

6.867 Machine learning

Final exam

December 3, 2004

Your name and MIT ID:

J. D. 00000000

(Optional) The grade you would give to yourself + a brief justification.

A... why not?

Problem 1

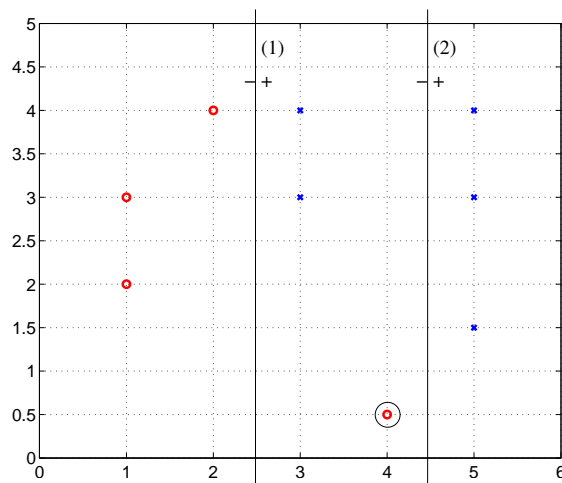


Figure 1: Labeled training points for problem 1.

Consider the labeled training points in Figure 1, where 'x' and 'o' denote positive and negative labels, respectively. We wish to apply AdaBoost with decision stumps to solve the classification problem. In each boosting iteration, we select the stump that minimizes the weighted training error, breaking ties arbitrarily.

1. **(3 points)** In figure 1, draw the decision boundary corresponding to the first decision stump that the boosting algorithm would choose. Label this boundary (1), and also indicate +/- side of the decision boundary.
2. **(2 points)** In the same figure 1 also circle the point(s) that have the highest weight after the first boosting iteration.
3. **(2 points)** What is the weighted error of the first decision stump after the first boosting iteration, i.e., after the points have been reweighted? 0.5
4. **(3 points)** Draw the decision boundary corresponding to the second decision stump, again in Figure 1, and label it with (2), also indicating the +/- side of the boundary.
5. **(3 points)** Would some of the points be misclassified by the combined classifier after the two boosting iterations? Provide a brief justification. (the points will be awarded for the justification, not whether your y/n answer is correct)

Yes. For example, the circled point in the figure is misclassified by the first decision stump and could be classified correctly in the combination only if the weight/votes of the second stump is higher than the first. If it were higher, however, then the points misclassified by the second stump would be misclassified in the combination.

Problem 2

1. **(2 points)** Consider a linear SVM trained with n labeled points in \mathcal{R}^2 without slack penalties and resulting in $k = 2$ support vectors ($k < n$). By adding one additional labeled training point and retraining the SVM classifier, what is the maximum number of support vectors in the resulting solution?

- () k
() $k + 1$
() $k + 2$
(X) $n + 1$

2. We train two SVM classifiers to separate points in \mathcal{R}^2 . The classifiers differ only in terms of the kernel function. Classifier 1 uses the linear kernel $K_1(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}'$, and classifier 2 uses $K_2(\mathbf{x}, \mathbf{x}') = p(\mathbf{x})p(\mathbf{x}')$, where $p(\mathbf{x})$ is a 3-component Gaussian mixture density, estimated on the basis of related other problems.

- (a) **(3 points)** What is the VC-dimension of the second SVM classifier that uses kernel $K_2(\mathbf{x}, \mathbf{x}')$?

2

The feature space is 1-dimensional; each point $\mathbf{x} \in \mathcal{R}^2$ is mapped to a non-negative number $p(\mathbf{x})$.

- (b) **(T/F – 2 points)** The second SVM classifier can only separate points that are likely according to $p(\mathbf{x})$ from those that have low probability under $p(\mathbf{x})$.

T

- (c) **(4 points)** If both SVM classifiers achieve zero training error on n labeled points, which classifier would have a better generalization guarantee? Provide a brief justification.

The first classifier has VC-dimension 3 while the second one has VC-dimension 2. The complexity penalty for the first one is therefore higher. When the number of training errors is the same for the two classifiers, the bound on the expected error is smaller for the second classifier.

Problem 3

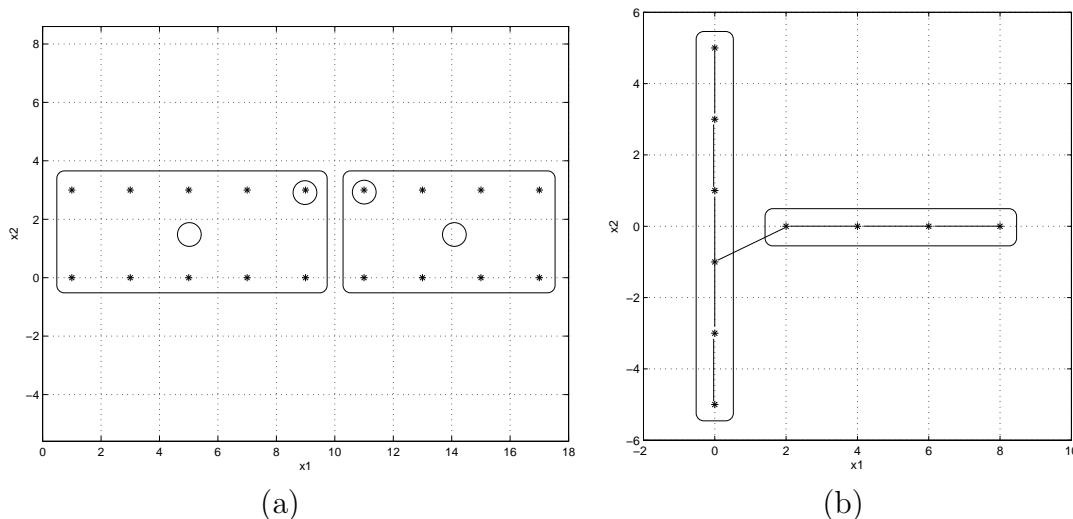


Figure 2: Data sets for clustering. Points are located at integer coordinates.

1. **(4 points)** First consider the data plotted in Figure 2a, which consist of two rows of equally spaced points. If k -means clustering ($k = 2$) is initialised with the two points whose coordinates are $(9, 3)$ and $(11, 3)$, indicate the final clusters obtained (after the algorithm converges) on Figure 2a.
2. **(4 points)** Now consider the data in Figure 2b. We will use spectral clustering to divide these points into two clusters. Our version of spectral clustering uses a neighbourhood graph obtained by connecting each point to its two nearest neighbors (breaking ties randomly), and by weighting the resulting edges between points \mathbf{x}_i and \mathbf{x}_j by $W_{ij} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|)$. Indicate on Figure 2b the clusters that we will obtain from spectral clustering. Provide a brief justification.

The random walk induced by the weights can switch between the clusters in the figure in only two places, $(0, -1)$ and $(2, 0)$. Since the weights decay with distance, the weights corresponding to transitions within clusters are higher than those going across in both places. The random walk would therefore tend to remain within the clusters indicated in the figure.

3. **(4 points)** Can the solution obtained in the previous part for the data in Figure 2b also be obtained by k -means clustering ($k = 2$)? Justify your answer.

No. In the k -means algorithm points are assigned to the closest mean (cluster centroid). The centroids of the left and right clusters in the figure are $(0, 0)$ and $(5, 0)$, respectively. Point $(2, 0)$, for example, is closer to the left cluster centroid $(0, 0)$ and wouldn't be assigned to the right cluster. The two clusters in the figure therefore cannot be fixed points of the k -means algorithm.

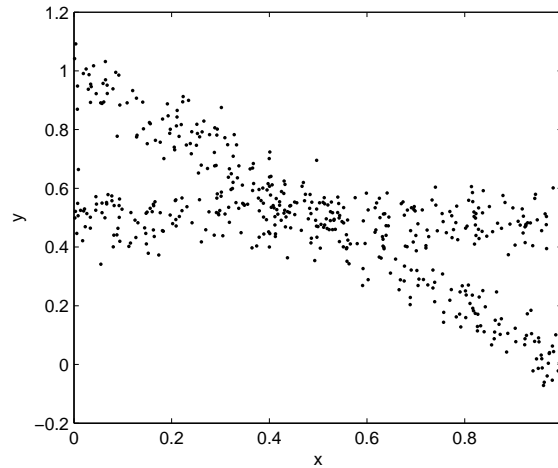


Figure 3: Training sample from a mixture of two linear models

Problem 4

The data in Figure 3 comes from a mixture of two linear regression models with Gaussian noise:

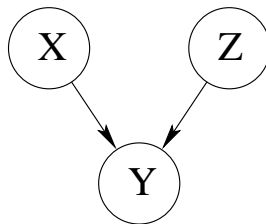
$$P(y|x; \theta) = p_1 \mathcal{N}(y; w_{10} + w_{11}x, \sigma_1^2) + p_2 \mathcal{N}(y; w_{20} + w_{21}x, \sigma_2^2)$$

where $p_1 + p_2 = 1$ and $\theta = (p_1, p_2, w_{10}, w_{11}, w_{20}, w_{21}, \sigma_1, \sigma_2)$. We hope to estimate θ from such data via the EM algorithm.

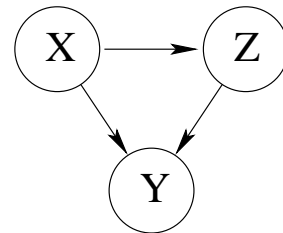
To this end, let $z \in \{1, 2\}$ be the mixture index, variable indicating which of the regression models is used to generate y given x .

1. **(6 points)** Connect the random variables X , Y , and Z with directed edges so that the graphical model on the left represents the mixture of linear regression models described above, and the one on the right represents a mixture-of-experts model. For both models, Y denotes the output variable, X the input, and Z is the choice of the linear regression model or expert.

mixture of linear regressions



mixture of experts

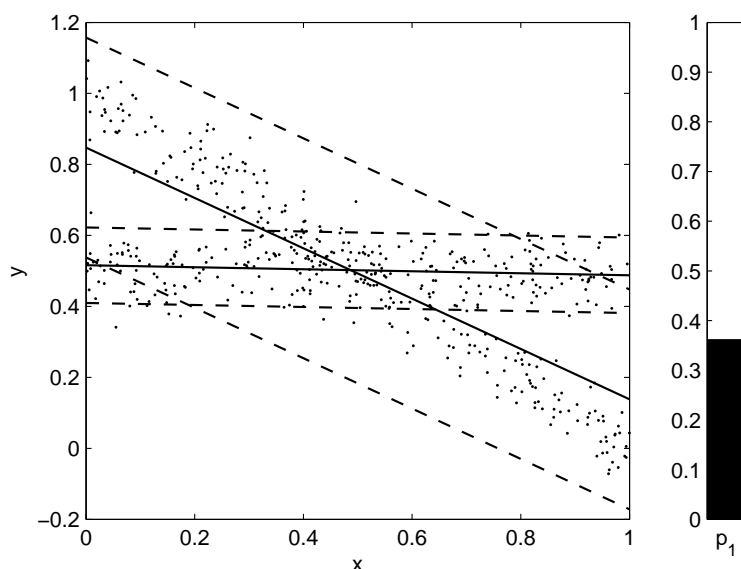


We use a single plot to represent the model parameters (see the figure below). Each linear regression model appears as a solid line ($y = w_{i0} + w_{i1}x$) in between two parallel dotted lines at vertical distance $2\sigma_i$ to the solid line. Thus each regression model “covers” the data that falls between the dotted lines. When $w_{10} = w_{20}$ and $w_{11} = w_{21}$ you would only see a single solid line in the figure; you may still see two different sets of dotted lines corresponding to different values of σ_1 and σ_2 . The solid bar to the right represents p_1 (and $p_2 = 1 - p_1$).

For example, if

$$\begin{aligned}\theta &= (p_1, p_2, w_{10}, w_{11}, w_{20}, w_{21}, \sigma_1, \sigma_2) \\ &= (0.35, 0.65, 0.5, 0, 0.85, -0.7, 0.05, 0.15)\end{aligned}$$

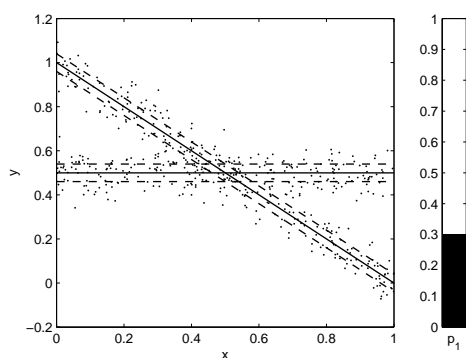
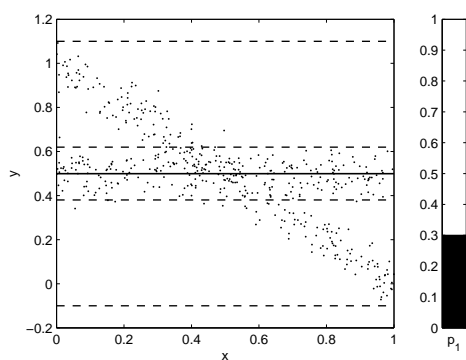
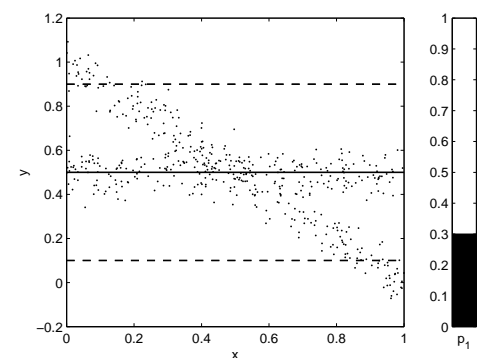
the plot is



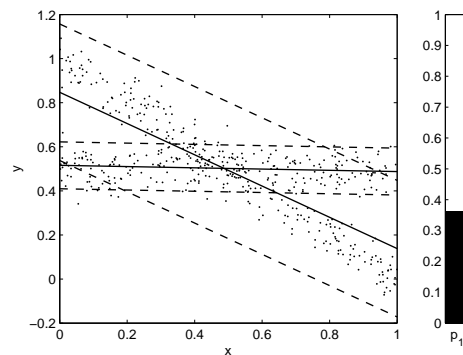
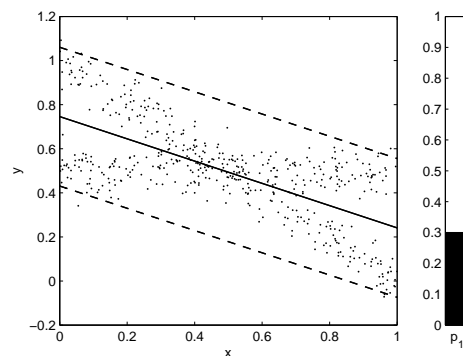
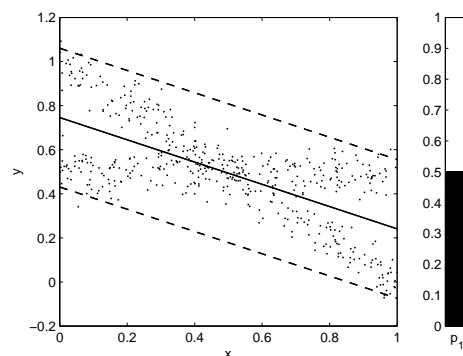
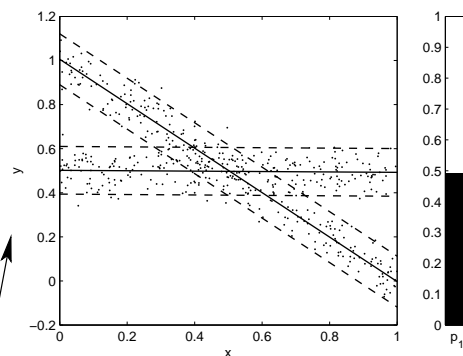
2. **(6 points)** We are now ready to estimate the parameters θ via EM. There are, however, many ways to initialize the parameters for the algorithm.

On the next page you are asked to connect 3 different initializations (left column) with the parameters that would result after one EM iteration (right column). Different initializations may lead to the same set of parameters. Your answer should consist of 3 arrows, one from each initialization.

Initialization



Next iteration



Problem 5

Assume that the following sequences are very long and the pattern highlighted with spaces is repeated:

Sequence 1: 1 0 0 1 0 0 1 0 0 1 0 0 ... 1 0 0

Sequence 2: 1 1 0 0 1 0 0 1 0 0 ... 1 0 0

- (4 points)** If we model each sequence with a different first-order HMM, what is the number of hidden states that a reasonable model selection method would report?

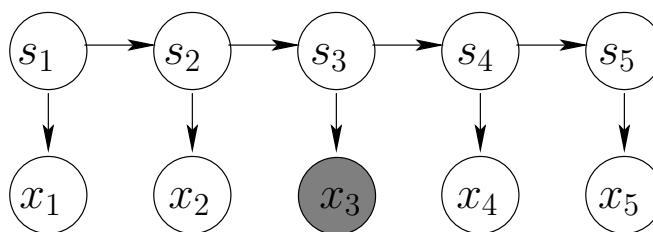
HMM for Sequence 1 HMM for Sequence 2

No. of hidden states

3

4

- (2 points)** The following Bayesian network depicts a sequence of 5 observations from an HMM, where s_1, s_2, s_3, s_4, s_5 is the hidden state sequence.



Are x_1 and x_5 independent given x_3 ? Briefly justify your answer.

They are not independent. The moralized ancestral graph corresponding to x_1 , x_3 , and x_5 is the same graph with arrows replaced with undirected edges. x_1 and x_5 are not separated given x_3 , and thus not independent.

- (3 points)** Does the order of Markov dependencies in the observed sequence always determine the number of hidden states of the HMM that generated the sequence? Provide a brief justification.

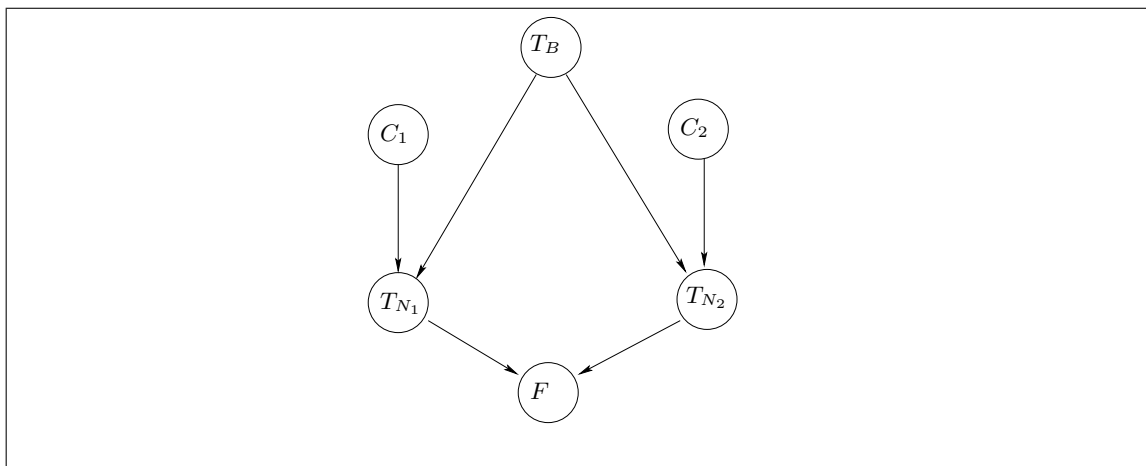
No. The answer to the previous question implies that observations corresponding to (typical) HMMs have no Markov properties (of any order). This holds, for example, when there are only two possible hidden states. Thus Markov properties of the observation sequence cannot in general determine the number of hidden states.

