

INSTRUCTIONS

- You have 3 hours.
- The exam is closed book, closed notes except a two-page crib sheet. Non-programmable calculators only.
- Mark your answers ON THE EXAM ITSELF. If not sure of your answer you may wish to provide a *brief* explanation.

First name	
Last name	
SID	
Login	
Name of the person to your left	
Name of the person to your right	

For staff use only:

Q1. The OMNIBUS	/24
Q2. HMM: Where is the key?	/21
Q3. Sampling	/9
Q4. Worst-Case Markov Decision Processes	/12
Q5. Tree-Augmented Naive Bayes	/19
Q6. Finding Working Kernels	/10
Q7. Learning a Ranking for Twoogle Hiring	/10
Total	/105

Q1. [24 pts] The OMNIBUS

Each question is worth 1 point. Leaving a question blank is worth 0 points. **Answering a multiple choice question with k possible choices incorrectly is worth $-1/(k-1)$ points (so -1 points for true/false questions, $-1/2$ for questions with three options, etc.). This gives you an expected value of 0 for random guessing.**

(a) [1 pt] **CS 188**

Circle the best motto for AI.

1. Maximize your expected utilities.

(b) [5 pts] **Search**

- (i) [true or false] Uniform-cost search will never expand more nodes than A*-search.
- (ii) [true or false] Depth-first search will always expand more nodes than breadth-first search.
- (iii) [true or false] The heuristic $h(n) = 0$ is admissible for every search problem.
- (iv) [true or false] The heuristic $h(n) = 1$ is admissible for every search problem.
- (v) [true or false] The heuristic $h(n) = c(n)$, where $c(n)$ is the true cheapest cost to get from the node n to a goal state, is admissible for every search problem.

(c) [2 pts] **CSPs**

- (i) [true or false] The most-constrained variable heuristic provides a way to select the next variable to assign in a backtracking search for solving a CSP.
- (ii) [true or false] By using the most-constrained variable heuristic and the least-constraining value heuristic we can solve every CSP in time linear in the number of variables.

(d) [3 pts] **Games**

- (i) [true or false] When using alpha-beta pruning, it is possible to get an incorrect value at the root node by choosing a bad ordering when expanding children.
- (ii) [true or false] When using alpha-beta pruning, the computational savings are independent of the order in which children are expanded.
- (iii) [true or false] When using expectimax to compute a policy, re-scaling the values of all the leaf nodes by multiplying them all with 10 can result in a different policy being optimal.

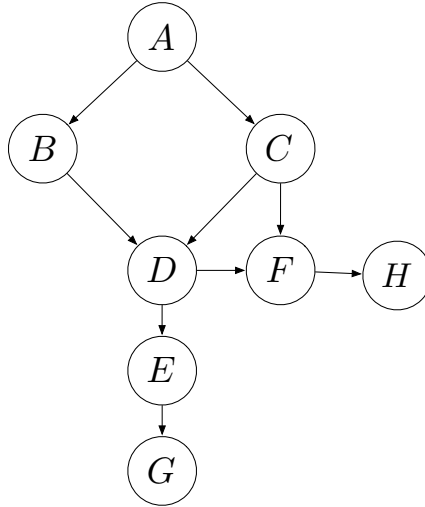
(e) [3 pts] **MDPs** For this question, assume that the MDP has a finite number of states.

- (i) [true or false] For an MDP (S, A, T, γ, R) if we only change the reward function R the optimal policy is guaranteed to remain the same.
- (ii) [true or false] Value iteration is guaranteed to converge if the discount factor (γ) satisfies $0 < \gamma < 1$.
- (iii) [true or false] Policies found by value iteration are superior to policies found by policy iteration.

(f) [2 pts] **Reinforcement Learning**

- (i) [true or false] Q-learning can learn the optimal Q-function Q^* without ever executing the optimal policy.
- (ii) [true or false] If an MDP has a transition model T that assigns non-zero probability for all triples $T(s, a, s')$ then Q-learning will fail.

(g) [8 pts] **Bayes' Nets** For each of the conditional independence assertions given below, circle whether they are guaranteed to be true, guaranteed to be false, or cannot be determined for the given Bayes' net.



