Organization of Web Information (CS 728) **Final Exam**
Computer Science and Engineering **2011-05-01**
Indian Institute of Technology Bombay **09:30–13:00 SIC305**

NAME _____ ROLL _____

This exam has 9 printed page/s. Write your roll number on **EVERY SIDE (and not just sheet)**, because we may take apart your answer book and/or xerox it for correction. Write your answer clearly within the spaces provided and on any last blank page. **If you need more space than is provided, you probably made a mistake in interpreting the question.** Start with rough work elsewhere, but you need not attach rough work. Use the marks alongside each question for time management. **Illogical or incoherent answers are worse than wrong answers or even *no* answer, and may fetch negative credit.** You may not use any computing or communication device during the exam. You may use textbooks, class notes written by you, approved material downloaded **prior to the exam** from the course Web page, course news group, or the Internet, or notes made available by me for xeroxing. If you use class notes from other student/s, you must obtain them **prior to the exam** and **write down his/her/their name/s and roll number/s** here.

1. We will explore the annotation of images on the Web with class labels by exploiting labeled documents "connected to" (e.g., containing or hyperlinking) those images. The documents are said to be from the *source* domain where training data is plentiful. Documents are represented by feature vectors $x_i^{(s)} \in \mathbb{R}^a$, "$(s)$" for "source". We have available documents annotated with class labels $\mathcal{A}^{(s)} = \left\{ \left(x_i^{(s)}, y_i^{(s)} \right) : i = 1, \ldots, n^{(s)} \right\}$. For simplicity we will assume labels $y_i^{(s)} \in \pm 1$. The images are said to be from the *target* domain where training data is scarce. Images are represented by feature vectors $x_j^{(t)} \in \mathbb{R}^b$, "$(t)$" for "target". The limited target labeled data is $\mathcal{A}^{(t)} = \left\{ \left(x_j^{(t)}, y_j^{(t)} \right) : j = 1, \ldots, n^{(t)} \right\}$. Generally, $n^{(s)} \gg n^{(t)}$.

As part of the learning process we will estimate a *translator function* $T : \mathbb{R}^a \times \mathbb{R}^b \to \mathbb{R}$ that predicts the degree of (positive or negative) association between a document and an image. The class of a test image will be the sign of

$$f_T(x^{(t)}) = \sum_{i=1}^{n^{(s)}} y_i^{(s)} T(x_i^{(s)}, x^{(t)}).$$

Additional support for learning T will come from *connections* between the domains: $c(x_i^{(s)}, x_j^{(t)})$ represents the strength (perhaps 0/1) of connection between document i and image j.

1(a) As an abstract optimization without restricting the specific nature of T, what form of objective function over T would you set up? Write high-level symbolic expressions involving suitable loss functions, f_T, and regularization terms. (Hint: the objective has three prominent terms/parts.)

		3

- The power of the translator has to be limited by some form or regularization, $\Omega(T)$.
- For the known connections, T should align with c. If c is large and positive we want T to be large and positive, and vice versa. I.e., for every $c \neq 0$, we want $c(x_i^{(s)}, x_j^{(t)}) T(x_i^{(s)}, x_j^{(t)})$ to be large and positive. In max-margin style, we may assert

$$c(x_i^{(s)}, x_j^{(t)}) T(x_i^{(s)}, x_j^{(t)}) \geq 1,$$

possibly with a slack variable which is included in the objective. Another alternative is to define a loss function $\chi(\diamond)$ that is monotone decreasing in $\diamond$, and add

$$\sum_{i,j} \chi\left(c(x_i^{(s)}, x_j^{(t)}) T(x_i^{(s)}, x_j^{(t)})\right)$$

to the objective.

- Finally we need training loss over $\mathcal{A}^{(t)}$. This can be a hinge loss as in SVM.

1(b) A reasonable way to design T is to use linear projections of documents in $\mathbb{R}^a$ and images in $\mathbb{R}^b$ to a *topic* space in $\mathbb{R}^c$ where $c \ll a, b$. Let the projection matrices be $W^{(s)} \in \mathbb{R}^{c \times a}$ and $W^{(t)} \in \mathbb{R}^{c \times b}$. As an inner product in topic space, write down $T(x_i^{(s)}, x^{(t)})$.

