Course: CSC591/791, Graph Data Mining: Theory, Algorithms, and Applications
Final Exam: Comprehensive, home-take exam.
STUDENT's ID: avshirod

This exam is intended to be an **individual** effort. You are allowed to use and reference the materials posted on Moodle, cited in the exam, your homework assignments, your notes, the class slides, and the lecture assignments. You are **NOT** allowed to use the other sources, such as your classmates and the internet at large.

1. Good luck!
2. You get BONUS 5 points if your solution is in LaTeX (both source and PDF)
3. Otherwise, a PDF (even a scanned hand-written solution) must be uploaded into Moodle
4. The Total Exam Score by the TA: ______ out of **(100 points)**

From the NCSU Code of Student Conduct:

DEFINITIONS OF ACADEMIC DISHONESTY

1. Academic dishonesty is the giving, taking, or presenting of information or material by a student that unethically or fraudulently aids oneself or another on any work which is to be considered in the determination of a grade or the completion of academic requirements or the enhancement of that student's record or academic career.
2. A student shall be guilty of a violation of academic integrity if he or she:
   - represents the work of others as his or her own;
   - obtains assistance in any academic work from another individual in a situation in which the student is expected to perform independently;
   - gives assistance to another individual in a situation in which that individual is expected to perform independently;
   - offers false data in support of laboratory or field work.

3. The act of submitting work for evaluation or to meet a requirement is regarded as assurance that the work is the result of the student's own thought and study, produced without assistance, and stated in that student's own words, except as quotation marks, references, or footnotes acknowledge the use of other sources. Submission of work used previously must first be approved by the instructor.

4. Regulations regarding academic dishonesty are set forth in writing in order to give students general notice of prohibited conduct. They should be read broadly and are not designed to define academic dishonesty in exhaustive terms.

5. If a student is in doubt regarding any matter relating to the standards of academic integrity in a given course or on a given assignment, that student shall consult with the faculty member responsible for the course before presenting the work.

By signing this exam, the student acknowledges the above terms and agrees to abide by NCSU policies on academic integrity.

Agreed: *avshirod* Date: *December 1st, 2016*

1. ________ out of **(15 points)**: Given an infectious disease following the SIS virus propagation model with transmission probability $\beta = 0.3$ and healing probability $\delta = 0.5$, in a contact network with 10 nodes and the following edges: $\{(0, 1), (0, 3), (1, 4), (2, 3), (2, 6), (5, 6), (5, 7), (6, 7), (7, 9), (8, 9)\}$, answer:

   (a) ________ out of **(1 point)**: What is the spectral radius of the network?

   (b) ________ out of **(1 point)**: What is the effective strength of the virus?

   (c) ________ out of **(1 point)**: According to the theorem studied in class [1], can the infection result in an epidemic?

   (d) ________ out of **(9 points)**: Using the NetShield algorithm [2], which 3 nodes should be immunized to minimize the spread of the infection?

   (e) ________ out of **(3 points)**: After immunizing the 3 nodes selected in (4), can the infection result in an epidemic?

References:

1 B. A. Prakash, D. Chakrabarti, M. Faloutsos, N. Valler, C. Faloutsos. Got the Flu (or Mumps)? Check the Eigenvalue! arXiv:1004.0060 [physics.soc-ph], 2010.

2 H. Tong, B. A. Prakash, C. Tsourakakis, T. Eliassi-Rad, C. Faloutsos, D. H. Chau. On the Vulnerability of Large Graphs. In ICDM, 2010.

Ans:
VPM - SIS

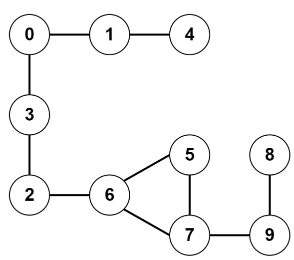


Figure 1: Visualizing the SIS network graph

$\beta = 0.3$ # Nodes $n = 10$
$\delta = 0.5$ # Edges $e = 10$

(a) Spectral Radius = 2.373 (Max eigenvalue of AdjMat - λ_1)

(b) Effective Strength
$S = \lambda_1 * C_{VPM} = \lambda_1 * \frac{\beta}{\delta} = 2.373 * \frac{0.3}{0.5} = 1.4238$

(c) Epidemic?
As effective strength $S = 1.4238 > 1$,
$\therefore$ Epidemic.