$B \perp\!\!\!\perp C$	Guaranteed true	Guaranteed false	Cannot be determined
$B \perp\!\!\!\perp C \mid G$	Guaranteed true	Guaranteed false	Cannot be determined
$B \perp\!\!\!\perp C \mid H$	Guaranteed true	Guaranteed false	Cannot be determined
$A \perp\!\!\!\perp D \mid G$	Guaranteed true	Guaranteed false	Cannot be determined
$A \perp\!\!\!\perp D \mid H$	Guaranteed true	Guaranteed false	Cannot be determined
$B \perp\!\!\!\perp C \mid A, F$	Guaranteed true	Guaranteed false	Cannot be determined
$F \perp\!\!\!\perp B \mid D, A$	Guaranteed true	Guaranteed false	Cannot be determined
$F \perp\!\!\!\perp B \mid D, C$	Guaranteed true	Guaranteed false	Cannot be determined

Q2. [21 pts] HMM: Where is the key?

The cs188 staff have a key to the homework bin. It is the master key that unlocks the bins to many classes, so we take special care to protect it.

Every day John Duchi goes to the gym, and on the days he has the key, 60% of the time he forgets it next to the bench press. When that happens one of the other three GSIs, equally likely, always finds it since they work out right after. Jon Barron likes to hang out at Brewed Awakening and 50% of the time he is there with the key, he forgets the key at the coffee shop. Luckily Lubomir always shows up there and finds the key whenever Jon Barron forgets it. Lubomir has a hole in his pocket and ends up losing the key 80% of the time somewhere on Euclid street. However, Arjun takes the same path to Soda and always finds the key. Arjun has a 10% chance to lose the key somewhere in the AI lab next to the Willow Garage robot, but then Lubomir picks it up.

The GSIs lose the key at most once per day, around noon (after losing it they become extra careful for the rest of the day), and they always find it the same day in the early afternoon.

- (a) [2 pts] Draw on the left the Markov chain capturing the location of the key and fill in the transition probability table on the right. In this table, the entry of row JD and column JD corresponds to $P(X_{t+1} = \text{JD} | X_t = \text{JD})$, the entry of row JD and column JB corresponds to $P(X_{t+1} = \text{JB} | X_t = \text{JD})$, and so forth.

	JD	JB	LB	AS
JD				
JB				
LB				
AS			0.10	

Monday early morning Prof. Abbeel handed the key to Jon Barron. (The initial state distribution assigns probability 1 to $X_0 = \text{JB}$ and probability 0 to all other states.)

- (b) [4 pts] The homework is due Tuesday at midnight so the GSIs need the key to open the bin. What is the probability for each GSI to have the key at that time? Let X_0 , X_{Mon} and X_{Tue} be random variables corresponding to who has the key when Prof. Abbeel hands it out, who has the key on Monday evening, and who has the key on Tuesday evening, respectively. Fill in the probabilities in the table below.

	$P(X_0)$	$P(X_{\text{Mon}})$	$P(X_{\text{Tue}})$
JD	0		
JB	1		
LB	0		
AS	0		

- (c) [3 pts] The GSIs like their jobs so much that they decide to be professional GSIs permanently. They assign an extra credit homework (make computers truly understand natural language) due *at the end of time*. What is the probability that each GSI holds the key at a point infinitely far in the future. Hint:

$$P_{\infty}(x) = \sum_{x'} P(X_{\text{next day}} = x \mid X_{\text{current day}} = x')P_{\infty}(x')$$

Every evening the GSI who has the key feels obliged to write a short anonymous report on their opinion about the state of AI. Arjun and John Duchi are optimistic that we are right around the corner of solving AI and have an 80% chance of writing an optimistic report, while Lubomir and Jon Barron have an 80% chance of writing a pessimistic report. The following are the titles of the first few reports:

- Monday:** Survey: Computers Become Progressively Less Intelligent (pessimistic)
Tuesday: How to Solve Computer Vision in Three Days (optimistic)

- (d) [4 pts] In light of that new information, what is the probability distribution for the key on Tuesday midnight given that Jon Barron has it Monday morning? You may leave the result as a ratio or unnormalized.