Problem 6

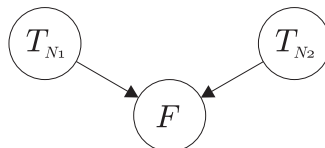
We wish to develop a graphical model for the following transportation problem. A transport company is trying to choose between two alternative routes for commuting between Boston and New York. In an experiment, two identical busses leave Boston at the same but otherwise random time, T_B . The busses take different routes, arriving at their (common) destination at times T_{N1} and T_{N2} .

Transit time for each route depends on the congestion along the route, and the two congestions are unrelated. Let us represent the random delays introduced along the routes by variables C_1 and C_2 . Finally, let F represent the identity of the bus which reaches New York first. We view F as a random variable that takes values 1 or 2.

1. **(6 points)** Complete the following directed graph (Bayesian network) with edges so that it captures the relationships between the variables in this transportation problem.



2. **(3 points)** Consider the following directed graph as a possible representation of the independences between the variables T_{N1} , T_{N2} , and F only:



Which of the following factorizations of the joint are consistent with the graph?

$$P(T_{N1})P(T_{N2})P(F|T_{N1}, T_{N2})$$

☐

$$P(T_{N1})P(T_{N2})P(F|T_{N1})$$

☐

$$P(T_{N1})P(T_{N2})P(F)$$

☐

6.867 Machine learning

Final exam (Fall 2003)

December 14, 2003

Problem 1: your information

1.1. Your name and MIT ID:

1.2. The grade you would give to yourself + brief justification (if you feel that there's no question your grade should be an A, then just say A):

A ... why not?

Problem 2

- 2.1. (3 points)** Let \mathcal{F} be a set of classifiers whose VC-dimension is 5. Suppose we have four training examples and labels, $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_4, y_4)\}$, and select a classifier \hat{f} from \mathcal{F} by minimizing classification error on the training set. In the absence of any other information about the set of classifiers \mathcal{F} , can we say that the prediction $\hat{f}(\mathbf{x}_5)$ for a new example \mathbf{x}_5 has any relation to the training set? Briefly justify your answer.

Since VC-dimension is 5, \mathcal{F} can shatter (some) five points. These points could be x_1, \dots, x_5 . Thus we can find $f_1 \in \mathcal{F}$ consistent with the four training examples and $f_1(\mathbf{x}_5) = 1$, as well as another classifier $f_{-1} \in \mathcal{F}$ also consistent with the training examples for which $f_{-1}(\mathbf{x}_5) = -1$. The training set therefore doesn't constrain the prediction at \mathbf{x}_5 .

- 2.2. (T/F – 2 points)** Consider a set of classifiers that includes all linear classifiers that use different choices of strict subsets of the components of the input vectors $\mathbf{x} \in \mathcal{R}^d$. Claim: the VC-dimension of this combined set cannot be more than $d + 1$.

T

A linear classifier based on a subset of features can be represented as a linear classifier based on all the features but we have simply set some of the parameters to zero. The set of classifiers here is therefore simply linear classifiers in \mathcal{R}^d .

- 2.3. (T/F – 2 points)** Structural risk minimization is based on comparing upper bounds on the generalization error, where the bounds hold with probability $1 - \delta$ over the choice of the training set. Claim: the value of the confidence parameter δ cannot affect model selection decisions.

F

The δ parameter changes the complexity penalty in a manner that depends on the VC-dimension and the number of training examples. It can therefore affect the model selection results.

- 2.4. (6 points)** Suppose we use class-conditional Gaussians to solve a binary classification task. The covariance matrices of the two Gaussians are constrained to be $\sigma^2 I$, where the value of σ^2 is fixed and I is the identity matrix. The only adjustable parameters are therefore the means of the class conditional Gaussians, and the prior class frequencies. We use the maximum likelihood criterion to train the model. Check all that apply.

- () For any three distinct training points and sufficiently small σ^2 , the classifier would have zero classification error on the training set
- () For any three training points and sufficiently large σ^2 , the classifier would always make one classification error on the training set
- () The classification error of this classifier on the training set is always at least that of a linear SVM, whether the points are linearly separable or not

None is the correct answer. The classifier is trained as a generative model, thus it places each Gaussian at the sample mean of the points in the class. The prior class frequencies will reflect the number of points in each class.

- *Since points in one class can be further away from their mean than from points in the other class, the first option cannot be correct.*
- *For the second option we can assume that the variance σ^2 is much larger than the distances between the points. In this case the Gaussians from each class assign roughly the same probability to any training point. The prior class frequencies favor the dominant class, therefore resulting in one error. This does not hold, however, when all the training points come from the same class. Thus you got 2pts for either answer to the second.*
- *The last option is incorrect since SVM does not minimize the number of misclassified points when the points are not linearly separable; the accuracy could go either way, albeit typically in favor of SVMs.*

Problem 3

- 3.1. (T/F – 2 points)** In the AdaBoost algorithm, the weights on all the misclassified points will go up by the same multiplicative factor.

T

You can verify this by inspecting the weight update. Note that the adaboost algorithm was formulated for weak classifiers that predict ± 1 labels. The weights of all misclassified points will be multiplied by $\exp(-\hat{\alpha}_i h(\mathbf{x}_i; \hat{\theta}_k)) = \exp(\hat{\alpha})$ before normalization.

- 3.2. (3 points)** Provide a brief rationale for the following observation about AdaBoost. The weighted error of the k^{th} weak classifier (measured relative to the weights at the beginning of the k^{th} iteration) tends to increase as a function of the iteration k .

The weighting on the training examples focuses on examples that are hard to classify correctly (few weak classifiers have classified such examples correctly). After a few iterations most of the weight will be on these hard examples and the weighted error on the next weak classifier will be closer to chance.

Consider a text classification problem, where documents are represented by binary (0/1) feature vectors $\phi = [\phi_1, \dots, \phi_m]^T$; here ϕ_i indicates whether word i appears in the document. We define a set of weak classifiers, $h(\phi; \theta) = y\phi_i$, parameterized by $\theta = \{i, y\}$ (the choice of the component, $i \in \{1, \dots, m\}$, and the class label, $y \in \{-1, 1\}$, that the component should be associated with). There are exactly $2m$ possible weak learners of this type.

We use this boosting algorithm for feature selection. The idea is to simply run the boosting algorithm and select the features or components in the order in which they were identified by the weak learners. We assume that the boosting algorithm finds the best available weak classifier at each iteration.

- 3.3. (T/F – 2 points)** The boosting algorithm described here can select the exact same weak classifier more than once.

T

The boosting algorithm optimizes each new α by assuming that all the previous votes remain fixed. It therefore does not optimize these coefficients jointly. The only way to correct the votes assigned to a weak learner later on is to introduce the same weak learner again. Since we only have a discrete set of possible weak learners here, it also makes sense to talk about selecting the exact same weak learner again.

- 3.4. (4 points)** Is the ranking of features generated by the boosting algorithm likely to be more useful for a linear classifier than the ranking from simple mutual information calculations (estimates $\hat{I}(y; \phi_i)$). Briefly justify your answer.

The boosting algorithm generates a linear combination of weak classifiers (here features). The algorithm therefore evaluates each new weak classifier (feature) relative to a linear prediction based on those already included. The mutual information criterion considers each feature individually and is therefore unable to recognize how multiple features might interact to benefit linear prediction.

Problem 4

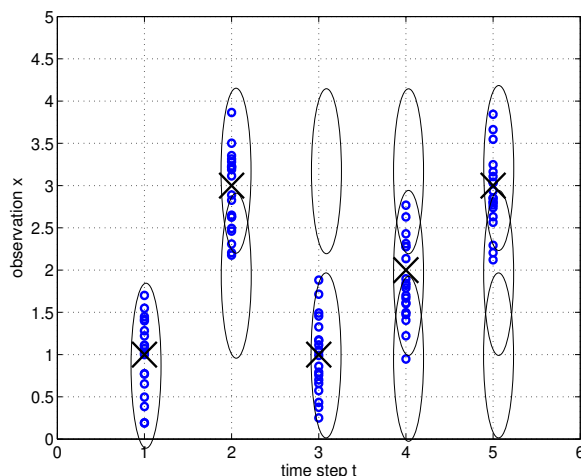


Figure 1a)

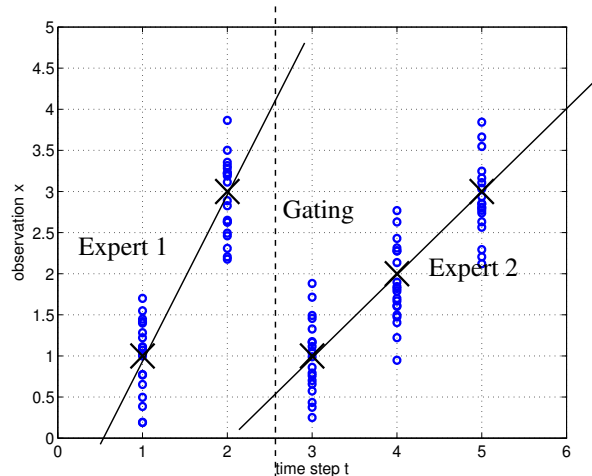
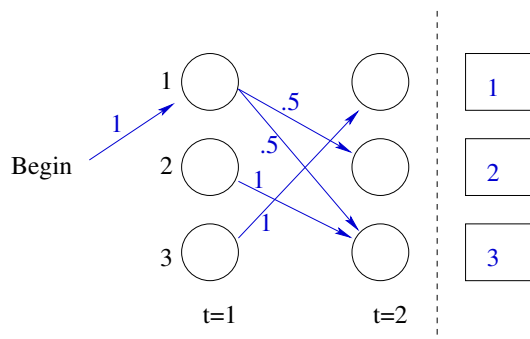


Figure 1b)

Figure 1: Time dependent observations. The data points in the figure are generated as sets of five consecutive time dependent observations, x_1, \dots, x_5 . The clusters come from repeatedly generating five consecutive samples. Each visible cluster consists of 20 points, and has approximately the same variance. The mean of each cluster is shown with a large X.

Consider the data in Figure 1 (see the caption for details). We begin by modeling this data with a three state HMM, where each state has a Gaussian output distribution with some mean and variance (means and variances can be set independently for each state).

- 4.1. (4 points)** Draw the state transition diagram and the initial state distribution for a three state HMM that models the data in Figure 1 in the maximum likelihood sense. Indicate the possible transitions and their probabilities in the figure below (whether or not the state is reachable after the first two steps). In other words, your drawing should characterize the 1st order homogeneous Markov chain governing the evolution of the states. Also indicate the means of the corresponding Gaussian output distributions (please use the boxes).



4.2. (4 points) In Figure 1a draw as ovals the clusters of outputs that would form if we repeatedly generated samples from your HMM over time steps $t = 1, \dots, 5$. The height of the ovals should reflect the variance of the clusters.

4.3. (4 points) Suppose at time $t = 2$ we observe $x_2 = 1.5$ but don't see the observations for other time points. What is the most likely state at $t = 2$ according to the marginal posterior probability $\gamma_2(s)$ defined as $P(s_2 = s | x_2 = 1.5)$.

2

We have only two possible paths for the first three states, 1,2,3, or 1,3,1. The marginal posterior probability comes from averaging the state occupancies across these possible paths, weighted by the corresponding probabilities. Given the observation at $t = 2$ (mean of the output distribution from state 2), the first path is more likely.

4.4. (2 points) What would be the most likely state at $t = 2$ if we also saw $x_3 = 0$ at $t = 3$? In this case $\gamma_2(s) = P(s_2 = s | x_2 = 1.5, x_3 = 0)$.

3

The new observation at $t = 3$ is very unlikely to have come from state 3, thus we switch to state sequence 1,3,1.

4.5. (4 points) We can also try to model the data with conditional mixtures (mixtures of experts), where the conditioning is based on the time step. Suppose we only use two experts which are linear regression models with additive Gaussian noise, i.e.,

$$P(x|t, \theta_i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp \left\{ -\frac{1}{2\sigma_i^2} (x - \theta_{i0} - \theta_{i1}t)^2 \right\}$$

for $i = 1, 2$. The gating network is a logistic regression model from t to binary selection of the experts. Assuming your estimation of the conditional mixture model is successfully in the maximum likelihood sense, draw the resulting mean predictions of the two linear regression models as a function of time t in Figure 1b). Also, with a vertical line, indicate where the gating network would change its preference from one expert to the other.

4.6. (T/F – 2 points) Claim: by repeatedly sampling from your conditional mixture model at successive time points $t = 1, 2, 3, 4, 5$, the resulting samples would resemble the data in Figure 1

T

See the figure.

- 4.7. (4 points)** Having two competing models for the same data, the HMM and the mixture of experts model, we'd like to select the better one. We think that any reasonable model selection criterion would be able to select the better model in this case. Which model would we choose? Provide a brief justification.

The mixture of experts model assigns a higher probability to the available data since the HMM puts some of the probability mass where there are no points. The HMM also has more parameters so any reasonable model selection criterion should select the mixture of experts model.

Problem 5

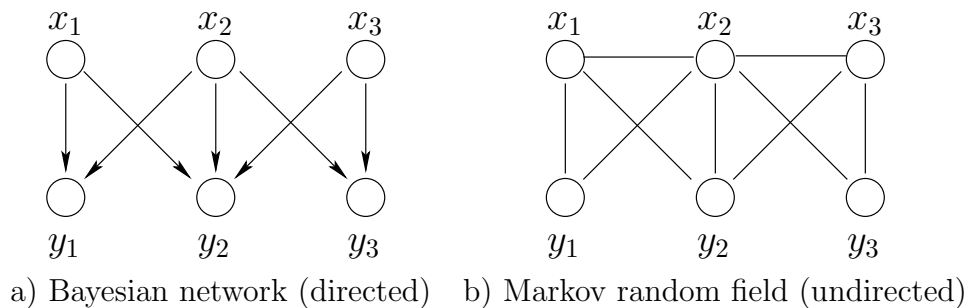


Figure 2: Graphical models

- 5.1. (2 points)** List two different types of independence properties satisfied by the Bayesian network model in Figure 2a.

1) x_1 and x_2 are marginally independent.
 2) y_1 and y_2 are conditionally independent given x_1 and x_2 .
 Lots of other possibilities.

- 5.2. (2 points)** Write the factorization of the joint distribution implied by the directed graph in Figure 2a.

$P(x_1)P(x_2)P(x_3)P(y_1|x_1, x_2)P(y_2|x_1, x_2, x_3)P(y_3|x_2, x_3).$

- 5.3. (2 points)** Provide an alternative factorization of the joint distribution, different from the previous one. Your factorization should be consistent with all the properties of the directed graph in Figure 2a. Consistency here means: whatever is implied by the graph should hold for the associated distribution.

Any factorization that incorporates all the independencies from the graph and a few more would be possible. For example, $P(x_1)P(x_2)P(x_3)P(y_1)P(y_2)P(y_3)$. In this case all the variables are independent, so any independence statement that we can derive from the graph clearly holds for this distribution as well.

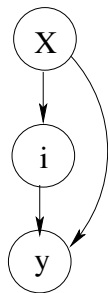
- 5.4. (4 points) Provide an independence statement that holds for the undirected model in Figure 2b but does NOT hold for the Bayesian network. Which edge(s) should we add to the undirected model so that it would be consistent with (wouldn't imply anything that is not true for) the Bayesian network?

x_1 is independent of x_3 given x_2 and y_2 . In the bayesian network any knowledge of y_2 would make x_1 and x_3 dependent. Adding an edge between x_1 and x_3 would suffice (cf. moralization).

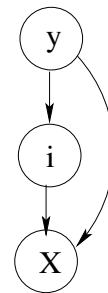
- 5.5. (2 points) Is your resulting undirected graph triangulated (Y/N)?

Y

- 5.6. (4 points) Provide two directed graphs representing 1) a mixture of two experts model for classification, and 2) a mixture of Gaussians classifiers with two mixture components per class. Please use the following notation: \mathbf{x} for the input observation, y for the class, and i for any selection of components.



A mixture of experts classifier, where $i = 1, 2$ selects the expert.



A mixture of Gaussians model, where $i = 1, 2$ selects the Gaussian component within each class.

6.867 Machine learning

Final exam

December 5, 2002

(2 points) Your name and MIT ID:

J Doe, #000

(4 points) The grade you would give to yourself + a brief justification:

A or perhaps A- if there are any typos or other errors in the solutions...

Problem 1

We wish to estimate a mixture of two experts model for the data displayed in Figure 1. The experts we can use here are linear regression models of the form

$$p(y|x, \mathbf{w}) = N(y; w_1x + w_0, \sigma^2)$$

where $N(y; \mu, \sigma^2)$ denotes a Gaussian distribution over y with mean μ and variance σ^2 . Each expert i can choose its parameters $\mathbf{w}_i = [w_{i0}, w_{i1}]^T$ and σ_i^2 independently from other experts. Note that the first subindex i in w_{ij} refers to the expert.

The gating network in the case of two experts is given by a logistic regression model

$$P(\text{expert} = 1|x, \mathbf{v}) = g(v_1x + v_0)$$

where $g(z) = (1 + \exp(-z))^{-1}$ and $\mathbf{v} = [v_0, v_1]^T$.