(d) 3 nodes that should be immunized to minimize the spread of the infection, as per *NetShield* algorithm $= \{6, 7, 5\}$

(e) Vaccinate above nodes. Epidemic now?
λ_1 of modified graph $= 1.732$
$S = \lambda_1 * C_{VPM} = 1.732 * 0.6 = 1.0392 > 1$
$\therefore$ Still Epidemic.

NOTE:
Although the NetShield algorithm implemented below is giving nodes $[6, 7, 5]$ as the output, it is not the optimum output (because it does not stop the epidemic.
If we vaccinate nodes $[6, 7, 0]$ instead, then we get eigenvalue as 1 and $S = 0.6 < 1$. And we'll be able to stop the epidemic.

Listing 1: Code for NetShield Algorithm

```
import networkx as nx
import numpy as np
from scipy.sparse import linalg

g = nx.Graph()
g = nx.read_edgelist('q1edges.txt', nodetype=int)

def get_largest_eigens(graph):
    adj = nx.to_scipy_sparse_matrix(graph, dtype=
        float)
    eigenvalue, eigenvector = linalg.eigs(adj.T, k=1,
         which='LR')
    return eigenvalue, eigenvector
# largest = u1.flatten().real

g.A = nx.adjacency_matrix(g)
g.eig = nx.linalg.adjacency_spectrum(g)
lambda_1, mu_1 = get_largest_eigens(g)

def shield_value_score(graph, node):
    eigenval, eigenvec = get_largest_eigens(graph)
    eigenvector = eigenvec.flatten()
    score = 2 * eigenval.real * (eigenvector[node].
        real ** 2)
    return score
```

```python
def netshield(graph, k):
    lambda_1, mu_1 = get_largest_eigens(graph)
    svs = [shield_value_score(graph, node) for node
        in graph.nodes()]
    s = []
    while (len(s) < k):

        def get_netshield_score(node):
            adj = nx.adjacency_matrix(graph)
            score = svs[node] - 2 * (adj[:,s] * mu_1[
                s].real) * mu_1[node].real
            return score

        scores = np.array([get_netshield_score(node)
            for node in graph.nodes()])
        scores[s] = -99999
        scores = scores.tolist()
        s.append(scores.index(max(scores)))
    return s

k = 3
to_vaccinize = netshield(g, 3)
print("Nodes to Vaccinize = %r" % to_vaccinize)

g_vacc = g.copy()
g_vacc.remove_nodes_from(to_vaccinize)
g_vacc.eig = nx.linalg.adjacency_spectrum(g_vacc)
print("Lambda_1 after Vaccination  = %r" % max(g_vacc
    .eig))

t = g.copy()
v = [6,7,0]
print("Nodes to Vaccinize = %r" % v)
t.remove_nodes_from(v)
print("Lambda_1 after Vaccination  = %r" % max(nx.
    linalg.adjacency_spectrum(t)))

'''
Output:
Nodes to Vaccinize = [6, 7, 5]
Lambda_1 after Vaccination  = (1.7320508075688774+0j)
Nodes to Vaccinize = [6, 7, 0]
Lambda_1 after Vaccination  = (1+0j)
'''
```

2. ________ out of **(15 points)**: Given random variables x, y, w, z, **prove or disprove** the following statements about their conditional independence ($\perp$) relationships (see the lecture on D-separation and example below; if you decide to disprove the stmt, then a counter-example of a DAG will suffice):

(a) ________ out of **(0 points)**: Example: $(x \perp y, w | z) \implies (x \perp y | z)$

Proof:

$$p(x, y|z) = \sum_w p(x, y, w|z) = \sum_w p(x|z)p(y, w|z) =$$
$$p(x|z) \sum_w p(y, w|z) = p(x|z)p(y|z)$$

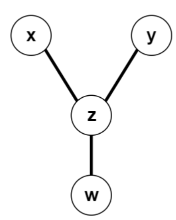
(b) ________ out of **(6 points)**: $(x \perp y, z) \& (x, y \perp w | z) \implies (x \perp w | z)$

(c) ________ out of **(9 points)**: $(x \perp y, z) \& (x \perp y | w) \implies (x \perp y | z, w)$

Ans:

(a) Proved.