On Thursday afternoon Prof. Abbeel noticed a suspiciously familiar key on top of the Willow Garage robot's head. He thought to himself, "This can't possibly be the master key." (He was wrong!) Lubomir managed to snatch the key and distract him before he inquired more about it and is the key holder Thursday at midnight (i.e., $X_{\text{Thu}} = \text{LB}$). In addition, the Friday report is this:

Thursday: ??? (report unknown)

Friday: AI is a scam. I know it, you know it, it is time for the world to know it! (pessimistic)

(e) [3 pts] Given that new information, what is the probability distribution for the holder of the key on Friday at midnight?

(f) [3 pts] Prof. Abbeel recalls that he saw Lubomir holding the same key on Tuesday night. Given this new information (in addition to the information in the previous part), what is the probability distribution for the holder of the key on Friday at midnight?

(g) [2 pts] Suppose in addition that we know that the titles of the reports for the rest of the week are:

Saturday: Befriend your PC now. Soon your life will depend on its wishes (optimistic)

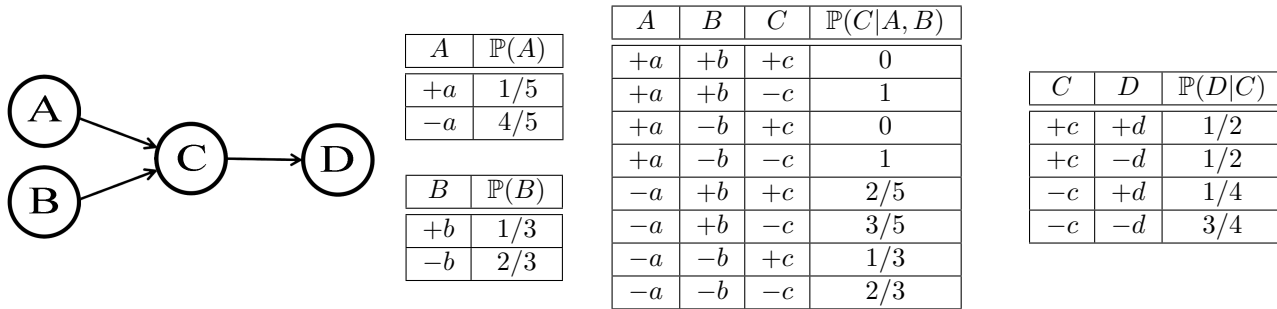
Sunday: How we got tricked into studying AI and how to change field without raising suspicion (pessimistic)

Will that new information change our answer to (f)? Choose one of these options:

1. Yes, reports for Saturday and Sunday affect our prediction for the key holder on Friday.
2. No, our prediction for Friday depends only on what happened in the past.

Q3. [9 pts] Sampling

Assume the following Bayes net, and the corresponding distributions over the variables in the Bayes net:



- (a) [1 pt] Your task is now to estimate $\mathbb{P}(+c | -a, -b, -d)$ using rejection sampling. Below are some samples that have been produced by prior sampling (that is, the rejection stage in rejection sampling hasn't happened yet). Cross out the samples that would be rejected by rejection sampling:

-a	-b	+c	+d	+a	-b	-c	-d
+a	-b	-c	+d	-a	+b	+c	+d
-a	-b	+c	-d	-a	-b	-c	-d

- (b) [2 pts] Using those samples, what value would you estimate for $\mathbb{P}(+c | -a, -b, -d)$ using rejection sampling?

- (c) [3 pts] Using the following samples (which were generated using likelihood weighting), estimate $\mathbb{P}(+c | -a, -b, -d)$ using likelihood weighting, or state why it cannot be computed.

-a	-b	-c	-d
-a	-b	+c	-d
-a	-b	+c	-d

- (d) [3 pts] Below are three sequences of samples. Circle **any** sequence that could have been generated by Gibbs sampling.

