

Solution Sketches
Exam3
COSC 6342 *Machine Learning*

April 30, 2009

Your Name:

Your SSN:

Problem 1 [5]: Information Gain

Problem 2 [8]: Ensemble Methods

Problem 3 [11]: Reinforcement Learning

Problem 4 [13]: Computations in Belief Networks /D-separation

Problem 5 [13]: Kernel Methods

Problem 6 [7]: Support Vector Machines

Problem 7 [7]: K-means

Problem 8 [6]: Gaussian Kernel Density Estimation

Σ [70]:

Grade:



The exam is “open books and notes” and you have 80 minutes to complete the exam. The exam is slightly too long; you are expected to solve only 90% of the problems in the allocated time. The exam will count about 26% towards the course grade.

1) Information Gain and Entropy [5]

Assume we have a classification problem involving 3 classes: professors, students, and staff members. There are 750 students, 150 staff members and 100 professors. All professors have blond hair, 50 staff members have blond hair, and 250 students have blond hair. Compute the information gain of the test “*hair_color='blond'*” that returns true or false. Just giving the formula that computes the information gain is fine; you do not need to compute the exact value of the formula! Use H as the entropy function in your formula (e.g. $H(1/3, 1/6, 1/2)$ is the entropy that 1/3 of the examples belong to class 1, 1/6 of the examples belong to class 2, and half of the examples belong to class 3). [5]

$$\text{Gain}(D, \text{blond}) = H(3/4, 3/20, 1/10) - 4/10 * H(5/8, 1/8, 1/4) - 6/10 * H(5/6, 1/6, 0)$$

2) Ensemble Methods [8]

a) Ensembles use multiple classifiers to make decisions. What properties should a set of base classifiers have to form a *good* ensemble? [2]

- **The key point is that the classifiers make different kind of errors which leads to a lower variance of the ensemble classifier.**
- **Accuracy: each classifier should have a somewhat okay accuracy**

b) What role does importance α_i of a classifier play in the AdaBoost algorithm—where is it used in the algorithm? Are high importance classifiers treated differently from low importance classifiers by AdaBoost; if yes, how? [4]

- **It describes the importance of a classifier $\alpha_i = (1/\sigma) \ln((1 - \epsilon_i) / \epsilon_i)$.**
- **It is used for two things:**
 - o **It is used in updating the weights of samples; more important classifiers will lead to more significant weight updates compared to classifiers of lesser importance.**
 - o **It determines the weight the individual classifier has in the decision making of the ensemble. AdaBoost is a linear combination of many weak classifiers. The higher importance classifier of some subset of examples will have higher weight in the decision making.**

$$\sum_{i=1}^T \alpha_i C_i(x)$$

b) Why are ensemble methods quite popular these days? [2]

- **Main Point: Good performance; high accuracy can be achieved, if “different” base classifiers can be found, even if the base classifier’s accuracy isn’t that high.**
- **Minor Point: Increase in processor speed and memory, made it feasible to use this computationally expensive approach.**

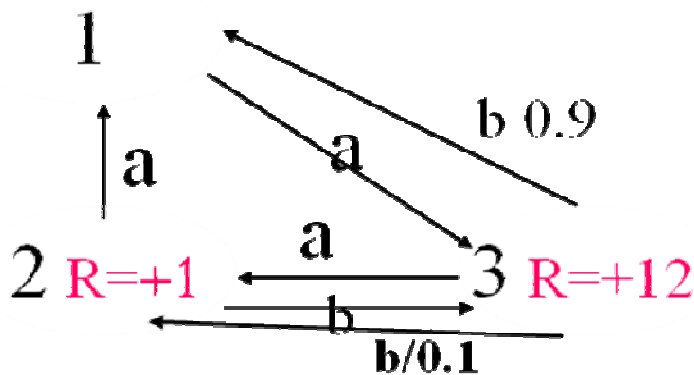
3) Reinforcement Learning [11]

a) Consider temporal difference learning and Bellman update; in which situations would you use which approach? [3]

- **If we know the transition probabilities and the reward for each state (have model assumption) and these two things do not change, the Bellman update should be used.**
- **Otherwise, temporal difference update should be used.**

b) What does the discount factor γ measure? [2]

- **The discount factor assesses the importance of an agent's future wellbeing.**



DEF World

c) Apply temporal difference learning to the DEF World, depicted above, relying on the following assumptions: [4]

- **The agent starts in state 3** and applies aaaa (applies action a 4 times)
- γ is 1.0 and α is 0.5
- If state 1 is visited a reward of 0 is obtained
- Utilities of the 3 states are initialized with 0

What are the utilities of states 1, 2, 3 after aaaa has been applied? Do not only give the final result but also how you derived the final result including formulas used!