		1

$x^{(s)} \in \mathbb{R}^a$ maps to $W^{(s)}x^{(s)} \in \mathbb{R}^c$, and $x^{(t)} \in \mathbb{R}^b$ maps to $W^{(t)}x^{(t)} \in \mathbb{R}^c$. Their inner product is $\left(W^{(s)}x^{(s)}\right)^\top W^{(t)}x^{(t)} = x^{(s)\top}\underline{W^{(s)\top}W^{(t)}}x^{(t)}$ ...

1(c) If you replace separate matrices $W^{(s)}$ and $W^{(t)}$ in the above expression with a single matrix S, what is its size?

		1

$\ldots = x^{(s)\top}\underline{S}x^{(t)}$, where $S \in \mathbb{R}^{a \times b}$.

1(d) Proposed a desirable regularization term over S and write down a concrete final optimization problem, preferably a convex one. Justify your design choices.

		3

What we really want is that S be low rank (parsimonious explanation of the data using very few topics). You will get full credit for pointing this out, or proposing the Frobenius norm $\|S\|_F^2$.

2. In product search, the user either uses a structured form interface, or an unstructured query is segmented into attribute names with corresponding *specified* values. These are like SQL select conditions. The product has other attributes that the user does not mention/specify. A reasonable principle for responding to such queries is that the products reported be close to the specified attribute values in the query, and *diverse* with respect to the unspecified attributes.

We will model each product as a node u in a complete graph (V, E), associated with a *cost* $c(u) \in \mathbb{R}_+$ that represents how well u fits specified attributes (the better the fit, the smaller is $c(u)$). There is also a *distance* $d(u, v) \in \mathbb{R}_+$ between products u and v computed using only the unspecified attributes. We will assume that d satisfies the triangle inequality.

If we show a subset $S \subset V$ of products, the total cost is $\sum_{u \in S} c(u)$. We also want the elements of S to be "far" from each other. A reasonable definition of such *dispersion* is $\sum_{u,v \in S} d(u, v)$. Since we want to minimize cost and maximize dispersion, a reasonable objective to maximize is

$$\sum_{u,v \in S} d(u, v) - \lambda \sum_{u \in S} c(u)$$

for some suitably tuned constant λ.

2(a) Often, there are also strong *prior* preferences on unspecified attributes. E.g., more megapixels and lower price are usually preferred; only diversity is not desirable. (How) can you adapt the above framework to incorporate such considerations?

		1

We can incorporate an additional term into $c(u)$ that is large if the attributes of the product are far from the desirable value/s, and small if they are close.

2(b) Let $y_u \in \{0,1\}$ be a decision variable representing whether $u \in S$. x_{uv} is another set of 0/1 decision variables, suitably constrained (see below). Complete the following integer linear program equivalent to the above optimization:

$$\max_{\{y_u\},\{x_{uv}\}} \sum_{\text{all } u,v} \_\_\_\_\_ x_{uv} - \lambda \sum_{\text{all } u} \_\_\_\_\_ y_u$$

$$\text{subject to} \quad \_\_\_\_\_ \leq \min\{\_\_\_\_\_, \_\_\_\_\_\} \quad \forall u, v \in V$$

2

$$\max_{\{y_u\},\{x_{uv}\}} \sum_{\text{all } u,v} d(u,v) x_{uv} - \lambda \sum_{\text{all } u} c(u) y_u$$

$$\text{subject to} \quad x_{uv} \leq \min\{y_u, y_v\} \quad \forall u, v \in V$$

I.e., x_{uv} is allowed to switch on only if both y_u and y_v are on.

2(c) Suppose we relax the integrality constraints $x_{uv}, y_u \in \{0,1\}$ to $x_{uv}, y_u \in [0,1]$, solve the LP, and get fractional solutions $\tilde{x}_{uv}, \tilde{y}_u$ and relaxed objective value $\tilde{O}$. We now generate a random threshold $\alpha \sim \mathcal{U}[0,1]$, and set $y_u = 1$ (i.e., $u \in S$) iff $y_u \geq \alpha$. What is $\mathbb{E}(y_u)$ (where the randomness is in the choice of α)?