	Sequence 1		Sequence 2		Sequence 3
1 :	-a -b -c +d	1 :	-a -b -c +d	1 :	-a -b -c +d
2 :	-a -b -c +d	2 :	-a -b -c -d	2 :	-a -b -c -d
3 :	-a -b +c +d	3 :	-a -b +c +d	3 :	-a +b -c -d

Q4. [12 pts] Worst-Case Markov Decision Processes

Most techniques for Markov Decision Processes focus on calculating $V^*(s)$, the maximum expected utility of state s (the expected discounted sum of rewards accumulated when starting from state s and acting optimally). This maximum expected utility $V^*(s)$ satisfies the following recursive expression, known as the Bellman Optimality Equation:

$$V^*(s) = \max_a \sum_{s'} T(s, a, s') [R(s, a, s') + \gamma V^*(s')].$$

In this question, instead of measuring the quality of a policy by its expected utility, we will consider the worst-case utility as our measure of quality. Concretely, $L^\pi(s)$ is the minimum utility it is possible to attain over all (potentially infinite) state-action sequences that can result from executing the policy π starting from state s . $L^*(s) = \max_\pi L^\pi(s)$ is the optimal worst-case utility. In words, $L^*(s)$ is the *greatest lower bound* on the utility of state s : the discounted sum of rewards that an agent acting optimally is guaranteed to achieve when starting in state s .

Let $C(s, a)$ be the set of all states that the agent has a non-zero probability of transferring to from state s using action a . Formally, $C(s, a) = \{s' \mid T(s, a, s') > 0\}$. This notation may be useful to you.

(a) [3 pts] Express $L^*(s)$ in a recursive form similar to the Bellman Optimality Equation.

(b) [2 pts] Recall that the Bellman update for value iteration is:

$$V_{i+1}(s) \leftarrow \max_a \sum_{s'} T(s, a, s') [R(s, a, s') + \gamma V_i(s')]$$

Formally define a similar update for calculating $L_{i+1}(s)$ using L_i .

(c) [3 pts] From this point on, you can assume that $R(s, a, s') = R(s)$ (rewards are a function of the current state) and that $R(s) \geq 0$ for all s . With these assumptions, the Bellman Optimality Equation for Q-functions is

$$Q^*(s, a) = R(s) + \sum_{s'} T(s, a, s') \left[\gamma \max_{a'} Q^*(s', a') \right]$$

Let $M(s, a)$ be the *greatest lower bound* on the utility of state s when taking action a (M is to L as Q is to V). (In words, if an agent plays optimally after taking action a from state s , this is the utility the agent is guaranteed to achieve.) Formally define $M^*(s, a)$, in a recursive form similar to how Q^* is defined.

(d) [2 pts] Recall that the Q-learning update for maximizing expected utility is:

$$Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + \alpha \left(R(s) + \gamma \max_{a'} Q(s', a') \right),$$

where α is the learning rate, $(s, a, s', R(s))$ is the sample that was just experienced (“we were in state s , we took action a , we ended up in state s' , and we received a reward $R(s)$). Circle the update equation below that results in $M(s, a) = M^*(s, a)$ when run sufficiently long under a policy that visits all state-action pairs infinitely often. If more than one of the update equations below achieves this, select the one that would converge more quickly. Note that in this problem, we do not know T or C when starting to learn.

- (i) $C(s, a) \leftarrow \{s'\} \cup C(s, a)$ (i.e. add s' to $C(s, a)$)

$$M(s, a) \leftarrow (1 - \alpha)M(s, a) + \alpha \left(R(s) + \gamma \sum_{s' \in C(s, a)} \max_{a'} M(s', a') \right)$$
- (ii) $C(s, a) \leftarrow \{s'\} \cup C(s, a)$ (i.e. add s' to $C(s, a)$)

$$M(s, a) \leftarrow (1 - \alpha)M(s, a) + \alpha \left(R(s) + \gamma \min_{s' \in C(s, a)} \max_{a'} M(s', a') \right)$$
- (iii) $C(s, a) \leftarrow \{s'\} \cup C(s, a)$ (i.e. add s' to $C(s, a)$)

$$M(s, a) \leftarrow R(s) + \gamma \min_{s' \in C(s, a)} \max_{a'} M(s', a')$$
- (iv)
$$M(s, a) \leftarrow (1 - \alpha)M(s, a) + \alpha \min \left\{ M(s, a), R(s) + \gamma \max_{a'} M(s', a') \right\}.$$