(b) **TPT:** $(x \perp y, z) \& (x, y \perp w | z) \implies (x \perp w | z)$



$$x \perp y, z \implies P(x|y, z) = P(x) \qquad (1)$$
$$x, y \perp w | z \implies P(x, y|w, z) = P(x, y|z) \qquad (2)$$

Now,

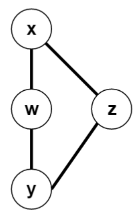
$$\begin{aligned} P(x|w, z) &= \sum_y P(x, y|w, z) && \text{Introducing sum over 'y'} \\ &= \sum_y P(x, y|z) && \text{from (1)} \\ &= P(x|z) && \text{from (2)} \end{aligned}$$

$\therefore x \perp w | z$

(c) **TPT:** $(x \perp y, z) \& (x \perp y | w) \implies (x \perp y | z, w)$

$$x \perp y, z \implies P(x|y, z) = P(x) \tag{1}$$
$$x \perp y | w \implies P(x|y, w) = P(x|w) \tag{2}$$

Now consider the following graph -



In above graph, the second condition does not hold True, as not all paths from x to y go through w $(x - z - y)$.

$\therefore$ *Disproved.*

3. ______ out of (**15 points**): Suppose you need to design an efficient algorithm for analysizing random walks of an undirected simple graph G. Let A be its adjacency matrix with Singular Valude Decomposition (SVD) $A = U\Lambda U^t$.

   (a) ______ out of **(5 points)**: Prove that the power matrix $A^k = U\Lambda^k U^t$.

   (b) ______ out of **(10 points)**: Devise an efficient algorithm to compute A^k for different values of k. What is its Big-O complexity?

Ans:
Random walks on Undirected Simple Graphs.
$A =$ Adj Matrix
$A = U\Lambda U^t$

A is a symmetric matrix $\implies A = A^T$.
Also, A is a positive, semi-definite matrix $\implies$ all λ_i are real and $\lambda_i \geq 0$

(a) Prove: $A^k = U\Lambda^k U^t$

As U and U^T are orthonormal matrices, (SVD property)
$\therefore U * U^T = I$ (Also $U^T * U = I$).

For $k = 2$,

$$\begin{aligned} A^2 = A * A &= U\Lambda U^T * U\Lambda U^T \\ &= U\Lambda I \Lambda U^T \\ \therefore A^2 &= U\Lambda^2 U^T \end{aligned}$$

Similarly for $k = 3$,

$$\begin{aligned} A^3 = A^2 * A &= U\Lambda^2 U^T * U\Lambda U^T \\ &= U\Lambda^2 I \Lambda U^T \\ \therefore A^3 &= U\Lambda^3 U^T \end{aligned}$$

By induction, we can say that -
$A^k = U\Lambda^k U^t$

(b) A pseudo code for the algorithm is as follows -

Listing 2: An efficient algorithm to compute A^k

```
# A is a n*n matrix
def power_mat(A, k):
        if k is even:
                a_square = A*A
                return pow(2,k/2) * a_square
        else:
                return pow(2,(k-1)/2) * A
```

As A is a symmetric square matrix, and it represents an undirected simple graph (with no cycles or self loops), we can mathematically see that, powers of A are scalar multiples of either A or A^2.
After computing powers of A upto A^8 for some random examples, it was observed that if k is even, then $A^k = 2^{k/2} * A^2$.
Otherwise (when k is odd), $A^k = 2^{(k-1)/2} * A$.

If k is even, then we have to compute $A*A$, which is a matrix multiplication of nxn size matrix. So the overall complexity of the algorithm is $O(n^{2.371})$ (using Strassen's modified algorithm for matrix multiplication).
But as $k \to n$, the general algorithm will have a complexity of $O(n^{3.371})$, but the algorithm presented above will requires the same amount of time.