1

$\Pr(y_u = 1) = \tilde{y}_u$ because we need $\alpha \in [0, \tilde{y}_u]$. Therefore $\mathbb{E}(y_u) = 1 \Pr(y_u = 1) + 0 \Pr(y_u = 0) = \tilde{y}_u$.

2(d) What is $\mathbb{E}(x_{uv})$?

2

Because every x_{uv} has a nonnegative coefficient, the fractional solution has no reason to pick $\tilde{x}_{uv} < \min\{\tilde{y}_u, \tilde{y}_v\}$; it will pick $\tilde{x}_{uv} = \min\{\tilde{y}_u, \tilde{y}_v\}$. After rounding, we can set $x_{uv} = 1$ iff $y_u = y_v = 1$. The probability of this happening is the same as $\Pr(\alpha \leq \min\{\tilde{y}_u, \tilde{y}_v\}) = \min\{\tilde{y}_u, \tilde{y}_v\}$. Therefore we again have $\mathbb{E}(x_{uv}) = \tilde{x}_{uv}$.

2(e) What is the expected rounded objective $\mathbb{E}(O)$?

1

By linearity, this is just O.

2(f) One problem with the above approach is that, if λ is not chosen well, the balance may become degenerate. An alternative is to maximize dispersion subject to cost being at most some budget B. In what follows, we will assume $c(u) = 1$ for all nodes (but this can be relaxed with some more work). This means we want to choose at most B nodes to maximize dispersion. Complete the following greedy algorithm:

```
set S = ∅
while B > 0 do
    if B = 1 then
        return _____
    find edge arg max_{(u,v)∈E} d(u,v)
    add u, v to S
    remove _____ from V
    remove all edges _____ from E
    B ← B − 2
return S
```

3

```
1: set S = ∅
2: while B > 0 do
3:   if B = 1 then
4:     return an arbitrary node u ∈ V
5:   find edge arg max_{(u,v)∈E} d(u, v)
6:   add u, v to S
7:   remove u, v from V
8:   remove all edges adjacent to u, v from E
9:   B ← B − 2
10: return S
```

2(g) Let O_B, G_B and O_{B-2}, G_{B-2} be optimal and greedy solutions for budgets B (first loop iteration) and $B-2$ (second loop iteration). Let $e = (i, j)$ be the edge with largest $d(e)$ first selected by greedy. Then $G_B = G_{B-2} \cup \{i, j\}$. Decompose the edges induced by G_B into three kinds:

- e itself
- (i, k) where $k \in G_B \setminus \{i, j\}$ and (j, k) where $k \in G_B \setminus \{i, j\}$
- (k, ℓ) where $k, \ell \in G_B \setminus \{i, j\}$

In contrast, let e' be some edge selected by optimal. By definition, $d(e) \geq d(e')$. Complete the choices of e':

- If $i, j \in O_B$, then set $e' =$ _____.
- If $|O_B \cap e| = 1$, then choose node $x \in O_B \setminus O_{B-2}$ such that $x \notin \{i, j\}$. Then, if $x \in$ _____, set $e' =$ _____, otherwise set $e' =$ _____.
- If $O_B \cap e = \varnothing$, then choose $e' =$ _____.

(Hint: You should read the next part before answering.)

		3

- If $i, j \in O_B$, then set $e' = (i, j)$.
- If $|O_B \cap e| = 1$, then choose node $x \in O_B \setminus O_{B-2}$ such that $x \notin \{i, j\}$. Then, if $x \in O_B$, set $e' = (i, x)$, otherwise set $e' = (j, x)$.
- If $O_B \cap e = \varnothing$, then choose $e' = (x, y)$ where $x, y \in O_B \setminus O_{B-2}$.

2(h) Complete the following: By triangle inequality, the $2(B-2)$ edges incident with e contribute at least _____ to G_B, whereas the $2(B-2)$ edges incident with e' have total weight at most _____. (This is the crux of the technique.)

		2

By triangle inequality, the $2(B-2)$ edges incident with e contribute at least $(B-2)w(e)$ to G_B, whereas the $2(B-2)$ edges incident with e' have total weight at most $2(B-2)w(e)$.

2(i) Using induction, prove that the objective of O_B is at most _____ times the objective of G_B.

		2

Induction on B. The approximation ratio is 2. I.e., optimal is at most twice as large as greedy.

