approx \Pr\left(y(v) \,\middle|\, y(N(v)), x(v)\right)$$

- Add terms for deterministic annealing? ▸ HW

Relaxation labeling: Sample results



- Randomly sample node, grow neighborhood, randomly erase fraction of known labels, reconstruct, evaluate
- Text+link better than link better than text-only
- Link better than text even when all labels wiped out! (associative prior: pages link to similar pages)

Undirected view of Markov network

- Each node u represents random variable X_u
- Undirected edges express potential dependencies
- Each clique $c \subset V$ has associated potential function ϕ_c
  - Input to ϕ is an assignment of values to X_c, say x_c
  - ϕ outputs a real number
- $\Pr(x) \propto \prod_{c \in C} \phi_c(x_c)$ (C is set of all cliques) — Hammersley-Clifford theorem
- $\Pr(x) = (1/Z) \prod_{c \in C} \phi_c(x_c)$ where $Z = \sum_x \prod_{c \in C} \phi_c(x_c)$ is the partition function

Conditional Markov networks

- Each node v has observable x_v and unobserved label y_v

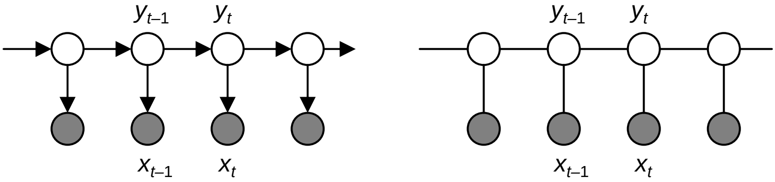
$$\Pr(y|x) = \frac{1}{Z(x)} \prod_{c \in C} \phi_c(x, y_c)$$

Potential functions and feature generators I

- $\Pr(y|x) = \frac{1}{Z(x)} \prod_{c \in C} \phi_c(x, y_c) = \frac{1}{Z(x)} \exp\left(\sum_{c \in C} \psi_c(x, y_c)\right)$
- Write (log) potential function ψ as

$$\psi_c(x, y_c) = \sum_k \beta_k f_k(x, y_c; c) = \beta' F(x, y_c; c)$$

- F is a feature (vector) generator
- c is a clique identifier; e.g., in case of a linear chain, $c = (t-1, t)$



Potential functions and feature generators II

- k is a feature identifier
- One feature may consider only t, y_t and x_t, and emit a number reflecting the compatibility between state y_t and observed word output x_t, or topic y_t and observed document x_t
- Another feature may consider only t, y_{t-1} and y_t, and emit a number reflecting the belief that a $y_{t-1} \rightarrow y_t$ can occur
- Have a weight β_k for each k
- Given fixed β, inference finds the most likely $y \in \mathcal{Y}$ (will see LP and QP relaxations soon)
- During training we fit β
- Training often uses inference as a subroutine

Training log-linear models

- Our goal is to find $\max_\beta L(\beta)$ where

$$L(\beta) = \sum_{i=1}^{n} \log \Pr(y_i | x_i)$$

$$= \sum_{i=1}^{n} \left[\sum_c \beta' F(x_i, y_{i.c}; c) - \log Z(x_i) \right]$$

$$\frac{\partial L}{\partial \beta} = \sum_{i=1}^{n} \left[\sum_c F(x_i, y_{i,c}; c) - \frac{\partial}{\partial \beta} \log Z(x_i) \right]$$

$$= \sum_{i=1}^{n} \left[\sum_c \left(F(x_i, y_{i,c}; c) - E_{Y|x_i} F(x_i, Y_c; c) \right) \right]$$ HW

- At optimum $F(x_i, y_{i,c}; c) = E_{Y|x_i} F(x_i, Y_c; c)$
- Once we have a procedure for the difficult part, we can easily use gradient-based methods to optimize for β
- For Markov chains, can use Viterbi decoding HW

Inference for Markov networks: LP relaxation I

- Labeling to minimize energy

$$\min_{y(V)} \left[\sum_{u \in V} c(u, y(u)) + \sum_{(u,v) \in E} w(u,v) \Gamma(y(u), y(v)) \right]$$

- c models local information at u
- Γ models compatibility of neighboring labels
- For two labels, sometimes easy via mincut

Inference for Markov networks: LP relaxation II

- Integer program formulation for $\Gamma(y, y') = [\![y \neq y']\!]$

$$
\begin{aligned}
\min \sum_{e \in E} w_e z_e + \sum_{u \in V, y \in \mathcal{Y}} c(u, y) x_{uy} & \\
\text{subject to} \sum_{y \in \mathcal{Y}} x_{uy} = 1 & \qquad \forall u \in V \\
z_e = \frac{1}{2} \sum_{y} z_{ey} & \qquad \forall e \in E \\
z_{ey} \geq x_{uy} - x_{vy} & \qquad \forall e = (u, v), \forall y \\
z_{ey} \geq x_{vy} - x_{uy} & \qquad \forall e = (u, v), \forall y \\
x_{uy} \in \{0, 1\} & \qquad \forall u \in V, y \in \mathcal{Y}
\end{aligned}
$$

▸ HW

- Can round to a factor of 2

Inference for Markov networks: QP relaxation I

- $\theta_{s;j}$ compatibility of node s with label j
- $\theta_{s,j;t,k}$ compatibility of edge (s,t) with labels (j,k)

$$\max \sum_{s,j} \theta_{s;j} [\![y(s) = j]\!] + \sum_{s,j;t,k} \theta_{s,j;t,k} [\![y(s) = j]\!] [\![y(t) = k]\!]$$

$$\text{subject to} \sum_{j} [\![y(s) = j]\!] = 1$$

Inference for Markov networks: QP relaxation II

- Relaxation of $[\![y(s) = j]\!]$ to $\mu(s, j)$:

$$\begin{aligned} \max \sum_{s,j} \theta_{s;j}\mu(s,j) + \sum_{s,j;t,k} \theta_{s,j;t,k}\mu(s,j)\mu(t,k) & \\ \text{subject to} \sum_{j} \mu(s,j) = 1 & \qquad \forall s \\ 0 \le \mu(s,j) \le 1 & \qquad \forall s, j \end{aligned}$$

- No integrality gap (proof via probabilistic method)
- Limitation: efficient QP solvers work only if $\Theta = \{\theta_{s,j;t,k}\}$ is negative definite
- If we try to make Θ negative definite, gap develops between QP optimum and label assignment

Concluding remarks

- Graphs and probability: at the intersection of statistics and classic AI knowledge representation
- Two computation paradigms: pushing weights along edges (Pagerank etc.) and computing local distributions or belief measures (graphical models)
- Lots of difficult problems!
  - Modeling
  - Optimization
  - Performance on real computers on large data
- Real applications both a challenge and an opportunity