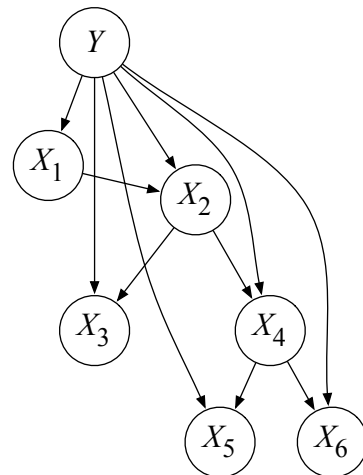
(e) [1 pt] Suppose our agent selected actions to maximize $L^*(s)$, and $\gamma = 1$. What non-MDP-related technique from this class would that resemble? (a one word answer will suffice)

(f) [1 pt] Suppose our agent selected actions to maximize $L_3(s)$ (our estimate of $L^*(s)$ after 3 iterations of our “value-iteration”-like backup in section *b*) and $\gamma = 1$. What non-MDP-related technique from this class would that resemble? (a brief answer will suffice)

Q5. [19 pts] Tree-Augmented Naive Bayes

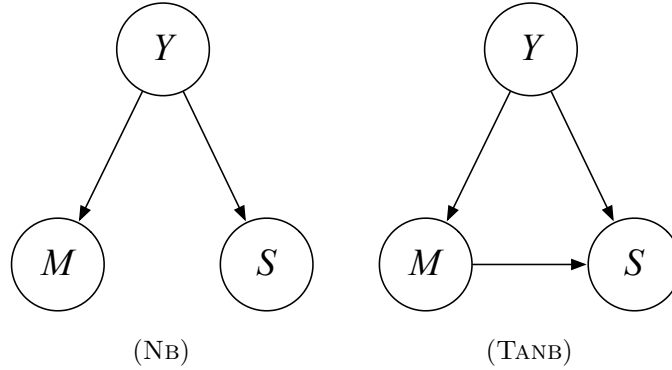
In section, we twice have tried to help Pacbaby distinguish his father, Pacman, from ghosts. Now Pacbaby has been transported back in time to the 1970s! Pacbaby has noticed that in the 1970s, nearly everyone who wears sunglasses also has a moustache, whether the person in question is Pacman, a ghost, or even a young Ms. Pacman. So Pacbaby decides that it's time for an upgrade from his Naive Bayes brain: he's getting a tree-augmented Naive Bayes brain so that the features he observes don't have to be independent.

In this question, we'll explore learning and inference in an abstraction of Pacbaby's new brain. A tree-augmented Naive Bayes model (TANB) is identical to a Naive Bayes model, except the features are no longer assumed conditionally independent given the class Y . Specifically, if (X_1, X_2, \dots, X_n) are the variables representing the features that Pacbaby can observe, a TANB allows X_1, \dots, X_n to be in a tree-structured Bayes net in addition to having Y as a parent. The example we explore is to the right.



- (a) [1 pt] Suppose we observe no variables as evidence in the TANB above. What is the classification rule for the TANB? Write the formula in terms of the CPTs (Conditional Probability Tables) and prior probabilities in the TANB.
- (b) [2 pts] Assume we observe all the variables $X_1 = x_1, X_2 = x_2, \dots, X_6 = x_6$ in the TANB above. What is the classification rule for the TANB? Write the formula in terms of the CPTs and prior probabilities in the TANB.
- (c) [3 pts] Specify an elimination order that is efficient for the query $\mathbb{P}(Y \mid X_5 = x_5)$ in the TANB above (including Y in your ordering). How many variables are in the biggest factor (there may be more than one; if so, list only one of the largest) induced by variable elimination with your ordering? Which variables are they?

- (d) [3 pts] Specify an elimination order that is efficient for the query $P(X_3 \mid X_5 = x_5)$ in the TANB above (including X_3 in your ordering). How many variables are in the biggest factor (there may be more than one; if so, list only one of the largest) induced by variable elimination with your ordering? Which variables are they?
- (e) [2 pts] Does it make sense to run Gibbs sampling to do inference in a TANB? In two or fewer sentences, justify your answer.
- (f) [2 pts] Suppose we are given a dataset of observations of Y and all the variables X_1, \dots, X_6 in the TANB above. Let C denote the total count of observations, $C(Y = y)$ denotes the number of observations of the event $Y = y$, $C(Y = y, X_i = x_i)$ denotes the count of the times the event $Y = y, X_i = x_i$ occurred, and so on. Using the C notation, write the maximum likelihood estimates for all CPTs involving the variable X_4 .
- (g) [2 pts] In the notation of the question above, write the Laplace smoothed estimates for all CPTs involving the variable X_4 (for amount of smoothing k).