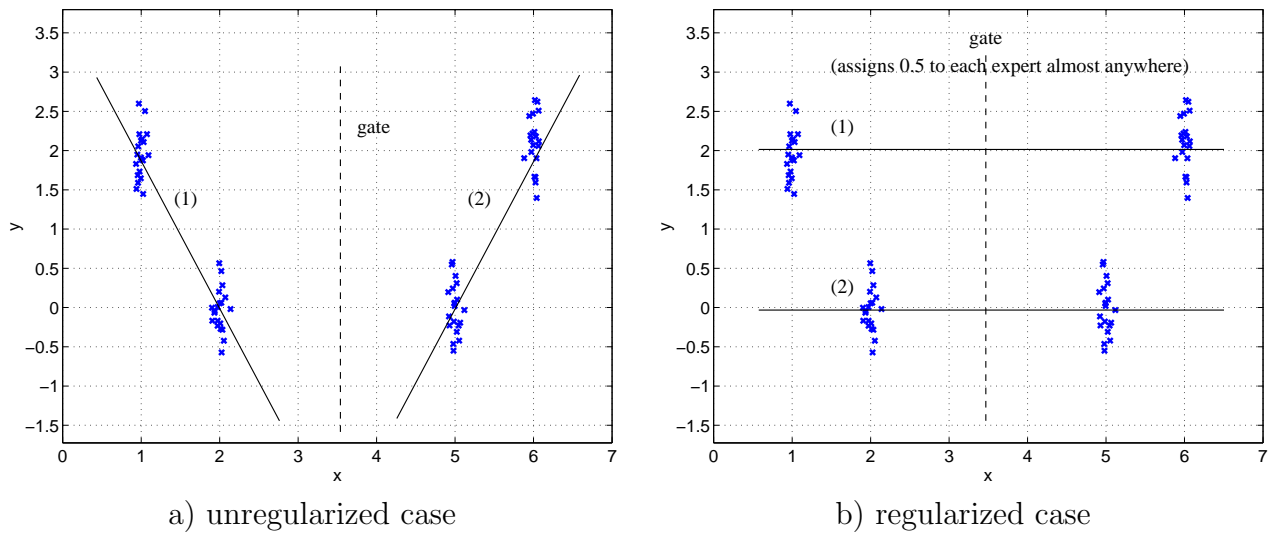


Figure 1: Data for mixtures of experts

1. **(4 points)** Suppose we estimate a mixture of two experts model based on the data in Figure 1. You can assume that the estimation is successful in the sense that we will find a setting of the parameters that maximizes the log-likelihood of the data. Please indicate (approximately) in Figure 1a) the mean predictions from the two experts as well as the decision boundary for the gating network. Label the mean predictions – functions of x – with “(1)” and “(2)” corresponding to the two experts, and the decision boundary with “gate”.
2. **(4 points)** We now switch to a regularized maximum likelihood objective by incorporating the following regularization penalty

$$-\frac{c}{2}(w_{11}^2 + w_{21}^2)$$

into the log-likelihood objective. Note that the penalty includes only one parameter from each of the experts. By increasing c , we impose a stronger penalty. Similarly to the previous question, please indicate in Figure 1b) the optimal regularized solution for the mixture of two experts model when the regularization parameter c is set to a very large value.

3. **(3 points)** Are the variances in the predictive Gaussian distributions of the experts
 - () larger,
 - () smaller,
 - (x) about the same
 after the regularization?

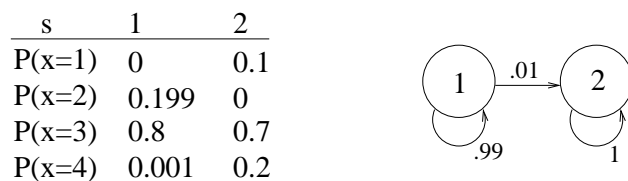


Figure 2: A two-state HMM for Problem 2

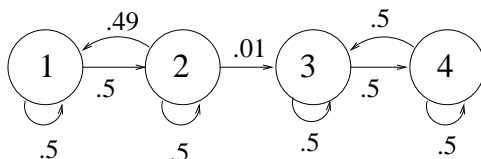


Figure 3: An alternative, four-state HMM for Problem 2

Problem 2

Figure 2 shows a two-state HMM. The transition probabilities of the Markov chain are given in the transition diagram. The output distribution corresponding to each state is defined over $\{1, 2, 3, 4\}$ and is given in the table next to the diagram. The HMM is equally likely to start from either of the two states.

- (3 points)** Give an example of an output sequence of length 2 which can not be generated by the HMM in Figure 2.
- (2 points)** We generated a sequence of $6,867^{2002}$ observations from the HMM, and found that the last observation in the sequence was 3. What is the most likely hidden state corresponding to that last observation?
- (2 points)** Consider an output sequence 3 3. What is the most likely sequence of hidden states corresponding to these observations?
- (2 points)** Now, consider an output sequence 3 3 4. What are *the first two states* of the most likely hidden state sequence?

1,2

2

1,1

2,2

5. **(4 points)** We can try to increase the modeling capacity of the HMM a bit by breaking each state into two states. Following this idea, we created the diagram in Figure 3. Can we set the initial state distribution and the output distributions so that this 4-state model, with the transition probabilities indicated in the diagram, would be equivalent to the original 2-state model? If yes, how? If no, why not?

No we cannot. First note that we have to associate the first two states in the 4-state model with state 1 of the 2-state model. The probability of leaving the first two states in the 4-state model, however, depends on time (whether the chain happens to be in state 1 or 2). In contrast, in the 2-state model the probability of transitioning to 2 is always 0.01.

6. **(T/F – 2 points)** The Markov chain in Figure 3 is ergodic

F

Problem 3

Figure 4 shows a graphical model over four binary valued variables, x_1, \dots, x_4 . We do not know the parameters of the probability distribution associated with the graph.

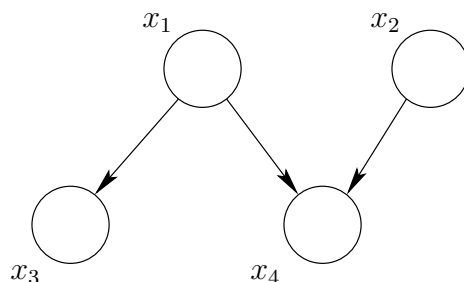


Figure 4: A graphical model

1. **(2 points)** Would it typically help to know the value of x_3 so as to gain more information about x_2 ? (please answer **yes** or **no**) no
2. **(2 points)** Assume we already know the value of x_4 . Would it help in this case to know the value of x_3 to gain more information about x_2 ? (please answer **yes** or **no**) yes
3. **(3 points)** List three different conditional independence statements between the four variables that can be inferred from the graph. You can include marginal independence by saying “given nothing”.
 - (a) (x_1) is independent of (x_2) given (nothing)
 - (b) (x_3) is independent of (x_2) given (nothing)
 - (c) (x_3) is independent of (x_4) given (x_1)
4. **(2 points)** The following table gives a possible partial specification of the conditional probability $P(x_4|x_1, x_2)$ associated with the graph. Fill in the missing values so that we could omit the arrow $x_1 \rightarrow x_4$ in the graph and the graph would still adequately represent the probability distribution.

$P(x_4 = 1 x_1 = 0, x_2 = 0)$	0.8
$P(x_4 = 1 x_1 = 0, x_2 = 1)$	0.4
$P(x_4 = 1 x_1 = 1, x_2 = 0)$	0.8
$P(x_4 = 1 x_1 = 1, x_2 = 1)$	0.4

5. **(4 points)** Let's again focus on the original graph in Figure 4. Since we don't know the underlying probability distribution, we need to estimate it from observed data. Unfortunately, the dataset we have is incomplete and contains only observations for x_2 and x_3 . In other words, the dataset is $D = \{(x_2^t, x_3^t), t = 1, \dots, n\}$. In the joint distribution

$$P(x_1, x_2, x_3, x_4) = P(x_1)P(x_2)P(x_3|x_1)P(x_4|x_1, x_2)$$

we have a number of components (smaller probability tables) that need to be estimated. Please indicate which components we can hope to estimate (adjust) on the basis of the available data?

(x) $P(x_1)$

(x) $P(x_2)$

(x) $P(x_3|x_1)$

() $P(x_4|x_1, x_2)$

6. **(4 points)** If we use the EM algorithm to carry out the estimation task, what posterior probabilities do we have to evaluate in the E-step? Please provide the necessary posterior probabilities in the form $P(\dots|x_2^t, x_3^t)$.

We only need to evaluate $P(x_1|x_3^t)$.

Since x_4 is never observed, x_1 and x_2 are always independent. We thus have to estimate only two independent parts $x_1 \rightarrow x_3$ and x_2 , where x_1 is unobserved. In the EM-algorithm we need to fill-in the missing values for x_1 , and thus evaluate $P(x_1|x_3^t)$

Problem 4

We try to select here between two models. Both models are logistic regression models but differ in terms of the type of features that are used in making the predictions. More specifically, the models have the common squashed additive form

$$P(y = 1|x, \mathbf{w}) = g(w_0 + w_1\phi_1(x) + \dots + \phi_m(x))$$

where the input is a real number $x \in \mathcal{R}$. The models differ in terms of the number and the type of basis functions used:

model 1 : $m = 1, \phi_1(x) = x$

model 2 : $m = 2, \phi_1(x) = x, \phi_2(x) = \sin(x)$

1. **(4 points)** Suppose we have n training examples (x^t, y^t) , $t = 1, \dots, n$, and we evaluate structural risk minimization scores (bounds on the generalization error) for the two classifiers. Which of the following statements are valid in general for our two models:

☐ Score for model 1 \geq Score for model 2

☐ Score for model 1 \leq Score for model 2

☐ Score for model 1 = Score for model 2

☒ Each of the above three cases may be correct depending on the data

☐ None of the above

2. **(4 points)** We will now switch to the Bayesian information criterion (BIC) for selecting among the two models. Let $L_1(n)$ be the log-probability of the labels that model 1 assigns to n training labels, where the probabilities are evaluated at the maximum likelihood setting of the parameters. Let $L_2(n)$ be the corresponding log-probability for model 2. We imagine here that $L_1(n)$ and $L_2(n)$ are evaluated on the basis of the first n training examples from a much larger set.

Now, in our empirical studies, we found that these log-probabilities are related in a simple way:

$$L_2(n) - L_1(n) \approx 0.01 \cdot n$$

How will we end up selecting between the two models as a function of the number of training examples? Please choose one of the following cases.

- ☐ Always select 1
 - ☐ Always select 2
 - ☒ First select 1, then 2 for larger n
 - ☐ First select 2, then 1 for larger n
3. **(4 points)** Provide a brief justification for your answer to the previous question.

For large n we would select model 2 since it has a consistent (albeit small) advantage. Initially, however, we would choose model 1 due to smaller complexity penalty.

To see this a bit more precisely, let's recall the form of the BIC score: for model 1 it is defined as $BIC_1 = L_1(n) - \frac{d_1}{2} \log(n)$, where d_1 is the number of parameters in model 1. The difference between the BIC scores for the two models is therefore

$$\begin{aligned} BIC_2 - BIC_1 &= \overbrace{L_2(n) - L_1(n)}^{0.01 \cdot n} - \overbrace{\frac{d_2 - d_1}{2}}^{3-2} \log(n) \\ &= 0.01 \cdot n - \frac{1}{2} \log(n) \end{aligned}$$

When n is small the complexity term dominates and $BIC_2 < BIC_1$ (the difference is negative). For large n the linear increase of the log-likelihood difference overcomes the logarithmic penalty and $BIC_2 > BIC_1$.

Problem 5

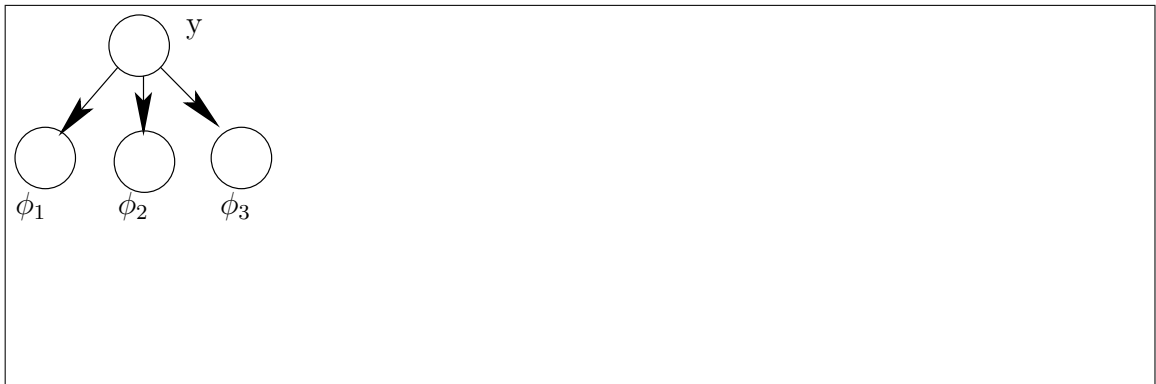
Consider a simple two-class document (text) classification problem. Each document is represented by a binary feature vector $[\phi_1^i, \dots, \phi_N^i]$, where $\phi_k^i = 1$ if keyword k is present in the document, and zero otherwise. N is the number of keywords we have chosen to include. We use a Naive Bayes model for this classification task. The joint distribution of the features and the binary labels $y \in \{0, 1\}$ is in this case given by

$$P(\phi_1, \dots, \phi_N, y) = P(y) \prod_{k=1}^N P(\phi_k | y)$$

where, for example,

$$P(\phi_k = 1 | y = 0) = \theta_{k,0}, \quad P(\phi_k = 1 | y = 1) = \theta_{k,1}$$

1. **(2 points)** In the space below, draw the graphical model corresponding to Naive Bayes generative model described above. Assume that $N = 3$ (three keywords).



2. **(4 points)** To be able to make use of training examples with possibly missing labels, we will have to resort to the EM algorithm. In the EM algorithm we need to evaluate the posterior probability of the label y given the document. We will use a message passing algorithm (belief propagation) to get this posterior probability. The problem here is that we relied on a rather careless friend to evaluate whether a document contains any of the keywords. In other words, we do not fully trust the “observed” values of the features. Let $\hat{\phi}_k$ be the “observed” value for the k^{th} feature in a given document. The evidence we now have about the actual value of ϕ_k is given by $P(\hat{\phi}_k | \phi_k)$, which is a table that models how we expect the friend to respond. Given that we observe $\hat{\phi}_k = 1$, what is the message that ϕ_k needs to send to y ?

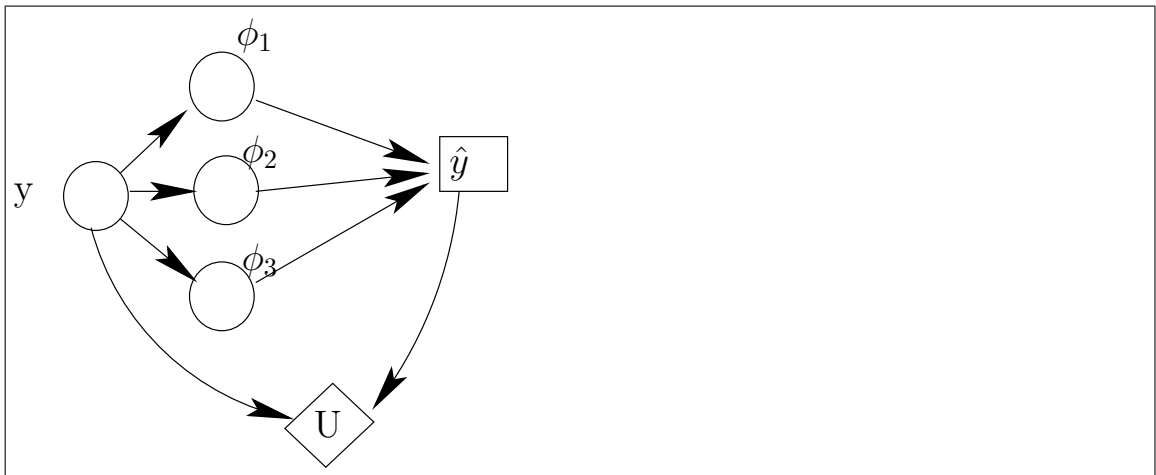
There are a couple of options here depending on how you choose to incorporate the evidence $\hat{\phi}_k = 1$. Perhaps the simplest way is to put it directly in the single node potential $\psi_k(\phi_k)$. In other words, we define $\psi_k(\phi_k) = P(\hat{\phi}_k = 1|\phi_k)$, and $\psi_{ky}(\phi_k, y) = P(\phi_k|y)$. In this notation, the message is

$$m_{k \rightarrow y}(y) = \sum_{\phi_k=0,1} \psi_k(\phi_k) \psi_{ky}(\phi_k, y)$$

3. **(3 points)** We know that the EM algorithm is in some sense monotonic. Let $\hat{\theta}_{k,y}^{(t)}$ be the estimate of the parameters $\theta_{k,y}$ in the beginning of iteration t of the EM algorithm, and $\theta^{(t)}$ be the vector of all parameter estimates in that iteration, $[\hat{\theta}_{1,0}^{(t)}, \hat{\theta}_{1,1}^{(t)}, \dots, \hat{\theta}_{N,y=1}^{(t)}]$. Which of the following quantities increases monotonically with t ?

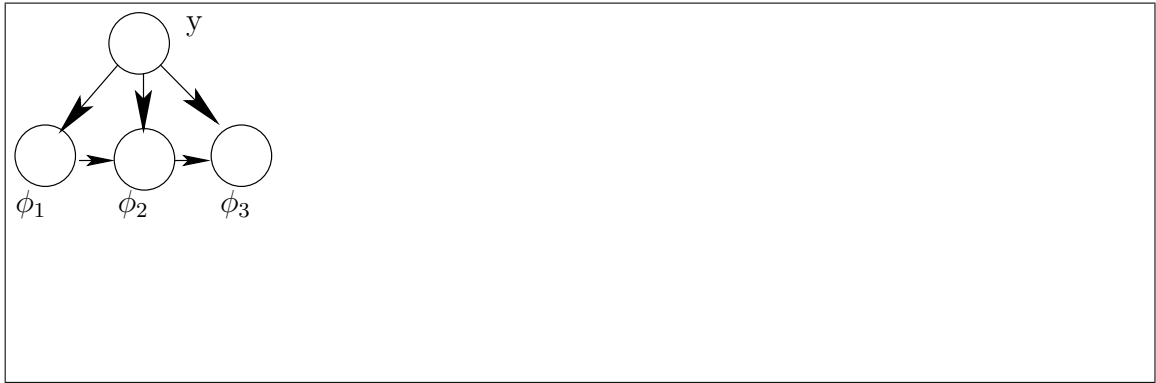
- () $P(\phi_k = 1|y, \theta_{k,y}^{(t)})$ for all k
 (x) $\prod_{i=1}^N P(\phi_1^i, \dots, \phi_N^i | \theta^{(t)})$
 () $\prod_{i=1}^N P(y_i = 1 | \phi_1^i, \dots, \phi_N^i, \theta^{(t)})$

4. **(4 points)** The class labels actually correspond to “relevant” and “irrelevant” documents. In classifying any document as relevant or irrelevant, we have to take into account that we might prefer to miss a few relevant documents if we can avoid misclassifying a large number of irrelevant documents as relevant. To express such a preference we define a utility $U(y, \hat{y})$, where y is the correct label and \hat{y} is how we classify the document. Draw an influence diagram that incorporates the Naive Bayes model, our decisions, and the utility. Mark each node in the graph according to the variables (or utility) that they represent.