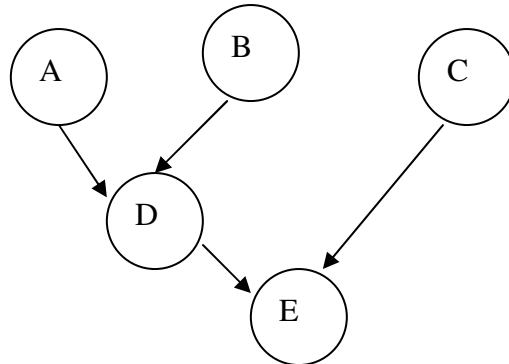
- **a:** $U_3 = U_3 + \alpha(R_3 + \gamma U_2 - U_3) = 0 + 0.5(12 + 0 - 0) = 6$
- **aa:** $U_2 = U_2 + \alpha(R_2 + \gamma U_1 - U_2) = 0 + 0.5(1 + 0 - 0) = 0.5$
- **aaa:** $U_1 = U_1 + \alpha(R_1 + \gamma U_3 - U_1) = 0 + 0.5(0 + 6 - 0) = 3$
- **aaaa:** $U_3 = U_3 + \alpha(R_2 + \gamma U_2 - U_3) = 6 + 0.5(12 + 0.5 - 6) = 9.25$

d) Give the Bellman equation for state 3 of the DEF world [2]

$$U(3) = 12 + \gamma * \max (U(2), 0.9*U(1)+ 0.1 * U(2))$$

4.) Computations in Belief Networks /D-separation [13]

Assume that the following Belief Network is given that consists of nodes A, B, C, D, and E that can take values of true and false.



a) Using the given probabilities of the probability tables of the above belief network ($P(D|A,B; E|C,D; A; B; C; \dots)$) give a formula to compute $P(E|B)$. Explain all nontrivial steps you used to obtain the formula! [9]

$P(E|B) = P(E, D | B) + P(E, \sim D|B)$ where:

- **$P(E, D|B) = P(D|B) \times P(E|D, B) = P(D|B) \times P(E|D)$**
//because $E \perp D$ and $B \perp D$ are d-separable
 - **$P(D|B) = P(D, A | B) + P(D, \sim A|B)$**
 $= P(D|A, B) \times P(A|B) + P(D|\sim A, B) \times P(\sim A|B)$
 $= P(D|A, B) \times P(A) + P(D|\sim A, B) \times P(\sim A)$
//because A and B are independent
 - **$P(E|D) = P(E, C|D) + P(E, \sim C|D)$**
 $= P(E|C, D) \times P(C|D) + P(E|\sim C, D) \times P(\sim C|D)$
 $= P(E|C, D) \times P(C) + P(E|\sim C, D) \times P(\sim C)$
- **Similarly, $P(E, \sim D|B) = P(\sim D|B) \times P(E|\sim D, B) = P(\sim D|B) \times P(E|\sim D)$**
 - **$P(\sim D|B) = P(\sim D|A, B) \times P(A) + P(\sim D|\sim A, B) \times P(\sim A)$**
 - **$P(E|\sim D) = P(E|C, \sim D) \times P(C) + P(E|\sim C, \sim D) \times P(\sim C)$**

b) Is $A|D,E$ d-separable from $C|D,E$? Give reasons for your answer! [2]

- **YES. There is only 1 path from A to C: A-D-E-C. They are d-separable according to the $A \rightarrow D \rightarrow E$ pattern with D being part of the evidence.**

c) Is A,B| \emptyset d-separable from C| \emptyset ? Give reasons for your answer! \emptyset :=”no evidence” [2]

- **YES. There are two paths: A-D-E-C and B-D-E-C; both paths are blocked due to the $D \rightarrow E \leftarrow C$ pattern with E not being part of the evidence**

5) Kernel Methods and k-means [13]

a) What is the “kernel trick”? [2]

- **The idea of the kernel trick is using linear or low dimensional algorithm in a higher dimensional space by mapping the original observations into a higher dimensional space; that is, although the algorithm operates in the higher dimensional space all computations are done in the lower dimensional leading to a more powerful algorithm without sacrificing performance.**

b) The ‘Top10 Data Mining algorithm article says that “you can ‘kernelize’ k-means”; what does this mean? How does a kernel k-means algorithm work? How is it different from the ordinary k-means algorithm? [5]

- **How does it work: Clusters data in the transformed space; examples e are mapped to a higher dimensional space by applying the kernel function mapping” $\Phi(e)$**
- **Different from K-means: mapping changes distance function; therefore algorithm can find clusters of non-convex shape;**

(*) *other answers should also receive credit.*

c) Assume k_1 and k_2 are kernel functions. Show that

$$k(u,v) = k_1(u,v) + k_2(u,v)$$

is a kernel function! [6]

Proof:

k_1 and k_2 are kernels; therefore mappings Φ_1 and Φ_2 exist:

- $k_1(u,v) = \langle \Phi_1(u), \Phi_1(v) \rangle$
- $k_2(u,v) = \langle \Phi_2(u), \Phi_2(v) \rangle$

Let $\Phi(u) = \Phi_1(u) + \Phi_2(u)$

$$\begin{aligned} \text{Let } k(u,v) &= \langle \Phi(u), \Phi(v) \rangle \\ &= k_1(u,v) + k_2(u,v) \\ &= \langle \Phi_1(u), \Phi_1(v) \rangle + \langle \Phi_2(u), \Phi_2(v) \rangle \\ &= \sum_i (\Phi_1^i(u) + \Phi_1^i(v)) + \sum_i (\Phi_2^i(u) + \Phi_2^i(v)) \\ &= \sum_i (\Phi_1^i(u) + \Phi_2^i(u)) + \sum_i (\Phi_2^i(v) + \Phi_1^i(v)) \\ &= \langle \Phi(u), \Phi(v) \rangle \end{aligned}$$

In summary, there is exist the mapping function $\Phi(u)=k_1(u,v) + k_2(u,v)$, with $k(u,v)= \langle \Phi(u), \Phi(v)\rangle$; therefore, k is a kernel.

****Other proofs may receive full credit.***

6) Support Vector Machine [7]

a) Why do most support vector machine approaches usually map examples to a higher dimensional space? [2]

- **Usually, the data in the original space is not linear separable, so applying a linear classifier like SVM may not be good. Therefore, most SVM approaches map examples to a higher dimensional space where the data become linear separable (or much fewer examples are misclassified by a linear classifiers that operates in the higher dimensional space).**

b) What role does the penalty factor C play in the soft margin hyperplane approach? How would select a value for C , when using the approach for a real world classification problem? [3]

- **The penalty factor C deals with the tradeoff between training accuracy and the width of the obtained hyperplane.**
 - o ***C is a penalty factor used for trading off complexity (number of support vectors) and data misfit (number of non-separable points).*** ← this answer of many students receives full credit too.
- **In real world classification, C can be selected by using cross-validation, usually maximizing accuracy.**

c) Does the optimization procedure always find the margin optimal hyperplane, assuming that examples are linearly separable? Give a reason for your answer! [2]

- **Yes. When the data is linear separable, convex quadratic optimization that is guaranteed to find the optimal solution.**

7) k-means [7]

a) K-means only finds clusters that are a local (but not a global) maximum of the objective function J it minimizes. Explain, why this is the case! [4]

- **Because the way it clusters the data is greedy – descent. It assigned each data point to its *closest* centroid, then updates the centroids and repeats the process.**

b) What can be said about the shapes of clusters k-means can find? [2]

- **Limited to convex polygons**

c) What is the storage complexity of k-means? [1]

- **$O(n)$ where n is the number of data points.**

8) Gaussian Kernel Density Estimation [6]

Assume we have a 2D dataset X containing 4 objects: $X = \{(1,0), (0,1), (1,2), (3,4)\}$; moreover, we use the Gaussian kernel density function to measure the density of X . Assume we want to compute the density at point $(1,1)$ and you can also assume $h=1$ ($\sigma=1$). Give a sketch how the Gaussian Kernel Density Estimation approach determines the density for point $(1, 1)$. Be specific!

$$\begin{aligned} \mathbf{x} &= (1, 1) \\ \mathbf{p1} &= (1,0); & \mathbf{x} - \mathbf{p1} &= (0, 1) \\ \mathbf{p2} &= (0, 1) & \mathbf{x} - \mathbf{p2} &= (1, 0) \\ \mathbf{p3} &= (1, 2) & \mathbf{x} - \mathbf{p3} &= (0, -1) \\ \mathbf{p4} &= (3, 4) & \mathbf{x} - \mathbf{p4} &= (-2, -3) \end{aligned}$$

$$\begin{aligned} N &= \text{number of data points} = 4 \\ d &= \text{number of dimensions of input space} = 2 \\ h &= 1 \end{aligned}$$

Density for point x

$$P(x) = \frac{1}{Nh^d} \sum_{i=1}^4 K\left(\frac{x - p_i}{h}\right) = \frac{1}{4} \sum_{i=1}^4 K(x - p_i)$$

Gaussian Kernel:

$$K(x) = \frac{1}{\sqrt{2\pi}^d} e^{-\frac{x^T \Sigma^{-1} x}{2}} = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

Therefore, the density d in point x is:

$$d(x) = \frac{1}{4\sqrt{2\pi}} \left(e^{-1/2} + e^{-1/2} + e^{-1/2} + e^{-13/2} \right) = \frac{1}{4\sqrt{2\pi}} \left(3e^{-1/2} + e^{-13/2} \right)$$

* If you give $d(x)$ but use different constants, you can still receive full credit.