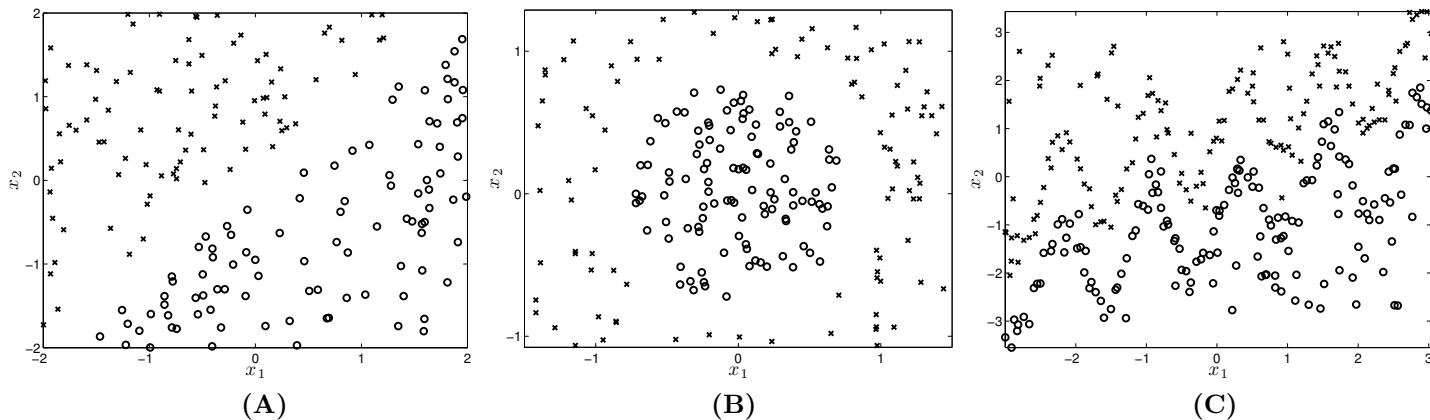


- (h) [2 pts] Consider the two graphs on the nodes Y (Pacbaby sees Pacman or not), M (Pacbaby sees a moustache), and S (Pacbaby sees sunglasses) above. Pacbaby observes $Y = 1$ and $Y = -1$ (Pacman or not Pacman) 50% of the time. Given $Y = 1$ (Pacman), Pacbaby observes $M = +m$ (moustache) 50% of the time and $S = +s$ (sunglasses on) 50% of the time. When Pacbaby observes $Y = -1$, the frequency of observations are identical (i.e. 50% $M = \pm m$ and 50% $S = \pm s$). In addition, Pacbaby notices that when $Y = +1$, anyone with a moustache also wears sunglasses, and anyone without a moustache does not wear sunglasses. If $Y = -1$, the presence or absence of a moustache has no influence on sunglasses. Based on this information, fill in the CPTs below (you can assume that Pacbaby has the true probabilities of the world).