5. **(2 points)** Let's modify the Naive Bayes model a bit, to account for some of the possible dependencies between the keywords. For example, suppose we order the

keywords so that it would be useful to model the dependency of ϕ_k on ϕ_{k-1} , $k = 2, \dots, N$ (the keywords may, for example, represent nested categories). We expect these dependencies to be the same for each class, but the parameters can be affected by the class label. Draw the graphical model for this – call it *Sophisticated Bayes* – model. Please assume again that $N = 3$.



6.867 Machine learning and neural networks

FALL 2001 – Final exam

December 11, 2001

(2 points) Your name and MIT ID #:

(4 points) The grade you would give to yourself + brief justification. If you feel that there's no question that your grade should be A (and you feel we agree with you) then just write "A".

Problem 1

1. (T/F – 2 points) The sequence of output symbols sampled from a hidden Markov model satisfies the first order Markov property
2. (T/F – 2 points) Increasing the number of values for the the hidden states in an HMM has much greater effect on the computational cost of forward-backward algorithm than increasing the length of the observation sequence.

3. **(T/F – 2 points)** In HMMs, if there are at least two distinct most likely hidden state sequences and the two state sequences cross in the middle (share a single state at an intermediate time point), then there are at least four most likely state sequences.
4. **(T/F – 2 points)** One advantage of Boosting is that it does not overfit.
5. **(T/F – 2 points)** Support vector machines are resistant to outliers, i.e., very noisy examples drawn from a different distribution.
6. **(T/F – 2 points)** Active learning can substantially reduce the number of training examples that we need.

T

F

F

T

Problem 2

Consider two classifiers: 1) an SVM with a quadratic (second order polynomial) kernel function and 2) an unconstrained mixture of two Gaussians model, one Gaussian per class label. These classifiers try to map examples in \mathcal{R}^2 to binary labels. We assume that the problem is separable, no slack penalties are added to the SVM classifier, and that we have sufficiently many training examples to estimate the covariance matrices of the two Gaussian components.

1. **(T/F – 2 points)** The two classifiers have the same VC-dimension.

T

2. **(4 points)** Suppose we evaluated the structural risk minimization score for the two classifiers. The score is the bound on the expected loss of the classifier, when the classifier is estimated on the basis of n training examples. Which of the two classifiers might yield the better (lower) score? Provide a brief justification.

The SVM would probably get a better score. Both classifiers have the same complexity penalty but SVM would better optimize the training error resulting in a lower (or equal) overall score.

3. **(4 points)** Suppose now that we regularize the estimation of the covariance matrices for the mixture of two Gaussians. In other words, we would estimate each class conditional covariance matrix according to

$$\hat{\Sigma}_{reg} = \frac{n}{n + n'} \hat{\Sigma} + \frac{n'}{n + n'} S \quad (1)$$

where n is the number of training examples, $\hat{\Sigma}$ is the unregularized estimate of the covariance matrix (sample covariance matrix of the examples in one class), S is our prior covariance matrix (same for both classes), and n' the equivalent sample size that we can use to balance between the prior and the data.

In computing the VC-dimension of a classifier, we can choose the set of points that we try to “shatter”. In particular, we can scale any k points by a large factor and use the resulting set of points for shattering. In light of this, would you expect our regularization to change the VC-dimension? Why or why not?

No. We can scale the points so that the sample covariance matrix becomes very large in comparison to the prior, essentially washing away any effect from the prior.

4. **(T/F – 2 points)** Regularization in the above sense would improve the structural risk minimization score for the mixture of two Gaussians.

F

Problem 3

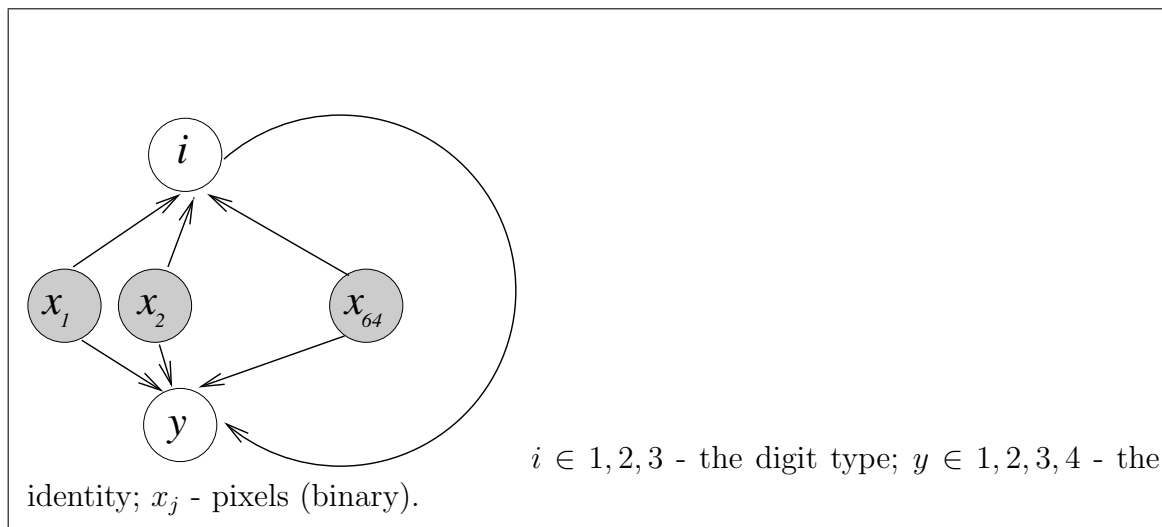
The problem here is to predict the identity of a person based on a single handwritten character. The observed characters (one of 'a', 'b', or 'c') are transformed into binary 8 by 8 pixel images. There are four different people we need to identify on the basis of such characters. To do this, we have a training set of about 200 examples, where each example consists of a binary 8x8 image and the identity of the person it belongs to. You can assume that the overall number of occurrences of each person and each character in the training set is roughly balanced.

We would like to use a mixture of experts architecture to solve this problem.

1. **(2 points)** How might the experts be useful ? Suggest what task each expert might solve.

Each expert would be responsible for classifying by a single digit; this corresponds to a mixture of 3 experts.

2. **(4 points)** Draw a graphical model that describes the mixture of experts architecture in this context. Indicate what the variables are and the number of values that they can take. Shade any nodes corresponding to variables that are always observed.



3. **(4 points)** Before implementing the mixture of experts architecture, we need to know the parametric form of the conditional probabilities in your graphical model. Provide a *reasonable* specification of the relevant conditional probabilities to the extent that you could ask your class-mate to implement the classifier.

We need to specify $P(i|x)$ and $P(y|i, x)$. $P(i|x)$ is a softmax regression model, taking as input x represented as a binary vector. $P(y|i, x)$ is also a softmax regression model, where we have a different set of weights for each value of i .

4. **(4 points)** So we implemented your method, ran the estimation algorithm once, and measured the test performance. The method was unfortunately performing at a chance level. Provide *two* possible explanations for this. (there may be multiple correct answers here)

Problem 1: there are too few training examples. We have $200/(4*3)$ which is approximately 10 training examples per expert.

Problem 2: like in mixture models, we might converge to a locally optimal solution.

5. **(3 points)** Would we get anything reasonable out of the estimation if, initially, all the experts were identical while the parameters of the gating network would be chosen randomly? By reasonable we mean training performance. Provide a brief justification.

Yes, the training examples would be assigned to experts with different probabilities because of the gating network. This would ensure that the M-step of the EM algorithm would make the experts different. Note that the posterior probability of assigning examples to different experts is based on both the gating network and how well each expert can predict the output from the input.

6. **(3 points)** Would we get anything reasonable out the estimation if now the gating network is initially set so that it assigns each training example uniformly to all experts but the experts themselves are initialized with random parameter values? Again, reasonable refers to the training error. Provide a brief justification.

Yes. Again, the posterior is based on both the gating network and the experts. In this case, the experts would make different predictions resulting in different posterior assignments of examples to experts. The gating network would be modified on the basis of these assignments.

Problem 4

Consider the following pair of observed sequences:

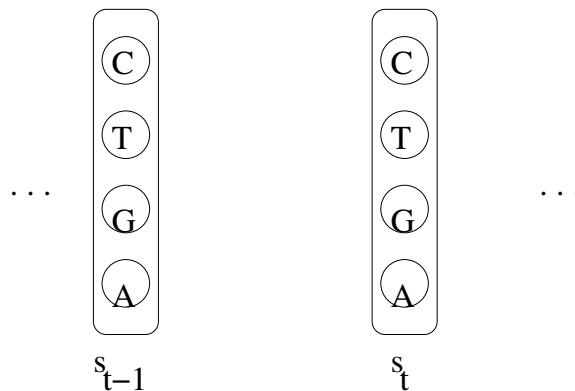
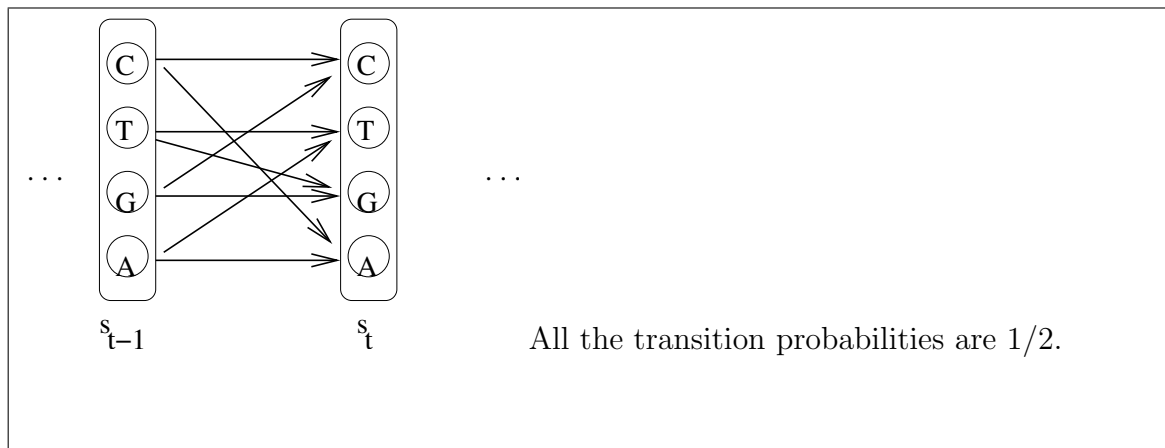
Sequence 1 (s_t): A A T T G G C C A A T T G G C C ...

Sequence 2 (x_t): 1 1 2 2 1 1 2 2 1 1 2 2 1 1 2 2 ...

Position t : 0 1 2 3 4 ...

where we assume that the pattern (highlighted with the spaces) will continue forever. Let $s_t \in \{A, G, T, C\}$, $t = 0, 1, 2, \dots$ denote the variables associated with the first sequence, and $x_t \in \{1, 2\}$, $t = 0, 1, 2, \dots$ the variables characterizing the second sequence. So, for example, given the sequences above, the observed values for these variables are $s_0 = A$, $s_1 = A$, $s_2 = C$, ..., and, similarly, $x_0 = 1$, $x_1 = 1$, $x_2 = 2$, ...

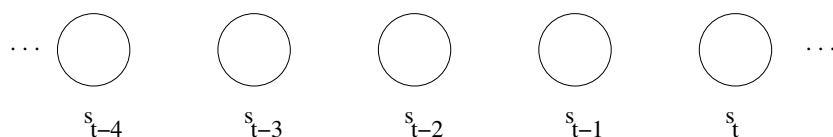
1. **(4 points)** If we use a simple first order homogeneous markov model to predict the first sequence (values for s_t only), what is the maximum likelihood solution that we would find? In the *transition diagram* below, please draw the relevant transitions and the associated probabilities (this should not require much calculation)



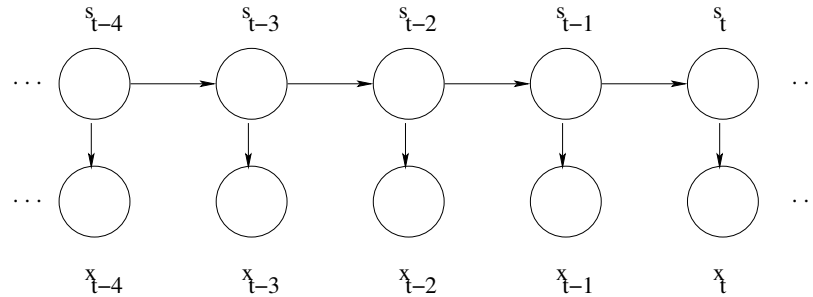
2. (T/F – 2 points) The resulting first order Markov model is *ergodic*

T

3. (4 points) To improve the Markov model a bit we would like to define a graphical model that predicts the value of s_t on the basis of the previous observed values s_{t-1}, s_{t-2}, \dots (looking as far back as needed). The model parameters/structure are assumed to remain the same if we shift the model one step. In other words, it is the same graphical model that predicts s_t on the basis of s_{t-1}, s_{t-2}, \dots as the model that predicts s_{t-1} on the basis of s_{t-2}, s_{t-3}, \dots . In the graph below, draw the *minimum number of arrows* that are needed to predict the first observed sequence perfectly (disregarding the first few symbols in the sequence). Since we slide the model along the sequence, you can draw the arrows only for s_t .



4. Now, to incorporate the second observation sequence, we will use a standard hidden Markov model:



where again $s_t \in \{A, G, T, C\}$ and $x_t \in \{1, 2\}$. We will estimate the parameters of this HMM in two different ways.

- (I) Treat the pair of observed sequences (s_t, x_t) (given above) as complete observations of the variables in the model and estimate the parameters in the maximum likelihood sense. The initial state distribution $P_0(s_0)$ is set according to the overall frequency of symbols in the first observed sequence (uniform).
- (II) Use only the second observed sequence (x_t) in estimating the parameters, again in the maximum likelihood sense. The initial state distribution is again uniform across the four symbols.

We assume that both estimation processes will be successful relative to their criteria.

- a) **(3 points)** What are the observation probabilities $P(x|s)$ ($x \in \{1, 2\}$, $s \in \{A, G, T, C\}$) resulting from the first estimation approach? (should not require much calculation)

$P(x = 1|s = A) = 1$, $P(x = 1|s = G) = 1$, $P(x = 2|s = T) = 1$, $P(x = 2|s = C) = 1$, all other probabilities are zero.

- b) **(3 points)** Which estimation approach is likely to yield a more accurate model over the second observed sequence (x_t) ? Briefly explain why.

The second one (II) since we can use the available four states to exactly capture the variability in the x_t sequence. Using the observation probabilities above, we'd get a model which assigns probability $1/4$ to each observed sequence of the above type (the only thing to predict is the starting state).

5. Consider now the two HMMs resulting from using each of the estimation approaches (approaches I and II above). These HMMs are estimated on the basis of the pair of observed sequences given above. We'd like to evaluate the probability that these two models assign to a new (different) observation sequence 1 2 1 2, i.e., $x_0 = 1, x_1 = 2, x_2 = 1, x_3 = 2$. For the first model, for which we have some idea about what the s_t variables will capture, we also want to know the the associated most likely hidden state sequence. (these should not require much calculation)

- a) **(2 points)** What is the probability that the first model (approach I) assigns to this new sequence of observations?

1/16

- b) **(2 points)** What is the probability that the second model (approach II) gives to the new sequence of observations?

zero (generates only repeated symbol sequences of the type 1 1 2 2 ...)

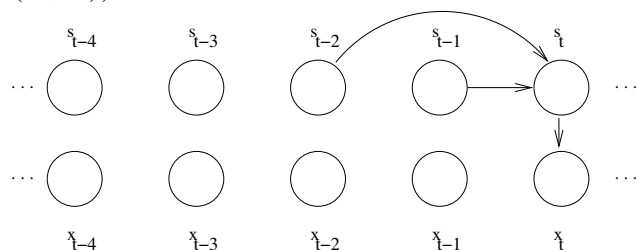
- c) **(2 points)** What is the most likely hidden state sequence in the first model (from approach I) associated with the new observed sequence?

A T G C

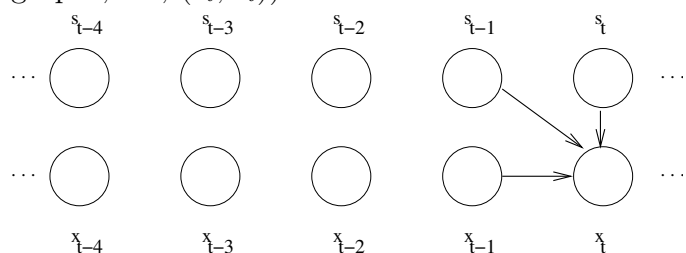
6. **(4 points)** Finally, let's assume that we observe only the second sequence (x_t) (the same sequence as given above). In building a graphical model over this sequence we are no longer limiting ourselves to HMMs. However, we only consider models whose structure/parameters remain the same as we slide along the sequence. The variables s_t are included as before as they might come handy as hidden variables in predicting the observed sequence.

- a) In the figure below, draw the arrows that any reasonable model selection criterion would find given an unlimited supply of the observed sequence x_t, x_{t+1}, \dots . You

only need to draw the arrows for the last pair of variables in the graphs, i.e., (s_t, x_t) .



- b) Given only a small number of observations, the model selection criterion might select a different model. In the figure below, indicate a possible alternate model that any reasonable model selection criterion would find given only a few examples. You only need to draw the arrows for the last pair of variables in the graphs, i.e., (s_t, x_t) .



Problem 5

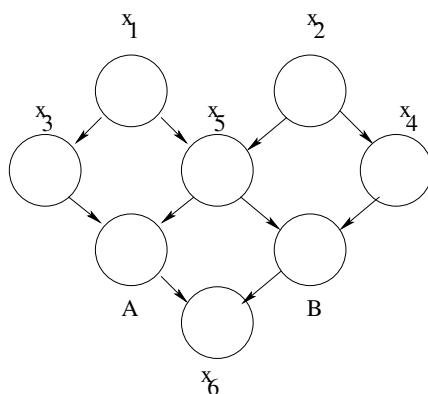


Figure 1: Decision makers A and B and their “context”

The Bayesian network in figure 1 claims to model how two people, call them A and B, make decisions in different contexts. The context is specified by the setting of the binary context variables x_1, x_2, \dots, x_6 . The values of these variables are not known unless we specifically ask for such values.