4. ________ out of (**10 points**): Let G be an undirected complete bi-partitie graph. Prove that if λ is an eigenvalue of its adjacency matrix then $-\lambda$ is also its eigenvalue. Hint: Assume that after the permutation of rows and columns, the adjacency matrux is of the form:

$$A = \begin{pmatrix} 0 & B \\ B^t & 0 \end{pmatrix}$$

Ans:
G: Undirected Complete Bi-partite graph
A: Adjacency matrix
For a complete undirected bi-partite graph G ($K_{m,n}$), the adjacency matrix would look like -

$$A = \begin{pmatrix} 0 & B \\ B^t & 0 \end{pmatrix}$$

The dimensions of B would be (m, n).

As told in lecture on graph spectra (Lecture #08), the graph spectra for such a graph is $= \pm\sqrt{m * n}$.
with all others eigenvalues zero except λ_{max} and λ_{min}.

Let's assume we have λ as one eigenvalue of A.
Now, as G is undirected bi-partite, all the eigenvalues must sum up to zero. ($Trace(A) = 0$).
$\therefore$ The sum of the rest of the eigenvalues must be equal to $-\lambda$.

As the two elements in adjacency matrix A are symmetric (B and B^T), we can see that there will be two distinct eigenvalues, and the rest will be zero.
Now, as one of the eigenvalues is λ (fixed because of our assumption), there remains only one other distinct value. And as the eigenvalues should sum up to zero, it has to be $-\lambda$.

5. ______ out of (**20 points**): Consider the following training examples:

Instance	x_1	x_2	x_3	class, y
1	T	T	T	+
2	T	T	T	+
3	F	F	T	+
4	F	T	F	-
5	T	F	T	-
6	T	F	F	-
7	F	F	F	-
8	F	T	F	-

(a) ______ out of **(10 points)**: You will be using a naïve Bayes classifier for this question. Given a test example with attributes $x_1 = T$, $x_2 = T$, and $x_3 = F$, which class will be assigned to this test example? Show your work clearly.

(b) ______ out of **(10 points)**: Repeat the question in part (a) using the following Bayesian Belief network as the classifier:

$$x_3 \to x_2 \to y \to x_1$$

Ans:

(a) $x : x_1, x_2, x_3$ $\qquad n = 8$
$y : +, -$
$P(x_1 = T, x_2 = T, x_3 = F, y = +) =?$
Let's assume value of y to be $+$.

$$P(y = +) = 3/8 = 0.375$$
$$P(y = -) = 5/8 = 0.625$$

$$\begin{aligned} P(x_1 = T, y = +) &= P(y = +) * P(x_1 = T|y = +) \\ &= 0.375 * 2/3 = 0.250 \end{aligned}$$

$$\begin{aligned} P(x_2 = T, y = +) &= P(y = +) * P(x_2 = T|y = +) \\ &= 0.375 * 2/3 = 0.250 \end{aligned}$$

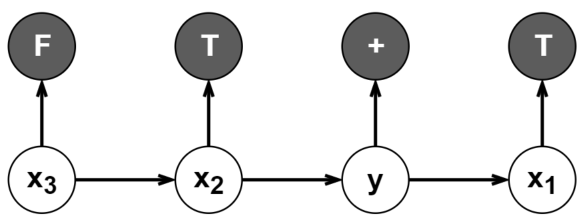
$$\begin{aligned} P(x_3 = T, y = +) &= P(y = +) * P(x_3 = F|y = +) \\ &= 0.375 * 0 = 0 \end{aligned}$$

Now, the probability of observed data = 1.
$\therefore P(x_1 = T, x_2 = T, x_3 = F) = 1$

$$\begin{aligned}
P(y = +|x_1 = T, x_2 = T, x_3 = F) &= \frac{P(x_1 = T, x_2 = T, x_3 = F, y = +)}{P(x_1 = T, x_2 = T, x_3 = F)} \\
&= P(y = +) * P(x_1 = T, x_2 = T, x_3 = F|y = +) \\
&= P(y = +) * P(x_1 = T|y = +) * \\
&\quad P(x_2 = T|y = +) * P(x_3 = F|y = +) \\
&= 0.375 * 0.250 * 0.250 * 0 \\
&= 0
\end{aligned}$$

$\therefore P(y = +|T, T, F) = 0$
$\therefore P(y = -|T, T, F) = 1$
$\therefore$ Classify as $y =' -'$.