For NB (left model)			For TANB (right model)																																								
<table border="1" style="margin: auto;"> <tr><th>y</th><th>$\mathbb{P}(Y = y)$</th></tr> <tr><td>1</td><td></td></tr> <tr><td>-1</td><td></td></tr> </table>			y	$\mathbb{P}(Y = y)$	1		-1		<table border="1" style="margin: auto;"> <tr><th>y</th><th>$\mathbb{P}(Y = y)$</th></tr> <tr><td>1</td><td></td></tr> <tr><td>-1</td><td></td></tr> </table>				y	$\mathbb{P}(Y = y)$	1		-1																										
y	$\mathbb{P}(Y = y)$																																										
1																																											
-1																																											
y	$\mathbb{P}(Y = y)$																																										
1																																											
-1																																											
<table border="1" style="margin: auto;"> <tr><th></th><th colspan="2">$\mathbb{P}(M = m Y = y)$</th></tr> <tr><th></th><th>$y = 1$</th><th>$y = -1$</th></tr> <tr><th>$m = 1$</th><td></td><td></td></tr> <tr><th>$m = -1$</th><td></td><td></td></tr> </table>				$\mathbb{P}(M = m Y = y)$			$y = 1$	$y = -1$	$m = 1$			$m = -1$			<table border="1" style="margin: auto;"> <tr><th></th><th colspan="2">$\mathbb{P}(M = m Y = y)$</th></tr> <tr><th></th><th>$y = 1$</th><th>$y = -1$</th></tr> <tr><th>$m = 1$</th><td></td><td></td></tr> <tr><th>$m = -1$</th><td></td><td></td></tr> </table>					$\mathbb{P}(M = m Y = y)$			$y = 1$	$y = -1$	$m = 1$			$m = -1$															
	$\mathbb{P}(M = m Y = y)$																																										
	$y = 1$	$y = -1$																																									
$m = 1$																																											
$m = -1$																																											
	$\mathbb{P}(M = m Y = y)$																																										
	$y = 1$	$y = -1$																																									
$m = 1$																																											
$m = -1$																																											
<table border="1" style="margin: auto;"> <tr><th></th><th colspan="2">$\mathbb{P}(S = s Y = y)$</th></tr> <tr><th></th><th>$y = 1$</th><th>$y = -1$</th></tr> <tr><th>$s = 1$</th><td></td><td></td></tr> <tr><th>$s = -1$</th><td></td><td></td></tr> </table>				$\mathbb{P}(S = s Y = y)$			$y = 1$	$y = -1$	$s = 1$			$s = -1$			<table border="1" style="margin: auto;"> <tr><th></th><th colspan="4">$\mathbb{P}(S = s Y = y, M = m)$</th></tr> <tr><th></th><th colspan="2">$y = 1$</th><th colspan="2">$y = -1$</th></tr> <tr><th></th><th>$m = 1$</th><th>$m = -1$</th><th>$m = 1$</th><th>$m = -1$</th></tr> <tr><th>$s = 1$</th><td></td><td></td><td></td><td></td></tr> <tr><th>$s = -1$</th><td></td><td></td><td></td><td></td></tr> </table>					$\mathbb{P}(S = s Y = y, M = m)$					$y = 1$		$y = -1$			$m = 1$	$m = -1$	$m = 1$	$m = -1$	$s = 1$					$s = -1$				
	$\mathbb{P}(S = s Y = y)$																																										
	$y = 1$	$y = -1$																																									
$s = 1$																																											
$s = -1$																																											
	$\mathbb{P}(S = s Y = y, M = m)$																																										
	$y = 1$		$y = -1$																																								
	$m = 1$	$m = -1$	$m = 1$	$m = -1$																																							
$s = 1$																																											
$s = -1$																																											

- (i) [2 pts] Pacbaby sees a character with a moustache and wearing a pair of sunglasses. What prediction does the Naive Bayes model NB make? What probability does the NB model assign its prediction? What prediction does the TANB model make? What probability does the TANB-brained Pacbaby assign this prediction? Which (if any) of the predictions assigns the correct posterior probabilities?

Q6. [10 pts] Finding Working Kernels



The above pictures represent three distinct two-dimensional datasets with positive examples labeled as o's and negative examples labeled as x's. Consider the following three kernel functions (where $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$):

- (i) Linear kernel: $K(x, z) = x^\top z = x \cdot z = x_1 z_1 + x_2 z_2$
- (ii) Polynomial kernel of degree 2: $K(x, z) = (1 + x^\top z)^2 = (1 + x \cdot z)^2$
- (iii) RBF (Gaussian) kernel: $K(x, z) = \exp\left(-\frac{1}{2\sigma^2} \|x - z\|^2\right) = \exp\left(-\frac{1}{2\sigma^2} (x - z)^\top (x - z)\right)$

(a) [6 pts] For each dataset (A, B, C) circle all kernels that make the dataset separable (assume $\sigma = .01$ for the RBF kernel):

Dataset (A): (i) (ii) (iii)

Dataset (B): (i) (ii) (iii)

Dataset (C): (i) (ii) (iii)

For parts (b) and (c), assume you train the perceptron using RBF (Gaussian) kernels: $K(x, z) = \exp\left(-\frac{1}{2\sigma^2} \|x - z\|^2\right)$. You run the perceptron algorithm on dataset (C) until you either encounter no more errors on the training data or you have encountered an error 1 million times and performed the associated update each time, whichever comes first.