1. **(6 points)** We are interested in finding out what information we'd have to acquire to ensure that A and B will make their decisions independently from one another. Specify the *smallest set of context variables* whose instantiation would render A and B independent. Briefly explain your reasoning (there may be more than one strategy for arriving at the same decision)

Context variables = $\{x_5 \text{ and } x_1\}$ or $\{x_5 \text{ and } x_2\}$.

Since x_6 is unobserved, it “drops out”. We have to find the remaining context variables that serve as common causes for the decisions. x_5 is one but knowing its value would render x_1 and x_2 dependent. So we have to additionally observe one of them.

You could also solve this by formally using the d-separation criterion.

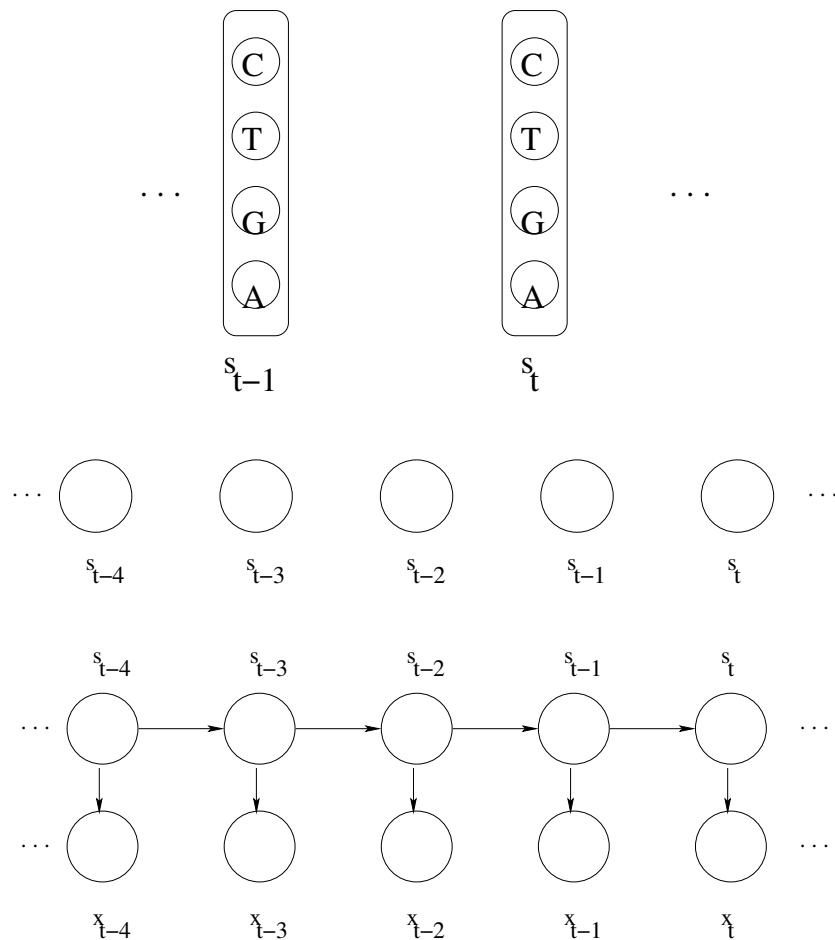
2. **(T/F – 2 points)** We can in general achieve independence with less information, i.e., we don't have to fully instantiate the selected context variables but provide some evidence about their values

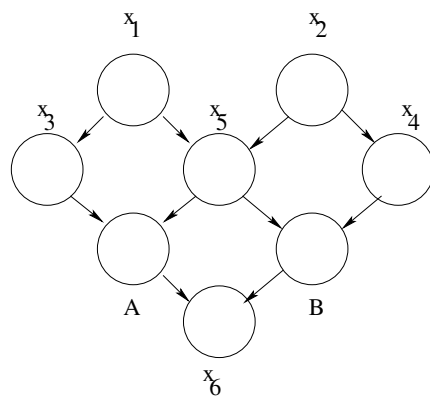
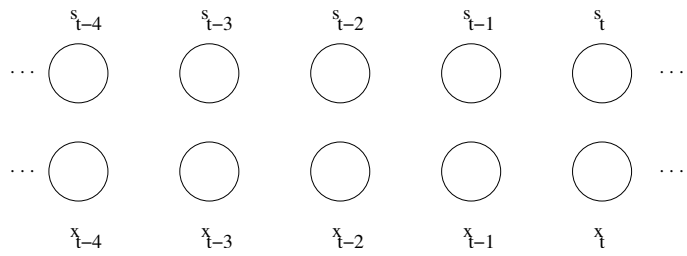
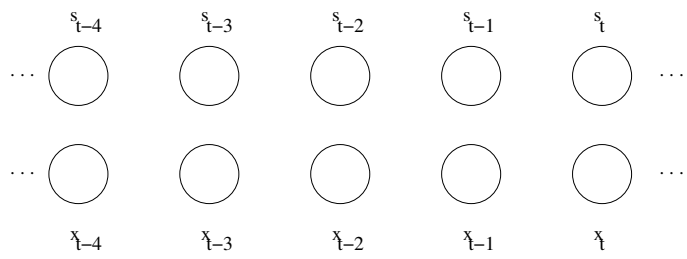
F

3. (4 points) Could your choice of the minimal set of context variables change if we also provided you with the actual probability values associated with the dependencies in the graph? Provide a brief justification.

Our answers could change. The probability values might imply additional independencies and therefore reduce the number of context variables we have to know the values for.

Additional set of figures





6.867 Machine learning

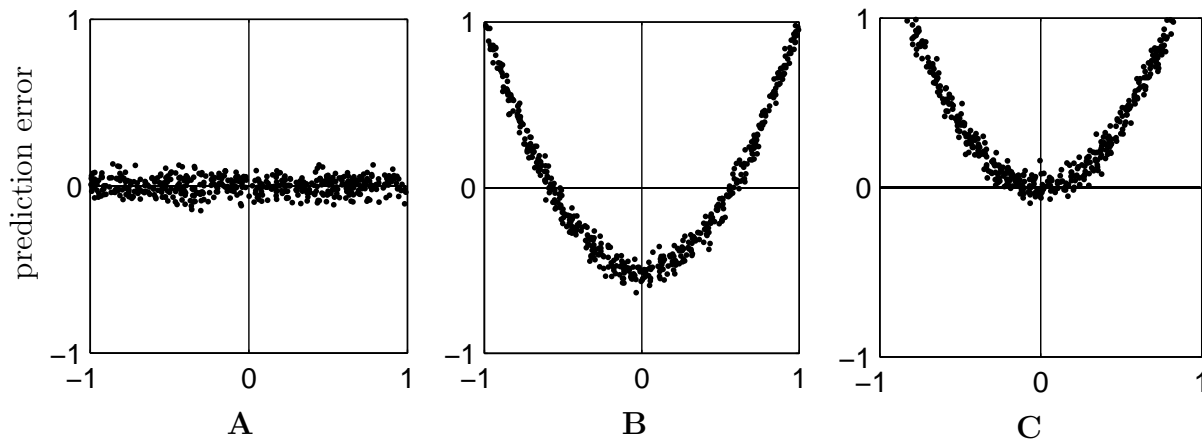
Mid-term exam

October 13, 2004

(2 points) Your name and MIT ID:

T. Assistant, 968672004

Problem 1



1. (6 points) Each plot above claims to represent prediction errors as a function of x for a trained regression model based on some dataset. Some of these plots could potentially be prediction errors for linear or quadratic regression models, while others couldn't. The regression models are trained with the least squares estimation criterion. Please indicate compatible models and plots.

	A	B	C
linear regression	(x)	(x)	()
quadratic regression	(x)	()	()

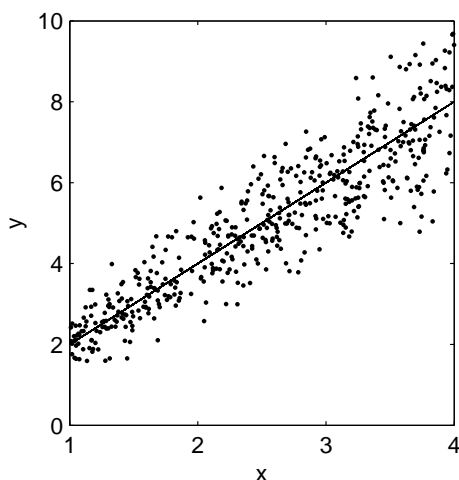
Problem 2

Here we explore a regression model where the noise variance is a function of the input (variance increases as a function of input). Specifically

$$y = wx + \epsilon$$

where the noise ϵ is normally distributed with mean 0 and standard deviation σx . The value of σ is assumed known and the input x is restricted to the interval $[1, 4]$. We can write the model more compactly as $y \sim N(wx, \sigma^2 x^2)$.

If we let x vary within $[1, 4]$ and sample outputs y from this model with some w , the regression plot might look like



1. **(2 points)** How is the ratio y/x distributed for a fixed (constant) x ?

Since $y \sim N(wx, \sigma^2 x^2)$, for any constant x , y/x is also Gaussian with mean w and variance $\sigma^2 x^2 / x^2 = \sigma^2$. So, $y/x \sim N(w, \sigma^2)$.

2. Suppose we now have n training points and targets $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, where each x_i is chosen at random from $[1, 4]$ and the corresponding y_i is subsequently sampled from $y_i \sim N(w^* x_i, \sigma^2 x_i^2)$ with some true underlying parameter value w^* ; the value of σ^2 is the same as in our model.

- (a) **(3 points)** What is the maximum-likelihood estimate of w as a function of the training data?

We know that $y/x \sim N(w, \sigma^2)$. We can therefore estimate w by interpreting y_i/x_i as observations. The maximum likelihood estimate of w is simply the mean

$$\hat{w}_n = \frac{1}{n} \sum y_i/x_i$$

- (b) **(3 points)** What is the variance of this estimator due to the noise in the target outputs as a function of n and σ^2 for fixed inputs x_1, \dots, x_n ? For later utility (if you omit this answer) you can denote the answer as $V(n, \sigma^2)$.

The variance of the estimator of a mean of a gaussian is σ^2/n .

Some potentially useful relations: if $z \sim N(\mu, \sigma^2)$, then $az \sim N(a\mu, \sigma^2 a^2)$ for a fixed a . If $z_1 \sim N(\mu_1, \sigma_1^2)$ and $z_2 \sim N(\mu_2, \sigma_2^2)$ and they are independent, then $\text{Var}(z_1 + z_2) = \sigma_1^2 + \sigma_2^2$.

3. In sequential active learning we are free to choose the next training input x_{n+1} , here within $[1, 4]$, for which we will then receive the corresponding noisy target y_{n+1} , sampled from the underlying model. Suppose we already have $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ and are trying to figure out which x_{n+1} to select. The goal is to choose x_{n+1} so as to help minimize the variance of the predictions $f(x; \hat{w}_n) = \hat{w}_n x$, where \hat{w}_n is the maximum likelihood estimate of the parameter w based on the first n training examples.

- (a) **(2 points)** What is the variance of $f(x; \hat{w}_n)$ due to the noise in the training outputs as a function of x , n , and σ^2 given fixed (already chosen) inputs x_1, \dots, x_n ?

$f(x; \hat{w}_n)$ is xw , and we know that the variance of w is σ^2/n (from the previous part). Thus the variance of $f(x; \hat{w}_n)$ is $x^2\sigma^2/n$.

- (b) **(2 points)** Which x_{n+1} would we choose (within $[1, 4]$) if we were to next select x with the maximum variance of $f(x; \hat{w}_n)$?

The variance is maximized when x^2 is maximum, that is $x = 4$.

- (c) **(T/F – 2 points)** Since the variance of $f(x; \hat{w}_n)$ only depends on x , n , and σ^2 , we could equally well select the next point at random from $[1, 4]$ and obtain the same reduction in the maximum variance.

T

The variance at $x = 4$ is $16\sigma^2/n$. This does not depend on the actual choice of the queried points, but only on the number of points queried.

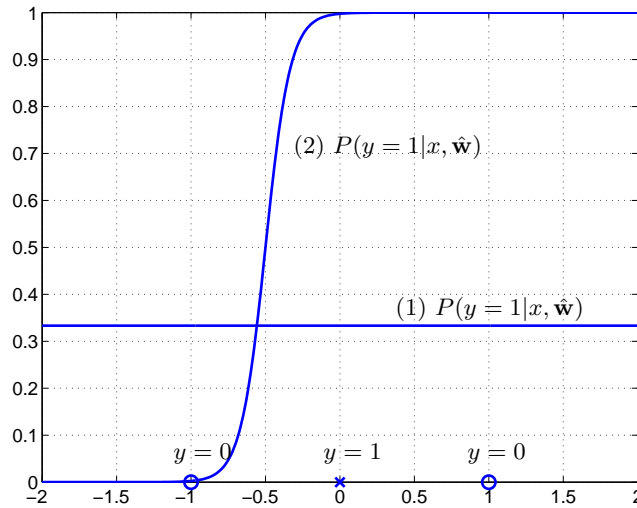


Figure 1: Two possible logistic regression solutions for the three labeled points.

Problem 3

Consider a simple one dimensional logistic regression model

$$P(y = 1|x, \mathbf{w}) = g(w_0 + w_1x)$$

where $g(z) = (1 + \exp(-z))^{-1}$ is the logistic function.

- Figure 1 shows two possible conditional distributions $P(y = 1|x, \mathbf{w})$, viewed as a function of x , that we can get by changing the parameters \mathbf{w} .

- (2 points)** Please indicate the number of classification errors for each conditional given the labeled examples in the same figure

Conditional (1) makes (1) classification errors

Conditional (2) makes (1) classification errors

- (3 points)** One of the conditionals in Figure 3 corresponds to the maximum likelihood setting of the parameters $\hat{\mathbf{w}}$ based on the labeled data in the figure. Which one is the ML solution (1 or 2)?

1

The likelihood under model (2) is 0, because model (2) assigns 0 probability to the sample at 1. The likelihood under model (1) is $2/3 \cdot 1/3 \cdot 2/3$.

- (2 points)** Would adding a regularization penalty $|w_1|^2/2$ to the log-likelihood estimation criterion affect your choice of solution (Y/N)?

N

At maximum likelihood $\hat{w}_1 = 0$ (cf. Conditional (1)), thus $|\hat{w}_1|^2/2$ is minimal even without a regularization penalty.

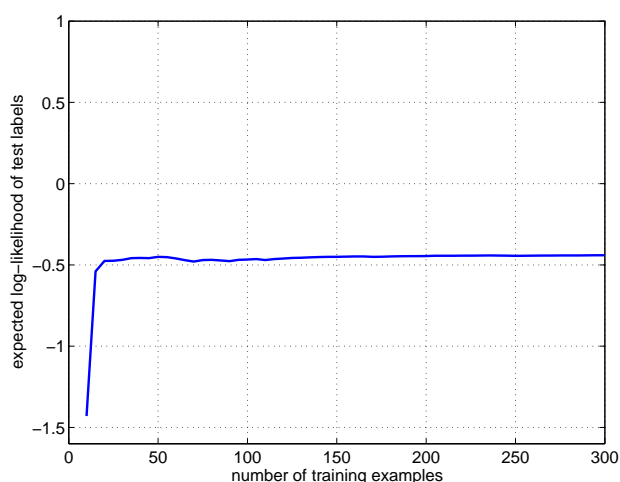


Figure 2: The expected log-likelihood of test labels as a function of the number of training examples.

2. **(4 points)** We can estimate the logistic regression parameters more accurately with more training data. Figure 2 shows the expected log-likelihood of test labels for a simple logistic regression model as a function of the number of training examples and labels. *Mark in the figure* the structural error (SE) and approximation error (AE), where “error” is measured in terms of log-likelihood.

SE is the distance from the horizontal part of the graph to $y = 0$. AE is everything below the horizontal part of the graph.

3. **(T/F – 2 points)** In general for small training sets, we are likely to reduce the approximation error by adding a regularization penalty $|w_1|^2/2$ to the log-likelihood criterion.

T

Regularization prevents over-fitting by constraining the input to the logistic function to be a smooth function of input. Such functions can be estimated with fewer samples. Put another way, you are reducing the variance of the estimator in favor of introducing some bias. Variance dominates when we have only few training samples.

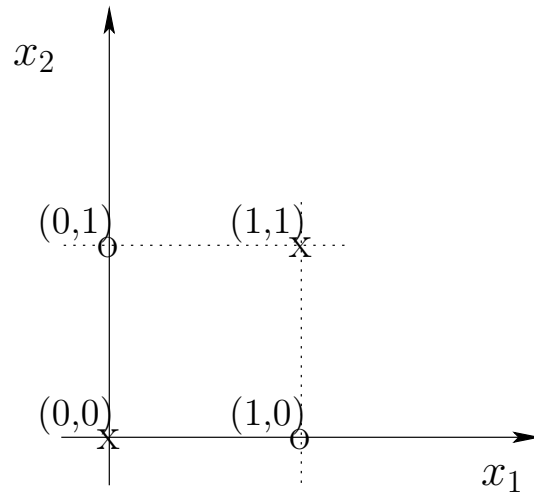


Figure 3: Equally likely input configurations in the training set

Problem 4

Here we will look at methods for selecting input features for a logistic regression model

$$P(y = 1|\mathbf{x}, \mathbf{w}) = g(w_0 + w_1x_1 + w_2x_2)$$

The available training examples are very simple, involving only binary valued inputs:

Number of copies	x_1	x_2	y
10	1	1	1
10	0	1	0
10	1	0	0
10	0	0	1

So, for example, there are 10 copies of $\mathbf{x} = [1, 1]^T$ in the training set, all labeled $y = 1$. The correct label is actually a deterministic function of the two features: $y = 1$ if $x_1 = x_2$ and zero otherwise.

We define greedy selection in this context as follows: we start with no features (train only with w_0) and successively try to add new features provided that each addition strictly improves the training log-likelihood. We use no other stopping criterion.

1. **(2 points)** Could greedy selection add either x_1 or x_2 in this case?
Answer Y or N.

N

We have equally many $y = 1$ and $y = 0$ examples under $x_1 = 1$, and the same is true for $x_1 = 0$. Thus we cannot bias the probabilities in any way based on the presence of x_1 alone. The same is true for x_2 .

2. **(2 points)** What is the classification error of the training examples that we could achieve by including both x_1 and x_2 in the logistic regression model?

0.25

3. **(3 points)** Suppose we define another possible feature to include, a function of x_1 and x_2 . Which of the following features, if any, would permit us to correctly classify all the training examples when used in combination with x_1 and x_2 in the logistic regression model:

$$\begin{array}{ll} (\quad) & x_1 - x_2 \\ (\text{ x }) & x_1 x_2 \\ (\quad) & x_2^2 \end{array}$$

$x_1 - x_2$ is a linear combination of existing features, thus adding it does not change the model in any way. The same is true for x_2^2 , because $x_2^2 = x_2$ for $x_2 \in \{0, 1\}$.