(b) $P(y|x_3 = F, x_2 = T, x_1 = T) =?$



Let $X : x_3 = F, x_2 = T, x_1 = T$
We know that $P(X) = 1$, as it is observed.

Now,

$$\begin{aligned}
P(y = +|X) &= \frac{P(y = +, X)}{P(X)} \\
&= P(x_3 = F)P(x_2 = T|x_3 = F)P(y = +|x_2 = T)P(x_1 = T|y = +) \\
&= 4/8 * 2/4 * 2/4 * 2/3 = 1/12 \\
&= 0.0833
\end{aligned}$$

$$\begin{aligned}
P(y = -|X) &= \frac{P(y = -, X}{X} \\
&= P(x_3 = F)P(x_2 = T|x_3 = F)P(y = -|x_2 = T)P(x_1 = T|y = -) \\
&= 4/8 * 2/4 * 2/4 * 2/5 = 1/20 \\
&= 0.05
\end{aligned}$$

As $P(y = +|X) > P(y = -|X)$, classify as '+'.

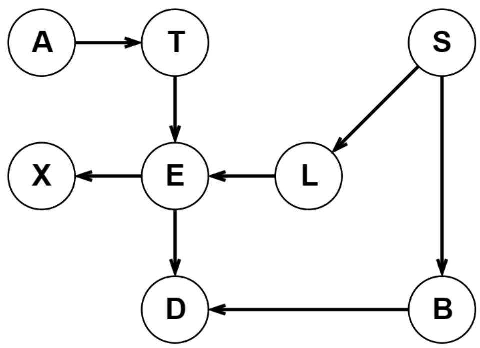
6. ________ out of (**10 points**): Show the key steps of the Junction Tree algorithm for the following DAG (Note: No need to show the details of the last step, i.e., the Belief Propagation inference step):

$$A \to T \to E \to D; S \to L \to E \to X; S \to B \to D$$

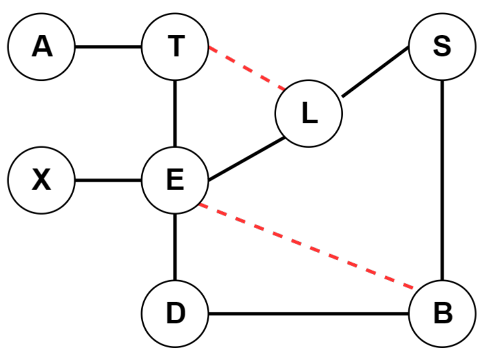
Ans:

Junction Tree Algorithm

(a) Start with DAG

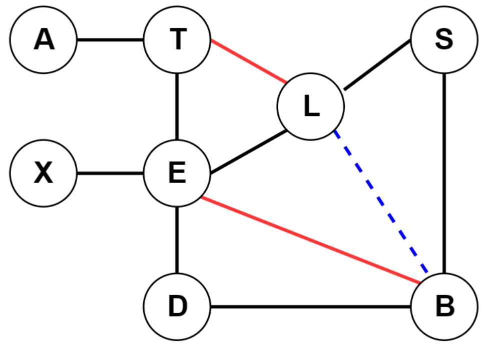


(b) Convert DAG $\to$ Undirected Graph (with Moralization)



(c) Triangulation

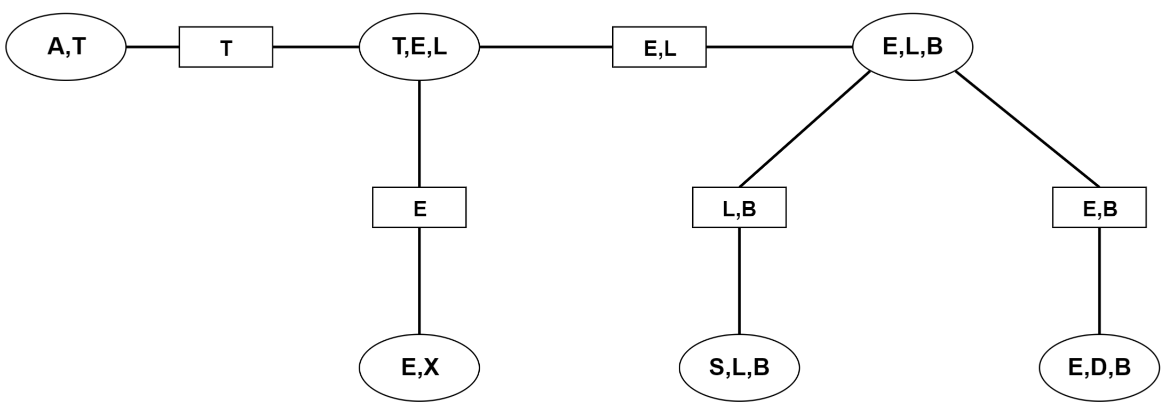
Cycle with more than 3 nodes $= S - B - E - L - S$
Converted to a triangle by adding edge $B - L$.