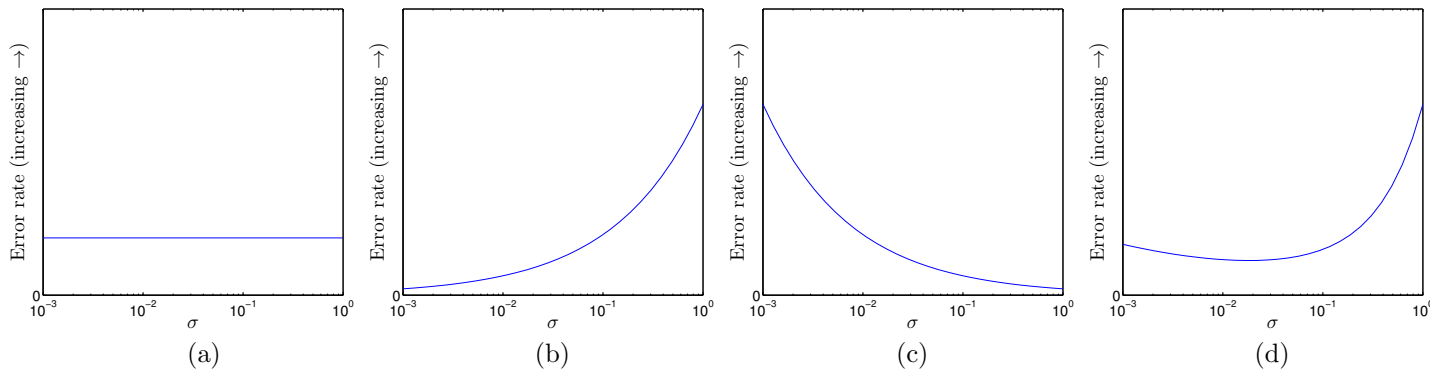


Figure 1: Possible plots of error rate (vertical axis) versus σ (horizontal axis)

(b) [2 pts] Which of the plots (a), (b), (c), or (d) in Fig. 1 is most likely to reflect the training set error rate of the learned classifier as a function of σ ?

(c) [2 pts] Which of the plots (a), (b), (c), or (d) in Fig. 1 is most likely to reflect the hold-out error rate as a function of σ ? Recall that “hold-out error-rate” is the error rate obtained by evaluating the classifier that was learned on training data on held-out (unused) data.

Q7. [10 pts] Learning a Ranking for Twoogle Hiring

You were just hired by Twoogle. Twoogle is expanding rapidly, and you decide to use your machine learning skills to assist them in their attempts to hire the best. To do so, you have the following available to you for each candidate i in the pool of candidates \mathcal{I} : (i) Their GPA, (ii) Whether they took CS164 with Hilfinger and achieved an A, (iii) Whether they took CS188 and achieved an A, (iv) Whether they have a job offer from GBook, (v) Whether they have a job offer from FacedIn, (vi) The number of misspelled words on their resume. You decide to represent each candidate $i \in \mathcal{I}$ by a corresponding 6-dimensional feature vector $f(x^{(i)})$. You believe that if you just knew the right weight vector $w \in \mathbb{R}^6$ you could reliably predict the quality of a candidate i by computing $w \cdot f(x^{(i)})$. To determine w your boss lets you sample pairs of candidates from the pool. For a pair of candidates (k, l) you can have them face off in a “twoogle-fight.” The result is $\text{score}(k \succ l)$, which tells you that candidate k is at least $\text{score}(k \succ l)$ better than candidate l . Note that the score will be negative when l is a better candidate than k . Assume you collected scores for a set of pairs of candidates \mathcal{P} .

- (a) [8 pts] Describe how you could use a perceptron-like algorithm to learn the weight vector w . Make sure to describe (i) Pseudo-code for the entire algorithm, (ii) In detail how the weight updates would be done.

- (b) [2 pts] You notice that your perceptron-like algorithm is unable to reach zero errors on your training data. You ask your boss if you could get access to more information about the candidates, but you are not getting it. Is there anything else you could do to potentially improve performance on your training data?