Now suppose we model the conditional by $P(y = 1|\mathbf{x}, \mathbf{w}) = g(w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1 x_2)$. Let w_1 and w_2 be such that when $w_3 = 0$ only $(1, 1)$ is misclassified. Since $x_1 x_2 = 0$ except at $(1, 1)$, w_3 does not affect the classification of the correctly classified training points. We can then choose w_3 such that $(1, 1)$ is also correctly classified.

4. **(2 points)** Could the greedy selection method choose this feature as the first feature to add when the available features are x_1 , x_2 and your choice of the new feature? Answer Y or N.

Y

Problem 5

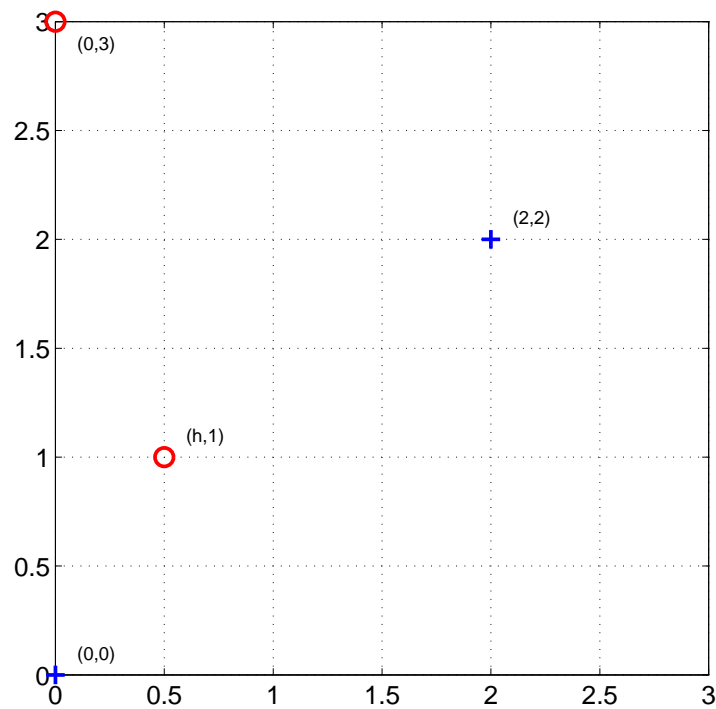


Figure 4: Labeled training examples

Suppose we only have four training examples in two dimensions (see Figure 4):

positive examples at $\mathbf{x}_1 = [0, 0]^T$, $\mathbf{x}_2 = [2, 2]^T$ and
negative examples at $\mathbf{x}_3 = [h, 1]^T$, $\mathbf{x}_4 = [0, 3]^T$.

where we treat $0 \leq h \leq 3$ as a parameter.

1. **(2 points)** How large can $h \geq 0$ be so that the training points are still linearly separable?
2. **(2 points)** Does the orientation of the maximum margin decision boundary change as a function of h when the points are separable? Answer Y or N.

$h \leq 1$

N

3. **(4 points)** What is the margin achieved by the maximum margin boundary as a function of h ?

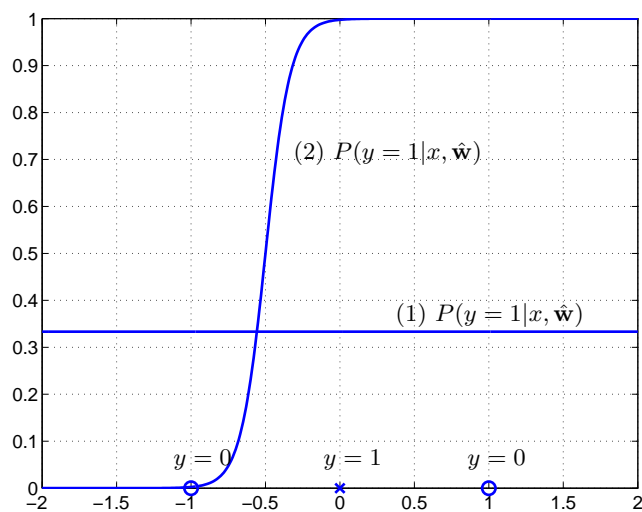
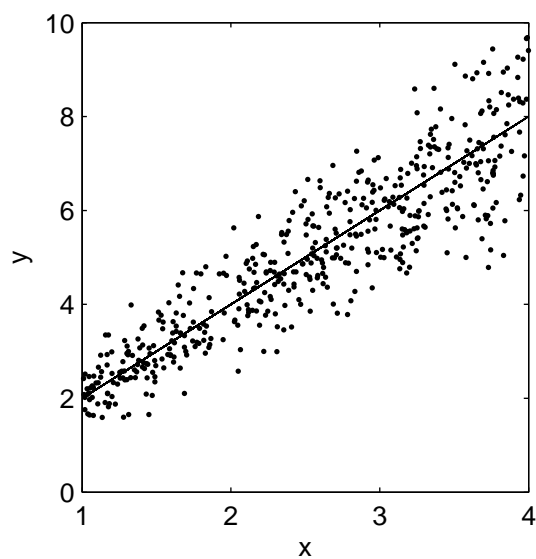
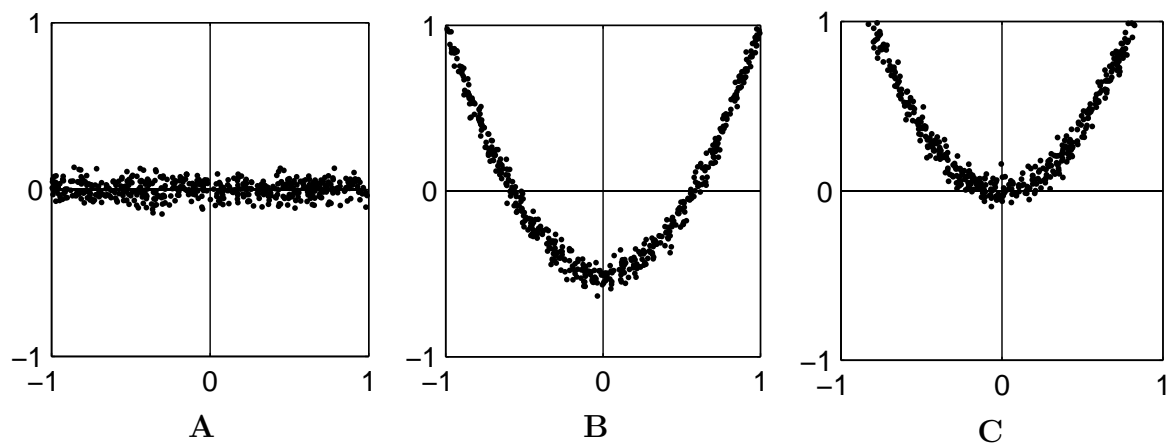
The margin is $(1 - h)/\sqrt{2}$, for $0 \leq h \leq 1$, and 0 for $h > 1$.

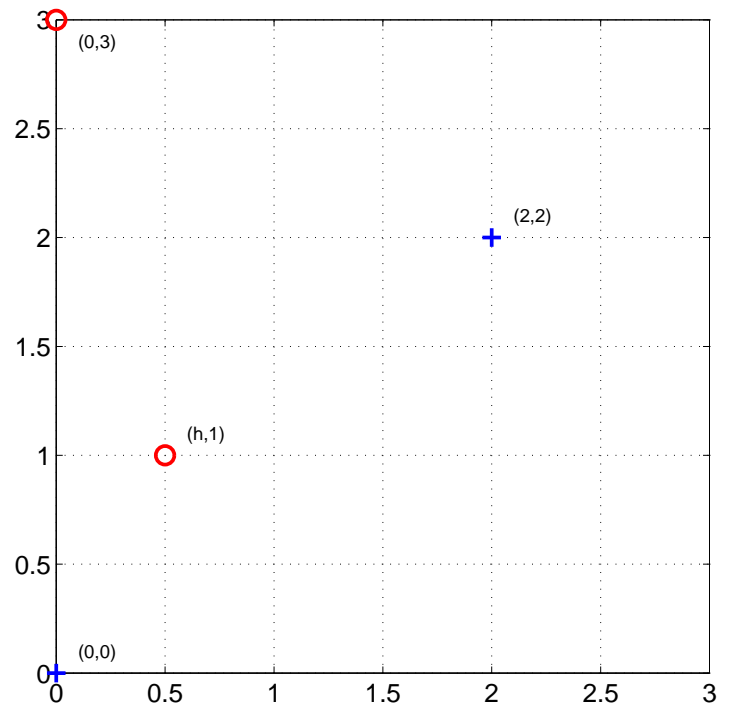
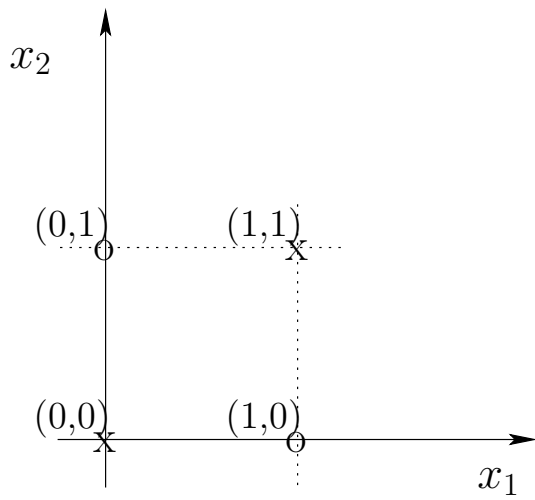
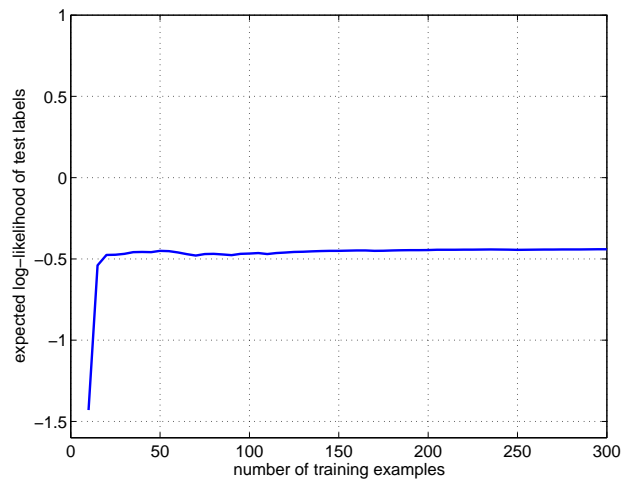
One way to calculate the margin is as follows: since the orientation of the boundary does not change as a function h , the margin is a linear function of h . The margin is zero at $h = 1$ and $1/\sqrt{2}$ at $h = 0$.

4. **(3 points)** Assume that $h = 1/2$ (as in the figure) and that we can only observe the x_2 -component of the input vectors. Without the other component, the labeled training points reduce to $(0, y = 1)$, $(2, y = 1)$, $(1, y = -1)$, and $(3, y = -1)$. What is the lowest order p of polynomial kernel that would allow us to correctly classify these points?

3

Additional set of figures





6.867 Machine learning

Mid-term exam

October 15, 2003

(2 points) Your name and MIT ID:

SOLUTIONS

Problem 1

Suppose we are trying to solve an active learning problem, where the possible inputs you can select form a discrete set. Specifically, we have a set of N unlabeled documents, Φ_1, \dots, Φ_N , where each document is represented as a binary feature vector

$$\Phi = [\phi_1, \dots, \phi_m]^T$$

and $\phi_i = 1$ if word i appears in the document and zero otherwise. Our goal is to quickly label these N documents with 0/1 labels. We can request a label for any of the N documents, preferably as few as possible. We also have a small number n of these documents already labeled to get us started.

We use a logistic regression model to solve the classification task:

$$P(y = 1|\Phi, \mathbf{w}) = g(\mathbf{w}^T \Phi)$$

where $g(\cdot)$ is the logistic function. Note that we do not include the bias term.

1. **(T/F – 2 points)** Any word that appears in all the N documents would effectively provide a bias term for the logistic regression model.
2. **(T/F – 2 points)** Any word that appears only in the available n labeled documents used for initially training the logistic regression model, would serve equally well as a bias term.

T

F

3. Having trained the logistic regression model on the basis of the n labeled documents, obtaining $\hat{\mathbf{w}}_n$, we'd like to request additional labeled documents. For this, we will use the following measure of uncertainty in our predictions:

$$E_{y \sim p_t} |y - p_t| = p_t |1 - p_t| + (1 - p_t) |0 - p_t| = 2p_t(1 - p_t)$$

where $p_t = P(y = 1 | \Phi_t, \hat{\mathbf{w}}_n)$, our current prediction of the probability that $y = 1$ for the t^{th} unlabeled document Φ_t .

- a) **(4 points)** We would request the label for the document/query point Φ_t that has
- () the smallest value of $2p_t(1 - p_t)$
 - (X) the largest value of $2p_t(1 - p_t)$
 - () an intermediate value of $2p_t(1 - p_t)$

Briefly explain the rationale behind the selection criterion that you chose.

$2p_t(1 - p_t)$ is a measure of uncertainty about the label for document t . We expect to be able to reduce the uncertainty the most by requesting labels for those documents with the largest value of $2p_t(1 - p_t)$, points that are closest to the decision boundary. Put another way, getting additional training examples that are close to the boundary ought to help the most in terms of figuring out exactly where the boundary should lie.

- b) **(2 points)** Sketch $\hat{\mathbf{w}}_n$ in Figure 1.1. Write down the equation, expressed solely in terms of Φ and $\hat{\mathbf{w}}_n$, that Φ has to satisfy for it to lie exactly on the decision boundary:

Points Φ that lie on the decision boundary must satisfy $\hat{\mathbf{w}}_n^T \Phi = 0$. Hence, $\hat{\mathbf{w}}_n^T$ is orthogonal to the decision boundary. Moreover, since we decide $y = 1$ when $\hat{\mathbf{w}}_n^T \Phi > 0$, $\hat{\mathbf{w}}_n^T$ points into the '+' region.

- c) **(4 points)** In figure 1.2, circle the next point we would select according to the criterion. Draw two decision boundaries that would result from incorporating the new point in the training set, labeling the boundaries as $y = 1$ and $y = 0$, depending on the outcome of the query.

The uncertainty is largest when $P(y = 1|\Phi) = g(\hat{\mathbf{w}}_n^T \Phi)$ is near one-half, e.g. when $\hat{\mathbf{w}}_n^T \Phi$ is near zero so that the point Φ is close to the decision boundary. Hence, we circle the point nearest the boundary. If this point (initially in the $y = 0$ region) turns out to be a $y = 1$ document, then we "tilt" the boundary towards that point so as to tend to move that example to the other side of the decision boundary. Otherwise, if the document is in fact a $y = 0$, then we tilt the boundary away from that point so that it is "deeper" in the $y = 0$ region.

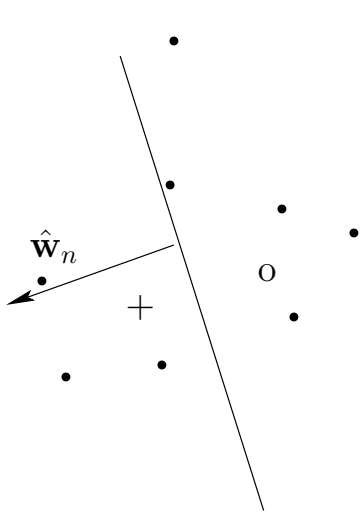


Figure 1.1. Two labeled points, unlabeled points, and the decision boundary. The point "+" corresponds to $y = 1$.

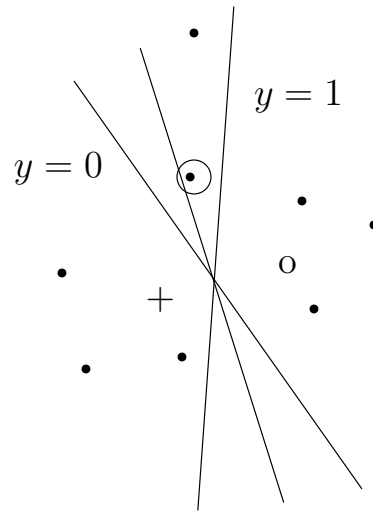


Figure 1.2. Two labeled points, unlabeled points, and the decision boundary. The point "+" corresponds to $y = 1$.

4. **(T/F – 2 points)** The criterion we have used here for active learning guarantees that the measure of uncertainty about the labels of the unlabeled points will decrease monotonically for each point after each query.

F

Roughly speaking, when we "tilt" the boundary to incorporate each new labeled example some unlabeled points may end up closer to the boundary which tends to increase the uncertainty in those points.

Problem 2

Consider a regression problem where the two dimensional input points $\mathbf{x} = [x_1, x_2]^T$ are constrained to lie within the unit square: $x_i \in [-1, 1]$, $i = 1, 2$. The training and test input points \mathbf{x} are sampled uniformly at random within the unit square. The target outputs y are governed by the following model

$$y \sim N(x_1^3 x_2^5 - 10x_1 x_2 + 7x_1^2 + 5x_2 - 3, 1)$$

In other words, the outputs are normally distributed with mean given by

$$x_1^3 x_2^5 - 10x_1 x_2 + 7x_1^2 + 5x_2 - 3$$

and variance 1.

We learn to predict y given \mathbf{x} using linear regression models with 1st through 10th order polynomial features. The models are nested in the sense that the higher order models will include all the lower order features. The estimation criterion is the mean squared error.

We first train a 1st, 2nd, 8th, and 10th order model using $n = 20$ training points, and then test the predictions on a large number of independently sampled points.

1. **(6 points)** Select all the appropriate model(s) for each column. If you think the highest, or lowest, error would be shared among several models, be sure to list all models.

	Lowest training error	Highest training error	Lowest test error (typically)
1st order	()	(X)	
2nd order	()	()	(X)
8th order	(X)	()	
10th order	(X)	()	()

Briefly explain your selection in the last column, i.e., the model you would expect to have the lowest test error:

The 10th order regression model would seriously overfit when presented only with $n = 20$ training points. The second order model on the other hand might find some useful structure in the data based only on 20 points. The true model is also dominated by the second order terms. Since $|x_1| \leq 1$ and $|x_2| \leq 1$ any higher order terms without large coefficients are vanishingly small.

2. **(6 points)** We now train the polynomial regression models using $n = 10^6$ (one million) training points. Again select the appropriate model(s) for each column. If

you think the highest, or lowest, error would be shared among several models, be sure to list all models.