3. In one variation on correlation clustering, we are given an undirected graph with n nodes where each edge e has two non-negative edge weights $w_{\text{in}}(e), w_{\text{out}}(e) \geq 0$. Suppose we are also given a partition the nodes of the graph. Over all edges e, if both endpoints of edge e are in the same partition, add $w_{\text{in}}(e)$ to the objective, otherwise, i.e., if the endpoints of edge e are in different

partitions, add $w_{\text{out}}(e)$ to the objective. The converse problem is, given the graph with edge weights, find a partitioning that maximizes the objective. The problem is NP-hard.

3(a) There are at most n partitions; let us number them in unary as $e_1 = (1, 0, 0, \ldots)^\top$, $e_2 = (0, 1, 0, \ldots)^\top$, etc. Let the vector $x_u \in \{e_1, \ldots, e_n\}$ denote the assignment of node u to one of the partitions. For nodes u, v, write down an expression using x_u and x_v whose value is 1 if u and v are the same partition.

		1

$x_u^\top x_v$.

3(b) Complete the objective in full using variables x_u where $x_u \in \{e_1, \ldots, e_n\}$:

$$\max_{\{x_u\}} \sum_{(u,v)\in E} w_{\text{in}}(u,v) \_\_\_\_\_ + w_{\text{out}}(e) \left(\_\_\_\_\_ - \_\_\_\_\_ \right)$$

		2

$$\max_{\{x_u\}} \sum_{(u,v)\in E} w_{\text{in}}(u,v) x_u^\top x_v + w_{\text{out}}(e) \left(\underset{\sim}{1} - x_u^\top x_v \right)$$

3(c) We will relax the above optimization such that each $x_u \in \mathbb{R}^n$. The objective will remain the same, but new constraints have to be added (complete with justification):

- $x_u^\top x_u =$ _____ for all u
- $x_u^\top x_v \geq$ _____ for all u_____v

		3

Answers:

- $x_u^\top x_u = \underset{\sim}{1}$ for all u. This is the continuous extension to $e_k^\top e_k = 1$.
- $x_u^\top x_v \geq \underset{\sim}{0}$ for all $u \underset{\sim}{\neq} v$. This comes from $e_k^\top e_{k'} \geq 0$. It also limits all angles between cluster vectors to $[0, \pi/2]$ which will be handy.

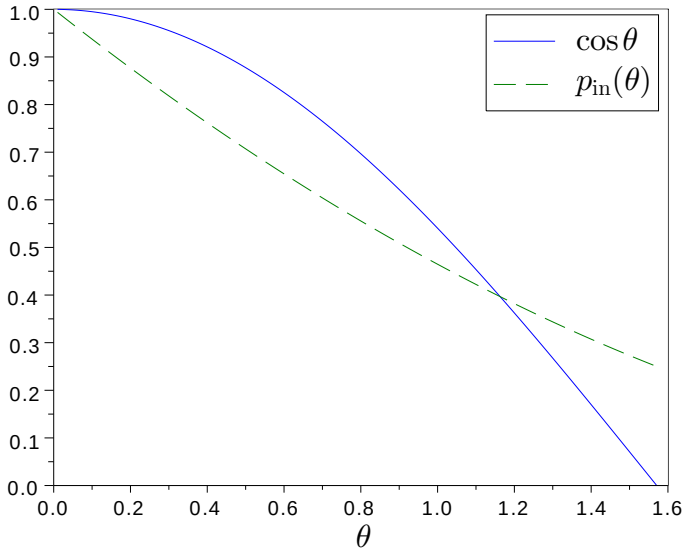
3(d) The resulting optimization problem can be verified to be semidefinite, and can be solved reasonably efficiently. The intuition is that if x_u and x_v are "close", u and v should be in the same partition. In fact, we will create a rounded solution with only four partitions. The four partitions will be induced by the intersection of two independent random hyperplanes passing through the origin in $\mathbb{R}^n$, with unit normal vectors q_1, q_2 uniformly distributed over the unit sphere. Suppose the angle between x_u and x_v is θ, i.e., $x_u^\top x_v = \cos\theta$. What is the probability $p_{\text{in}}(\theta)$, as a function of θ alone (and possibly cosmic constants), that u, v get assigned to the same partition?