(d) Find cliques

$$
\begin{aligned}
c_1 &= A, T \\
c_2 &= T, E, L \\
c_3 &= E, X \\
c_4 &= S, L, B \\
c_5 &= E, D, B \\
c_6 &= E, L, B
\end{aligned}
$$

(e) Create a junction tree



7. ______ out of (**15 points**): Using the Maximum Likelihood estimation, find the parameters for the following Bernoulli Trials problem (show all the derivations):
INPUT:

- M iid coin flips: $D = H, H, T, H, ...$
- Model: $p(H) = \theta$ and $p(T) = 1 - \theta$

OUTPUT: θ^*_{ML}

(a) ______ out of **(10 points)**: What is the likelihood function $l(\theta; D) = \log p(D|\theta)$? (Hint: Introduce a random variable x that is equal to 1 if the trial is a Head and it is equal to zero if the trial is a Tail.)

(b) ______ out of **(5 points)**: What is θ^*_{ML}?

Ans:

MLE - Bernoulli trials
i/p: 'M' `i.i.d.` coin flips
$D = H, H, T, H$
model: $P(H) = \theta$ and $P(T) = 1 - \theta$
o/p : θ^*_{ML}

(a) For Bernoulli trials,
$f(x) = p^x * (1-p)^(1-x)$
$x = 0, 1$

Converting $D = H, H, T, H, ... \rightarrow D = x_1, x_2, ..., x_m$
by assuming $x_i = 1$ if $D_i = H$
and $x_i = 0$ if $D_i = T$
$\therefore D_{new} = 1, 1, 0, 1, ...$
Now, based on assumptions above -

$$\begin{aligned} P(\theta|x) &= \theta^{x_1}(1-\theta)^{1-x_1} \ldots \theta^{x_m}(1-\theta)^{1-x_n} \\ &= \theta^{x_1+\ldots+x_m}(i-\theta)^{n-(x_1+\ldots+x_m)} \end{aligned}$$

$$\begin{aligned} \ln L(\theta|x) = \log P(\theta|D) &= \ln\theta \sum_{i=1}^{M} x_i + \ln(1-\theta)(n * \sum_{i=1}^{M} x_i) \\ &= n\bar{x}\ln\theta + n(1-\bar{x})\ln 1 - \theta \end{aligned}$$

(b) To find θ^*_{ML}, taking partial derivative wrt θ, -

$$\frac{\partial \ln L(\theta|x)}{\partial\theta} = n\left(\frac{\bar{x}}{\theta} + \frac{1-\bar{x}}{1-\theta}\right) = 0$$
$$\therefore \frac{\bar{x}}{\theta} = \frac{1-\bar{x}}{1-\theta}$$
$$\therefore \bar{x} - \bar{x}\theta = \theta - \bar{x}\theta$$
$$\therefore \theta^* = \bar{x}$$

Since $x = 0, 1$,
taking average/mean gives us the probability θ for $x = H$,
and $1 - \theta$ for $x = T$.

Intuitively, say for $M = 5$,

$$D = H, H, T, H, T \implies x_i = 1, 1, 0, 1, 0$$
$$P(x = H) = \frac{3}{5} \implies \bar{x} = \frac{\sum_{i=1}^{M} x_i}{M} = \frac{1+1+0+1+0}{5} = \frac{3}{5}$$
$$\text{and } P(x = T) = 1 - \frac{3}{5} = \frac{2}{5} \implies 1 - \bar{x} = \frac{2}{5}$$