	Lowest structural error	Highest approx. error	Lowest test error
1st order	()	()	()
2nd order	()	()	()
8th order	(X)	()	(X)
10th order	(X)	(X)	()

3. **(T/F – 2 points)** The approximation error of a polynomial regression model depends on the number of training points.

T

4. **(T/F – 2 points)** The structural error of a polynomial regression model depends on the number of training points.

F

Problem 3

We consider here linear and non-linear support vector machines (SVM) of the form:

$$\begin{aligned} \min w_1^2/2 \quad & \text{subject to} \quad y_i(w_1x_i + w_0) - 1 \geq 0, \quad i = 1, \dots, n, \quad \text{or} \\ \min \mathbf{w}^T \mathbf{w}/2 \quad & \text{subject to} \quad y_i(\mathbf{w}^T \Phi_i + w_0) - 1 \geq 0, \quad i = 1, \dots, n \end{aligned}$$

where Φ_i is a feature vector constructed from the corresponding real valued input x_i . We wish to compare the simple linear SVM classifier ($w_1x + w_0$) and the non-linear classifier ($\mathbf{w}^T \Phi + w_0$), where $\Phi = [x, x^2]^T$.

1. **(3 points)** Provide three input points x_1, x_2 , and x_3 and their associated ± 1 labels such that they cannot be separated with the simple linear classifier, but are separable by the non-linear classifier with $\Phi = [x, x^2]^T$. You may find Figure 3.1. helpful in answering this question.

$(x = 1, y = 1), (x = 2, y = -1), (x = 3, y = 1)$

2. **(3 points)** In the figure below (Figure 3.1), mark your three points x_1, x_2 , and x_3 as points in the feature space with their associated labels. Draw the *decision boundary* of the non-linear SVM classifier with $\Phi = [x, x^2]^T$ that separates the points.

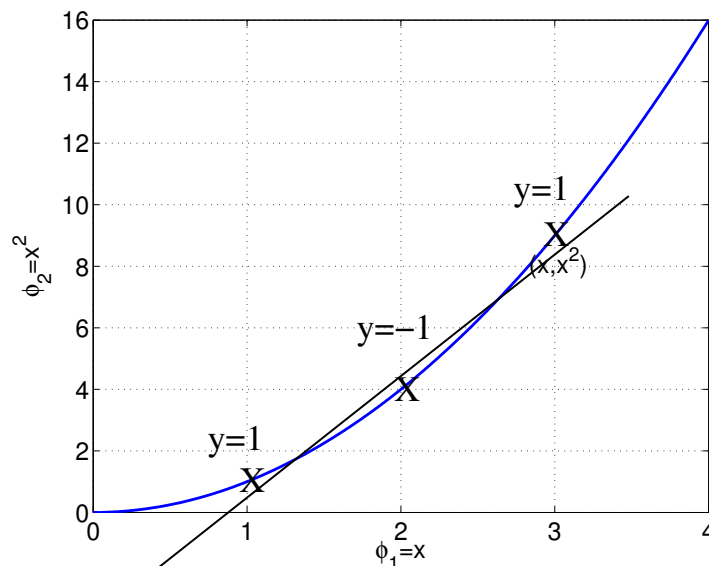


Figure 3.1. Feature space.

3. **(3 points)** Consider two labeled points $(x = 1, y = 1)$ and $(x = 3, y = -1)$. Is the margin we attain using feature vectors $\Phi = [x, x^2]^T$

☒ greater

☐ equal

☐ smaller

than the margin resulting from using the input x directly?

4. **(2 points)** In general, is the margin we would attain using scaled feature vectors $\Phi = [2x, 2x^2]^T$

☒ greater

☐ equal

☐ smaller

☐ any of the above

in comparison to the margin resulting from using $\Phi = [x, x^2]^T$?

5. **(T/F – 2 points)** The values of the margins obtained by two different kernels $K(x, x')$ and $\tilde{K}(x, x')$ on the same training set do not tell us which classifier will perform better on the test set.

T

We need to normalize the margin for it to be meaningful. For example, a simple scaling of the feature vectors would lead to a larger margin. Such a scaling does not change the decision boundary, however, and so the larger margin cannot directly inform us about generalization.

Problem 4

We consider here generative and discriminative approaches for solving the classification problem illustrated in Figure 4.1. Specifically, we will use a mixture of Gaussians model and regularized logistic regression models.

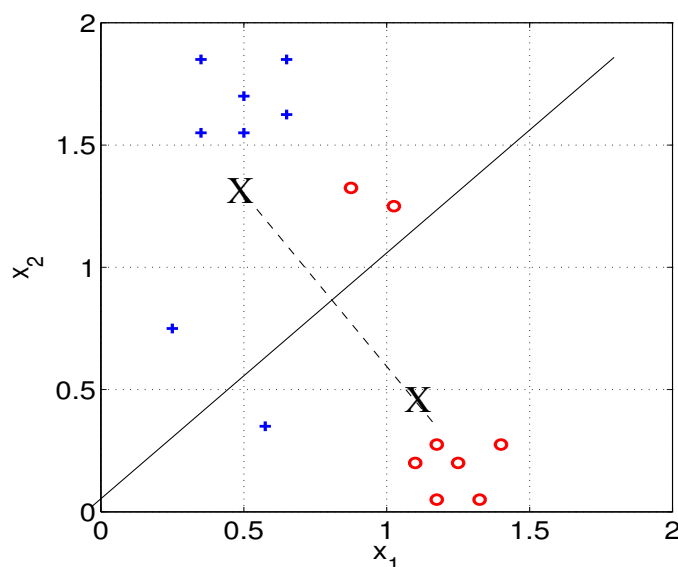


Figure 4.1. Labeled training set, where “+” corresponds to class $y = 1$.

1. We will first estimate a mixture of Gaussians model, one Gaussian per class, with the constraint that the covariance matrices are identity matrices. The mixing proportions (class frequencies) and the means of the two Gaussians are free parameters.
 - a) **(3 points)** Plot the maximum likelihood estimates of the means of the two class conditional Gaussians in Figure 4.1. Mark the means as points “x” and label them “0” and “1” according to the class.
The means should be close to the center of mass of the points.
 - b) **(2 points)** Draw the decision boundary in the same figure.
Since the two classes have the same number of points and the same covariance matrices, the decision boundary is a line and, moreover, should be drawn as the orthogonal bisector of the line segment connecting the class means.

2. We have also trained regularized linear logistic regression models

$$P(y = 1|\mathbf{x}, \mathbf{w}) = g(w_0 + w_1x_1 + w_2x_2)$$

for the same data. The regularization penalties, used in penalized conditional log-likelihood estimation, were $-Cw_i^2$, where $i = 0, 1, 2$. In other words, only one of the parameters were regularized in each case. Based on the data in Figure 4.1, we generated three plots, one for each regularized parameter, of the number of misclassified training points as a function of C (Figure 4.2). The three plots are not identified with the corresponding parameters, however. Please assign the “top”, “middle”, and “bottom” plots to the correct parameter, w_0 , w_1 , or w_2 , the parameter that was regularized in the plot. Provide a brief justification for each assignment.

- **(3 points)** “top” = (w_1)

By strongly regularizing w_1 we force the boundary to be horizontal in the figure. The logistic regression model tries to maximize the log-probability of classifying the data correctly. The highest penalty comes from the misclassified points and thus the boundary will tend to balance the (worst) errors. In the figure, this is roughly speaking $x_2 = 1$ line, resulting in 4 errors.

- **(3 points)** “middle” = (w_0)

If we regularize w_0 , then the boundary will eventually go through the origin (bias term set to zero). Based on the figure we can find a good linear boundary through the origin with only one error.

- **(3 points)** “bottom” = (w_2)

The training error is unaffected if we regularize w_2 (constrain the boundary to be vertical); the value of w_2 would be small already without regularization.

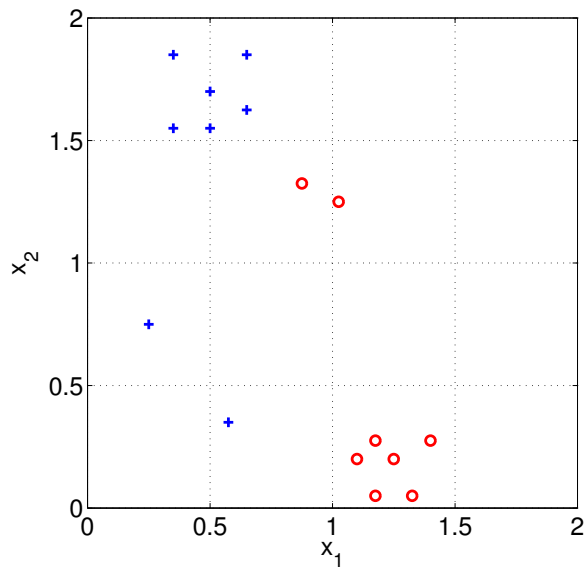


Figure 4.1 Labeled training set
(reproduced here for clarity)

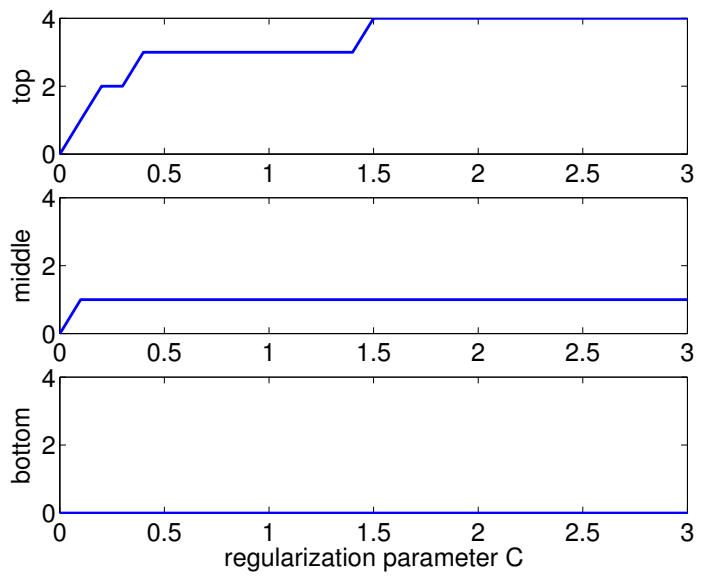


Figure 4.2. Training errors as a function
of regularization penalty

6.867 Machine learning

Mid-term exam

October 22, 2002

(2 points) Your name and MIT ID:

Problem 1

We are interested here in a particular 1-dimensional linear regression problem. The dataset corresponding to this problem has n examples $(x_1, y_1), \dots, (x_n, y_n)$, where x_i and y_i are real numbers for all i . Part of the difficulty here is that we don't have access to the inputs or outputs directly. We don't even know the number of examples in the dataset. We are, however, able to get a few numbers computed from the data.

Let $\mathbf{w}^* = [w_0^*, w_1^*]^T$ be the least squares solution we are after. In other words, \mathbf{w}^* minimizes

$$J(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^n (y_i - w_0 - w_1 x_i)^2$$

You can assume for our purposes here that the solution is unique.

1. (4 points) Check each statement that must be true if $\mathbf{w}^* = [w_0^*, w_1^*]^T$ is indeed the least squares solution

$$\begin{aligned} & () \quad (1/n) \sum_{i=1}^n (y_i - w_0^* - w_1^* x_i) y_i = 0 \\ & () \quad (1/n) \sum_{i=1}^n (y_i - w_0^* - w_1^* x_i) (y_i - \bar{y}) = 0 \\ & (x) \quad (1/n) \sum_{i=1}^n (y_i - w_0^* - w_1^* x_i) (x_i - \bar{x}) = 0 \\ & (x) \quad (1/n) \sum_{i=1}^n (y_i - w_0^* - w_1^* x_i) (w_0^* + w_1^* x_i) = 0 \end{aligned}$$

where \bar{x} and \bar{y} are the sample means based on the same dataset.

Taking the derivative with respect to w_1 and w_0 gives us the following conditions of optimality (as in the lectures)

$$\begin{aligned}\frac{\partial}{\partial w_0} J(\mathbf{w}) &= \frac{2}{n} \sum_{i=1}^n (y_i - w_0 - w_1 x_i) = 0 \\ \frac{\partial}{\partial w_1} J(\mathbf{w}) &= \frac{2}{n} \sum_{i=1}^n (y_i - w_0 - w_1 x_i) x_i = 0\end{aligned}$$

This means that the prediction error $(y_i - w_0 - w_1 x_i)$ does not co-vary with any linear function of the inputs (has a zero mean and does not co-vary with the inputs). $(x_i - \bar{x})$ and $(w_0^* + w_1^* x_i)$ are both linear functions of inputs.

2. **(4 points)** There are several numbers (statistics) computed from the data that we can use to infer \mathbf{w}^* . These are

$$\begin{aligned}\bar{x} &= \frac{1}{n} \sum_{i=1}^n x_i, \quad \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i, \quad C_{xx} = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 \\ C_{xy} &= \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}), \quad C_{yy} = \frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2\end{aligned}$$

Suppose we only care about the value of w_1^* . We'd like to determine w_1^* on the basis of only two numbers (statistics) listed above. Which two numbers do we need for this?

We need C_{xx} (spread of x) and C_{xy} (linear dependence between x and y). No justification was necessary as these basic points have appeared repeatedly in the course. If we want to derive these more mathematically, we can, for example, look at one of the answers to the previous question:

$$\begin{aligned}\frac{1}{n} \sum_{i=1}^n (y_i - w_0^* - w_1^* x_i)(x_i - \bar{x}) &= 0, \text{ which we can rewrite as} \\ \left[\frac{1}{n} \sum_{i=1}^n y_i (x_i - \bar{x}) \right] - w_0^* \left[\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x}) \right] - w_1^* \left[\frac{1}{n} \sum_{i=1}^n x_i (x_i - \bar{x}) \right] &= 0\end{aligned}$$

By using the fact that $(1/n) \sum_i (x_i - \bar{x}) = 0$ we see that

$$\begin{aligned}\frac{1}{n} \sum_{i=1}^n y_i (x_i - \bar{x}) &= \frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})(x_i - \bar{x}) = C_{xy} \\ \frac{1}{n} \sum_{i=1}^n x_i (x_i - \bar{x}) &= \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(x_i - \bar{x}) = C_{xx}\end{aligned}$$

Substituting these back into our equation above gives $C_{xy} - w_1^* C_{xx} = 0$.

3. Here we change the rules governing our access to the data. Instead of simply getting the statistics we want, we have to reconstruct these from examples that we query. There are two types of queries we can make. We can either request additional randomly chosen examples from the training set, or we can query the output corresponding to a specific input that we specify. (We assume that the dataset is large enough that there is always an example whose input x is close enough to our query).

The active learning scenario here is somewhat different from the typical one. Normally we would assume that the data is governed by a linear model and choose the input points so as to best recover this assumed model. Here the task is to recover the best fitting linear model to the data but we make no assumptions about whether the linear model is appropriate in the first place.

(2 points) Suppose in our case the input points are constrained to lie in the interval $[0, 1]$. If we followed the typical active learning approach, where we assume that the true model is linear, what are the input points we would query?

We would query the extreme points $x = 0$ and $x = 1$ as they constrain the linear function the most.

(3 points) In the new setting, where we try to recover the best fitting linear model or parameters \mathbf{w}^* , we should (choose only one):

- ☐ Query inputs as you have answered above
- ☒ Draw inputs and corresponding outputs at random from the dataset
- ☐ Use another strategy since neither of the above choices would yield satisfactory results

(4 points) Briefly justify your answer to the previous question

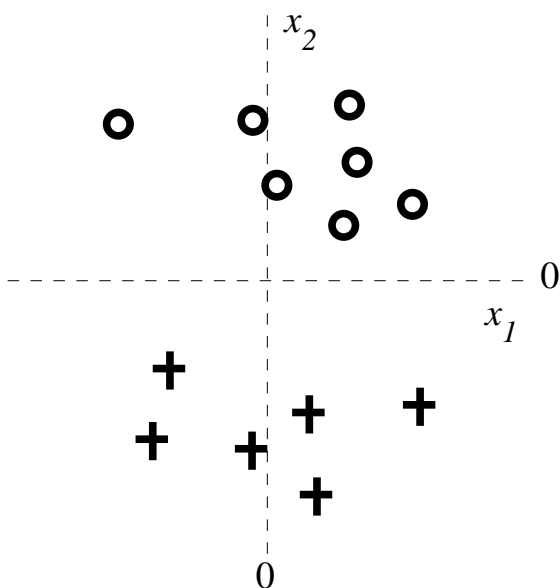
The objective is to recover the least squares solution. This solution depends on the frequency of inputs and outputs in the dataset. Without additional assumptions, the best thing to do is to draw a representative set of examples from the dataset so that the resulting least squares solution would approximate the solution based on the full dataset.

Problem 2

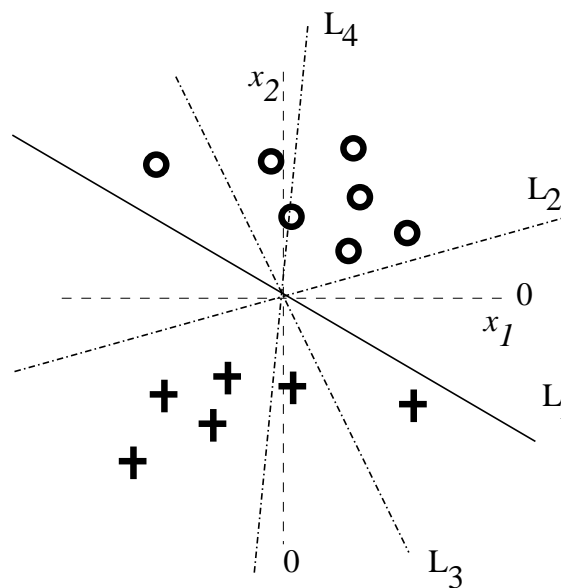
In this problem we will refer to the binary classification task depicted in Figure 1(a), which we attempt to solve with the simple linear logistic regression model

$$\hat{P}(y = 1 | \mathbf{x}, w_1, w_2) = g(w_1 x_1 + w_2 x_2) = \frac{1}{1 + \exp(-w_1 x_1 - w_2 x_2)}$$

(for simplicity we do not use the bias parameter w_0). The training data can be separated with zero training error - see line L_1 in Figure 1(b) for instance.



(a) The 2-dimensional data set used in Problem 1



(b) The points can be separated by L_1 (solid line). Possible other decision boundaries are shown by L_2, L_3, L_4 .

1. (6 points) Consider a regularization approach where we try to maximize

$$\sum_{i=1}^n \log p(y_i | \mathbf{x}_i, w_1, w_2) - \frac{C}{2} w_2^2$$

for large C . Note that **only** w_2 is penalized. We'd like to know which of the four lines in Figure 1(b) could arise as a result of such regularization. For each potential line L_2 , L_3 or L_4 determine whether it can result from regularizing w_2 . If not, explain very briefly why not.