		2

The probability that q_1 separates u and v is θ/π, and so the probability that neither q_1 nor q_2 separates them is $(1 - \theta/\pi)^2$.

3(e) Below we show a sketch of $p_{\text{in}}(\theta)$ and $\cos\theta$ against θ. Give a crude numeric lower bound on $\dfrac{p_{\text{in}}(\theta)}{\cos\theta}$ using eye estimation.

		1



Eye estimation suggests the value is at least 0.7. More careful numerical optimization gives 0.7895.

3(f) Similarly, provide an empirical lower bound to $\frac{p_{\text{out}}(\theta)}{1-\cos\theta}$ where $p_{\text{out}}(\theta) = 1 - p_{\text{in}}(\theta)$. (Hints: Can you bound $\frac{d}{d\theta}\frac{p_{\text{out}}(\theta)}{1-\cos\theta}$? Note that $\cos\theta \geq 1 - \theta^2/2$ and $\sin\theta \leq \theta$ for $\theta \in [0, \pi/2]$. What is the value of the ratio at $\pi/2$?)

		2

It can be shown that $p_{\text{out}}(\theta)/(1-\cos\theta)$ is monotonic decreasing with θ in the range $[0, \pi/2]$. At $\pi/2$ the value is $1 - p_{\text{in}}(\pi/2) = 1 - (1-1/2)^2 = 1 - 1/4 = 0.75$.

3(g) Let X_e be a random variable denoting the contribution of edge e to the correlation clustering objective. Note that the randomness here is over the choice of the two hyperplanes. Write down the expected correlation clustering objective as a function of X_es over all edges.

		2

$$\text{Let} \quad X_{uv} = \begin{cases} 0, & u, v \text{ are in the same cluster} \\ 1, & \text{otherwise} \end{cases}$$

Then our rounded objective is

$$\sum_{(u,v)} w_{\text{in}}(u,v) X_{uv} + \sum_{(u,v)} w_{\text{out}}(u,v)(1 - X_{uv})$$

3(h) What is the expected objective over the randomness in choosing the two hyperplanes, as compared to the objective attained by an optimal algorithm? Compare with the trivial algorithm (choose between a single cluster and one cluster per node).

		3

$$\mathbb{E}\left[\sum_{(u,v)} w_{\text{in}}(u,v)X_{uv} + \sum_{(u,v)} w_{\text{out}}(u,v)(1-X_{uv})\right]$$
$$= \sum_{(u,v)} w_{\text{in}}(u,v)p_{\text{in}}(\theta_{uv}) + \sum_{(u,v)} w_{\text{out}}(u,v)p_{\text{out}}(\theta_{uv})$$
$$\geq 0.75\left[\sum_{(u,v)\in E} w_{\text{in}}(u,v)x_u^\top x_v + w_{\text{out}}(e)\left(1 - x_u^\top x_v\right)\right] \geq 0.75\,\text{opt}$$

Therefore we get an approximation ratio of at least 0.75, whereas the trivial algorithm got only 0.5.

4. Among the definitions of graph proximity is *hitting time*. The hitting time h_{ij} from node i to node j is defined as the expected number of steps in a random walk starting from i before node j is visited for the first time. The parameters of the random walk are $p_{ij} = \Pr(j|i)$.

4(a) Recursively, h_{ij} can be written as

$$h_{ij} = \begin{cases} \_\_\_\_\_ + \sum_{(k,j)\in E} \_\_\_\_\_\_\_\_\_\_\_\_, & \text{if } i \_\_\_\_\_ j \\ 0 & \text{otherwise} \end{cases}$$

Complete with justification. 3

$$h_{ij} = \begin{cases} 1 + \sum_{(i,k)\in E} p_{ik}h_{kj}, & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases}$$

4(b) To avoid spurious long-range effects, a truncated version of the hitting time is also used. With the truncation distance set to T links, modify the above expression as follows:

$$h_{ij}^T = \begin{cases} 0, & i \_\_\_\_\_ j \\ 0, & T = \_\_\_\_\_ \\ \_\_\_\_\_ + \sum_{(i,k)\in E} \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, & \text{otherwise} \end{cases}$$

We will call h^T the "T-truncated hitting time". 2

$$h_{ij}^T = \begin{cases} 0, & i = j \\ 0, & T = 0 \\ 1 + \sum_{(i,k)\in E} p_{ik}h_{kj}^{T-1}, & \text{otherwise} \end{cases}$$

4(c) If there is no path of length at most T from i to j, what is the value of h_{ij}^T? 1

T

4(d) For any node i in a graph, the number of nodes that have a T-truncated hitting time within τ is at most $\frac{T^2}{\tau}$.

		1

Let j be a node that is within a T-truncated hitting time of τ. Let $P_{ij}^{<T}$ denote the probability of hitting node j starting at i in fewer than T steps and $\tilde{P}_{ij}^{t}$ the probability of hitting j in exactly t steps for the first time starting from i. Then

- $h_{ij} \leq \tau$ by assumption
- $h_{ij} \geq T(1 - P_{ij}^{<T})$ (lhs has additional nonnegative terms)

Combining, we get $T(1 - P_{ij}^{<T}) \leq \tau$, leading to $P_{ij}^{<T} \geq \frac{T-\tau}{T}$. Next, define $S(i,\tau)$ as the neighborhood of i consisting of nodes within hitting time τ from i. Then

$$|S(i,\tau)|\frac{T-\tau}{T} \leq \sum_{j \in S(i,\tau)} P_{ij}^{<T} = \sum_{j \in S(i,\tau)} \sum_{t=1}^{T-1} \tilde{P}_{ij}^{t} = \sum_{t=1}^{T-1} \sum_{j \in S(i,\tau)} \tilde{P}_{ij}^{t} \leq T-1.$$

from which we get

$$|S(i,\tau)| \leq \frac{T(T-1)}{T-\tau}.$$

4(e) If there are n nodes and m edges in the graph, what is the time required to compute the truncated hitting time from all nodes to a given destination node, using dynamic programming?

		1

$O(mT)$ time.

4(f) What is the time required to compute the truncated hitting time from a given source node to a given destination node using dynamic programming?

		1

$O(mT)$ time.

4(g) What is the time required to compute the truncated hitting time from a given source node to all nodes using dynamic programming?

		1

$O(nmT)$ time.

4(h) We run M independent T-length random walks starting at node i. Out of these, suppose m walks ever visit j, hitting it at time steps $t_1, \ldots, t_m$. Complete the expression for the unbiased estimator

$$\hat{h}_{ij}^{T} = \frac{1}{M}\sum_{r=1}^{m} \underline{\quad} + \left(1 - \frac{\quad}{M}\right) T$$

		3

$$\hat{h}_{ij}^{T} = \frac{1}{M}\sum_{r=1}^{m} t_r + \left(1 - \frac{m}{M}\right) T$$

4(i) For the rth trial, $r \in \{1, \ldots, M\}$, let $X(r, i, j)$ be a random variable representing the first arrival time at j starting at i. Define $X(r, i, j) = T$ if the walk from i never reaches j. Then $\hat{h}^T(i, j) = \frac{1}{M} \sum$ _____ and $\mathbb{E}(\hat{h}^T(i, j)) =$ _____. Complete with justification.

		3

$$\hat{h}^T(i, j) = \frac{1}{M} \sum_r X(r, i, j)$$

$$\mathbb{E}(\hat{h}^T(i, j)) = h^T(i, j)$$

4(j) For $X_r \in \{1, \ldots, T\}$, $r = 1, \ldots, M$, and $X = \sum_r X_r$, the Hoeffding bound states that $\Pr\left(\left|\hat{h}^T(ii, j) - h^T(i, j)\right| > \epsilon T\right) \leq 2 \exp\left(-2M\epsilon^2\right)$. How many walks M should we perform so that, for a given source node i, the estimates of truncated hitting time to *all* nodes are within ϵT of the correct value with probability at least $1 - \delta$?

		3

We want $2n \exp(-2M\epsilon^2) \leq \delta$, in other words, $\exp(2M\epsilon^2) \geq 2n/\delta$, or $M \geq \frac{1}{2\epsilon^2} \log \frac{2n}{\delta}$. But this is pretty pessimistic in practice. However, for large graphs, this is still much smaller than dynamic programming.

Total: 60