- L_2

No. When we regularize w_2 , the resulting boundary can rely less on the value of x_2 and therefore becomes more vertical. L_2 here seems to be more horizontal than the unregularized solution so it cannot come as a result of penalizing w_2 .

- L_3

Yes. Here w_2^2 is small relative to w_1^2 (as evidenced by high slope), and even though it would assign a rather low log-probability to the observed labels, it could be forced by a large regularization parameter C .

- L_4

No. For very large C , we get a boundary that is entirely vertical (line $x_1 = 0$ or the x_2 axis). L_4 here is reflected across the x_2 axis and represents a poorer solution than it's counter part on the other side. For moderate regularization we have to get the best solution that we can construct while keeping w_2 small. L_4 is not the best and thus cannot come as a result of regularizing w_2 .

2. (4 points) If we change the form of regularization to one-norm (absolute value) and also regularize w_1 we get the following penalized log-likelihood

$$\sum_{i=1}^n \log p(y_i | \mathbf{x}_i, w_1, w_2) - \frac{C}{2} (|w_1| + |w_2|).$$

Consider again the problem in Figure 1(a) and the same linear logistic regression model $\hat{P}(y = 1 | \mathbf{x}, w_1, w_2) = g(w_1 x_1 + w_2 x_2)$. As we increase the regularization parameter C which of the following scenarios do you expect to observe (choose only one):

- (x) First w_1 will become 0, then w_2 .
 () w_1 and w_2 will become zero simultaneously

- () First w_2 will become 0, then w_1 .
- () None of the weights will become exactly zero, only smaller as C increases

The data can be classified with zero training error and therefore also with high log-probability by looking at the value of x_2 alone, i.e. making $w_1 = 0$. Initially we might prefer to have a non-zero value for w_1 but it will go to zero rather quickly as we increase regularization. Note that we pay a regularization penalty for a non-zero value of w_1 and if it doesn't help classification why would we pay the penalty? The absolute value regularization ensures that w_1 will indeed go to exactly zero.

As C increases further, even w_2 will eventually become zero. We pay higher and higher cost for setting w_2 to a non-zero value. Eventually this cost overwhelms the gain from the log-probability of labels that we can achieve with a non-zero w_2 . Note that when $w_1 = w_2 = 0$, the log-probability of labels is a finite value $n \cdot \log(0.5)$.

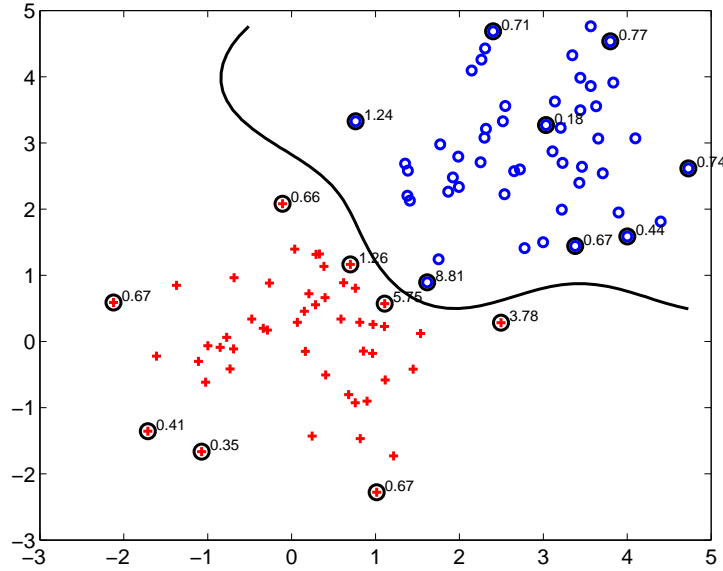


Figure 1: A 2-dim classification problem, the resulting SVM decision boundary with a radial basis kernel, as well as the support vectors (indicated by larger circles around them). The numbers next to the support vectors are the corresponding coefficients $\hat{\alpha}$.

Problem 3

Figure 1 illustrates a binary classification problem along with our solution using support vector machines (SVMs). We have used a radial basis kernel function given by

$$K(\mathbf{x}, \mathbf{x}') = \exp\{-\|\mathbf{x} - \mathbf{x}'\|^2/2\}$$

where $\|\cdot\|$ is a Euclidean distance and $\mathbf{x} = [x_1, x_2]^T$. The classification decision for any \mathbf{x} is made on the basis of the sign of

$$\hat{\mathbf{w}}^T \phi(\mathbf{x}) + \hat{w}_0 = \sum_{j \in \text{SV}} y_j \hat{\alpha}_j K(\mathbf{x}_j, \mathbf{x}) + \hat{w}_0 = f(\mathbf{x}; \hat{\alpha}, \hat{w}_0)$$

where $\hat{\mathbf{w}}$, \hat{w}_0 , $\hat{\alpha}_i$ are all coefficients estimated from the available data displayed in the figure and SV is the set of support vectors. $\phi(\mathbf{x})$ is the feature vector derived from \mathbf{x} corresponding to the radial basis kernel. In other words, $K(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}')$. While technically $\phi(\mathbf{x})$ is an infinite dimensional vector in this case, this fact plays no role in the questions below. You can assume and treat it as a finite dimensional vector if you like.

The support vectors we obtain for this classification problem (indicated with larger circles in the figure) seem a bit curious. Some of the support vectors appear to be far away from the decision boundary and yet be support vectors. Some of our questions below try to resolve this issue.

1. **(3 points)** What happens to our SVM predictions $f(\mathbf{x}; \hat{\alpha}, \hat{w}_0)$ with the radial basis kernel if we choose a test point \mathbf{x}_{far} far away from any of the training points \mathbf{x}_j (distances here measured in the space of the original points)?

The radial basis kernel $K(\mathbf{x}_{far}, \mathbf{x}_j)$ vanishes as the distance $\|\mathbf{x}_{far} - \mathbf{x}_j\|$ to the training point increases. The value of $f(\mathbf{x}; \hat{\alpha}, \hat{w}_0)$ therefore approaches \hat{w}_0 for any point \mathbf{x}_{far} sufficiently far from any of the training points.

2. **(3 points)** Let's assume for simplicity that $\hat{w}_0 = 0$. What equation do all the training points \mathbf{x}_j have to satisfy? Would \mathbf{x}_{far} satisfy the same equation?

If $\hat{w}_0 = 0$, then all the training points will satisfy

$$y_i \hat{\mathbf{w}}^T \phi(\mathbf{x}_i) - 1 \geq 0$$

since the problem is separable. \mathbf{x}_{far} cannot satisfy this equation regardless of the label associated with this point since $\hat{\mathbf{w}}^T \phi(\mathbf{x}_{far}) = f(\mathbf{x}_{far}; \hat{\alpha}, 0) \approx 0$.

3. **(4 points)** If we included \mathbf{x}_{far} in the training set, would it become a support vector? Briefly justify your answer.

\mathbf{x}_{far} would have to become a support vector. Our answers to the above questions indicate that this point could not satisfy the margin constraints without being included in the solution $f(\mathbf{x}; \hat{\alpha}, \hat{w}_0)$.

4. **(T/F – 2 points)** Leave-one-out cross-validation error is always small for support vector machines.

F

The claim is roughly the same as saying that the SVM always has a low generalization error – which is false. The question is admittedly a little ambiguous since you could have been thinking about a different notion of “small”.

Note that the number of support vectors is only partially related to the cross-validation error. We know that cross-validation error has to be smaller than the relative number of support vectors. However, we can have a large number of support vectors and yet very small cross-validation error. This happens when many of the support vectors are not “essential”.

5. **(T/F – 2 points)** The maximum margin decision boundaries that support vector machines construct have the lowest generalization error among all linear classifiers

F

The maximum margin hyperplane is often a reasonable choice but it is by no means optimal in all cases.

6. **(T/F – 2 points)** Any decision boundary that we get from a generative model with class-conditional Gaussian distributions could in principle be reproduced with an SVM and a polynomial kernel of degree less than or equal to three

T

A polynomial kernel of degree two suffices to represent any quadratic decision boundary such as the one from the generative model in question.

7. **(T/F – 2 points)** The decision boundary implied by a generative model (with parameterized class-conditional densities) can be optimal only if the class-conditional densities are correct for the problem at hand

F

The decision boundary may not depend on all aspects of the class-conditional densities. For example, in the trivial case where the class-conditional densities are the same for the two classes, the optimal decision boundary is based only on the prior class frequencies. We can easily reproduce this with any identical class-conditional densities.

Problem 4

Consider the following set of 3-dimensional points, sampled from two classes:

	x_1	x_2	x_3		x_1	x_2	x_3
labeled '1':	1,	1,	-1	labeled '0':	1,	1,	2
	0,	2,	-2		0,	2,	1
	0,	-1,	1		1,	-1,	-1
	0,	-2,	2		1,	-2,	-2

We have included 2-dimensional plots of pairs of features in the “Additional set of figures” section (figure 3).

1. **(4 points)** Explain briefly why features with higher mutual information with the label are likely to be more useful for classification task (in general, not necessarily in the given example).

A high value of $I(\text{feature}; \text{label})$ means that we can substantially reduce our uncertainty about the label by knowing the value of the feature. In other words, we have a good idea about the label if we know the value of the feature. For classification we want features that tell us most about the label.

2. **(3 points)** In the example above, which feature (x_1 , x_2 or x_3) has the highest mutual information with the class label, based on the training set?

x_1

Clearly, both x_2 and x_3 have zero mutual information with the label - their values do not appear to depend on the label. The value of x_1 does seem to provide some information about the label since, for example, class 0 has a higher chance of $x_1 = 1$ than class 1.

3. **(4 points)** Assume that the learning is done with quadratic logistic regression, where

x_2, x_3

$$P(y = 1 | \mathbf{x}, \mathbf{w}) = g(w_0 + w_1x_i + w_2x_j + w_3x_ix_j + w_4x_i^2 + w_5x_j^2)$$

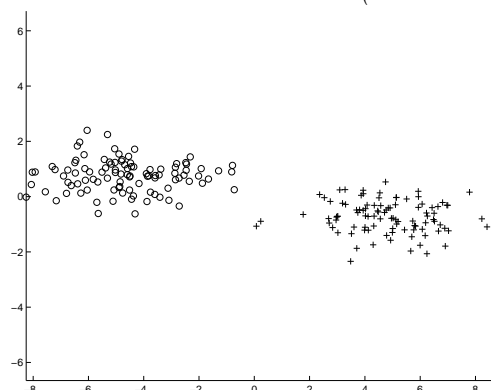
for some pair of features (x_i, x_j) . Based on the training set given above, which pair of features would result in the lowest training error for the logistic regression model?

One could refer to plots in Figure 3, or simply analyze the values of the features. The values of x_1, x_2 are the same for two pairs examples that belong to different classes: $(1, 1, -1)/(1, 1, 2)$ and $(0, 2, -2)/(0, 2, 1)$. We cannot classify all of these correctly no matter what kind of decision boundary we would have. The same is true for x_1, x_3 - consider, $(1, 1, -1)/(1, -1, -1)$ and $(0, -1, 1)/(0, 2, 1)$. However, including x_2, x_3 as features, all the training points appear distinct (see the plot). They can also be separated with a quadratic decision boundary. This is clear from the figure but you can also check that thresholding $x_2 \cdot x_3$ is sufficient for correct classification (set all the weights to zero except w_3).

4. **(T/F – 2 points)** From the point of view of classification it is always beneficial to remove features that have very high variance in the data

F

The feature with the highest variance can still have the highest mutual information with the label (as in the figure below).



5. (T/F – 2 points) A feature which has zero mutual information with the class label might be selected by a greedy selection method, if it happens to improve classifier's performance on the training set

T

The mutual information of a single feature and the label measures how good the feature is alone. Greedy selection picks features that are useful in conjunction with those already selected. It is possible that a feature which would be useless alone proves to be useful when combined with another.

Problem 5

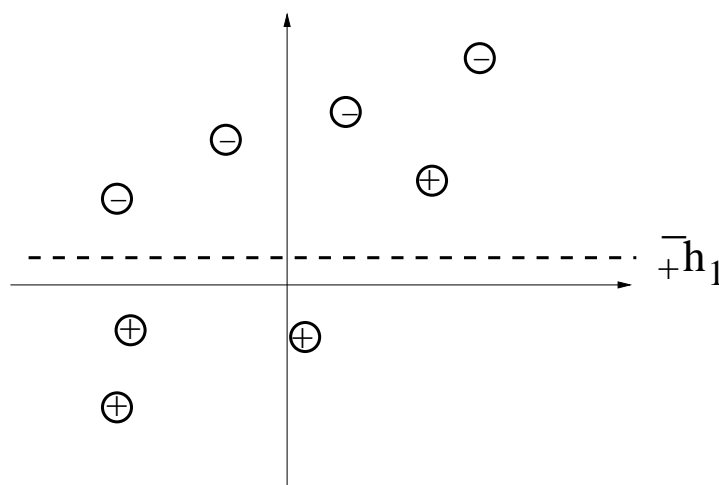


Figure 2: h_1 is chosen at the first iteration of boosting; what is the weight α_1 assigned to it?

1. **(3 points)** Figure 2 shows a dataset of 8 points, equally divided among the two classes (positive and negative). The figure also shows a particular choice of decision stump h_1 picked by AdaBoost in the first iteration. What is the weight α_1 that will be assigned to h_1 by AdaBoost? (Initial weights of all the data points are equal, or $1/8$.)

$\log_2 \sqrt{7}$

The weighted training error ϵ is $1/8$ - thus $\alpha = \frac{1}{2} \log_2 \frac{1-\epsilon}{\epsilon} = \frac{1}{2} \log_2 \frac{7/8}{1/8}$.

2. **(T/F – 2 points)** AdaBoost will eventually reach zero training error, regardless of the type of weak classifier it uses, provided enough weak classifiers have been combined.

F

Not if the data in the training set cannot be separated by a linear combination of the specific type of weak classifiers we are using.

3. **(T/F – 2 points)** The votes α_i assigned to the weak classifiers in boosting generally go down as the algorithm proceeds, because the weighted training error of the weak classifiers tends to go up

T

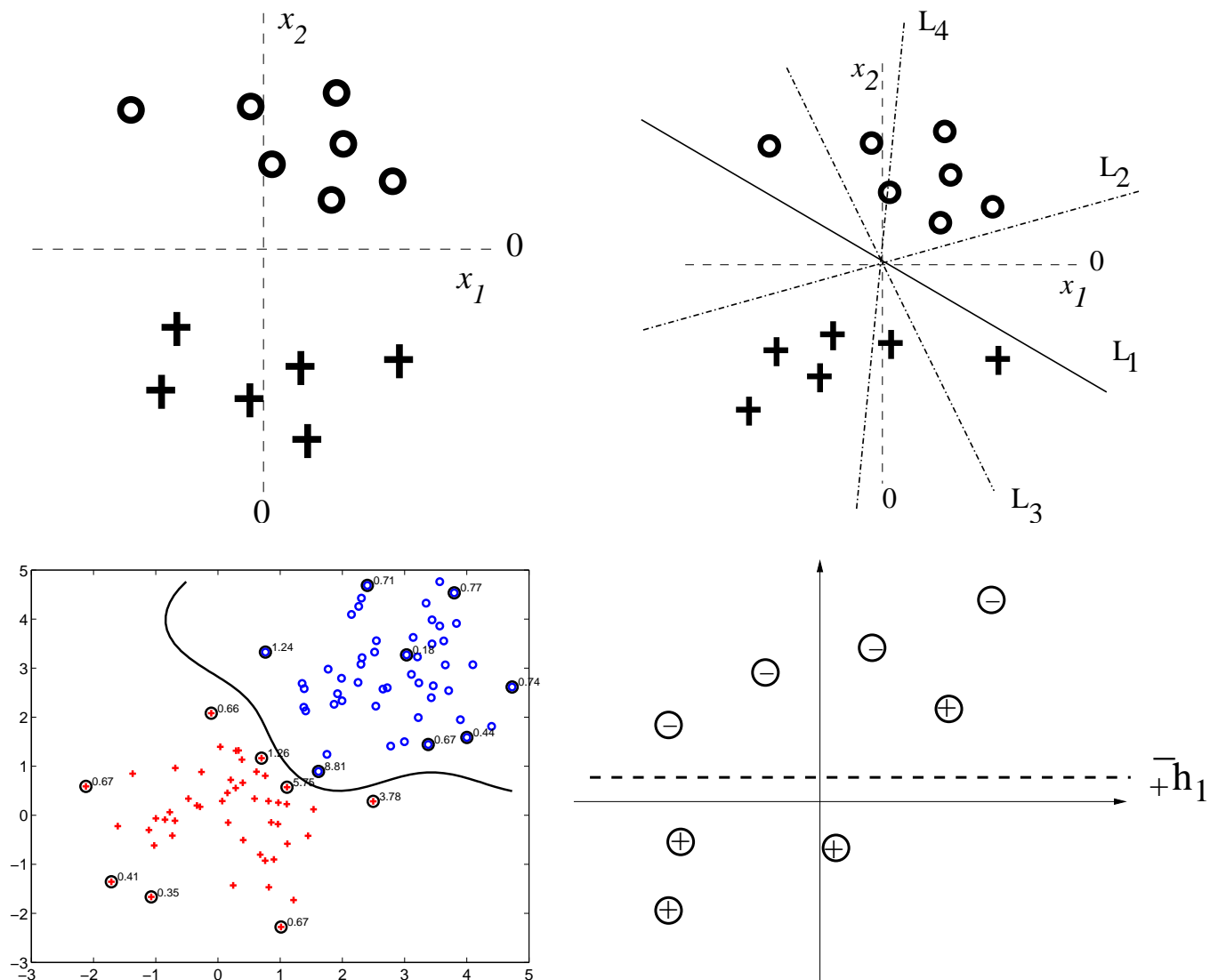
In the course of boosting iterations the weak classifiers are forced to try to classify more difficult examples. The weights will increase for examples that are repeatedly misclassified by the weak component classifiers. The weighted training error of the components therefore tends to go up and, as a result, their votes go down.

4. **(T/F – 2 points)** The votes α assigned to the classifiers assembled by AdaBoost are always non-negative

T

As defined in class, AdaBoost will choose classifiers with training error above $1/2$. This will ensure that $\log_2(1 - \epsilon/\epsilon)$, and therefore the vote, is positive. Note that if the classifier does worse than $1/2$ we can always “flip” the sign of its predictions and therefore get a classifier that does slightly better than $1/2$. The vote assigned to the “flipped” classifier would be non-negative.

Additional set of figures



there's more ...

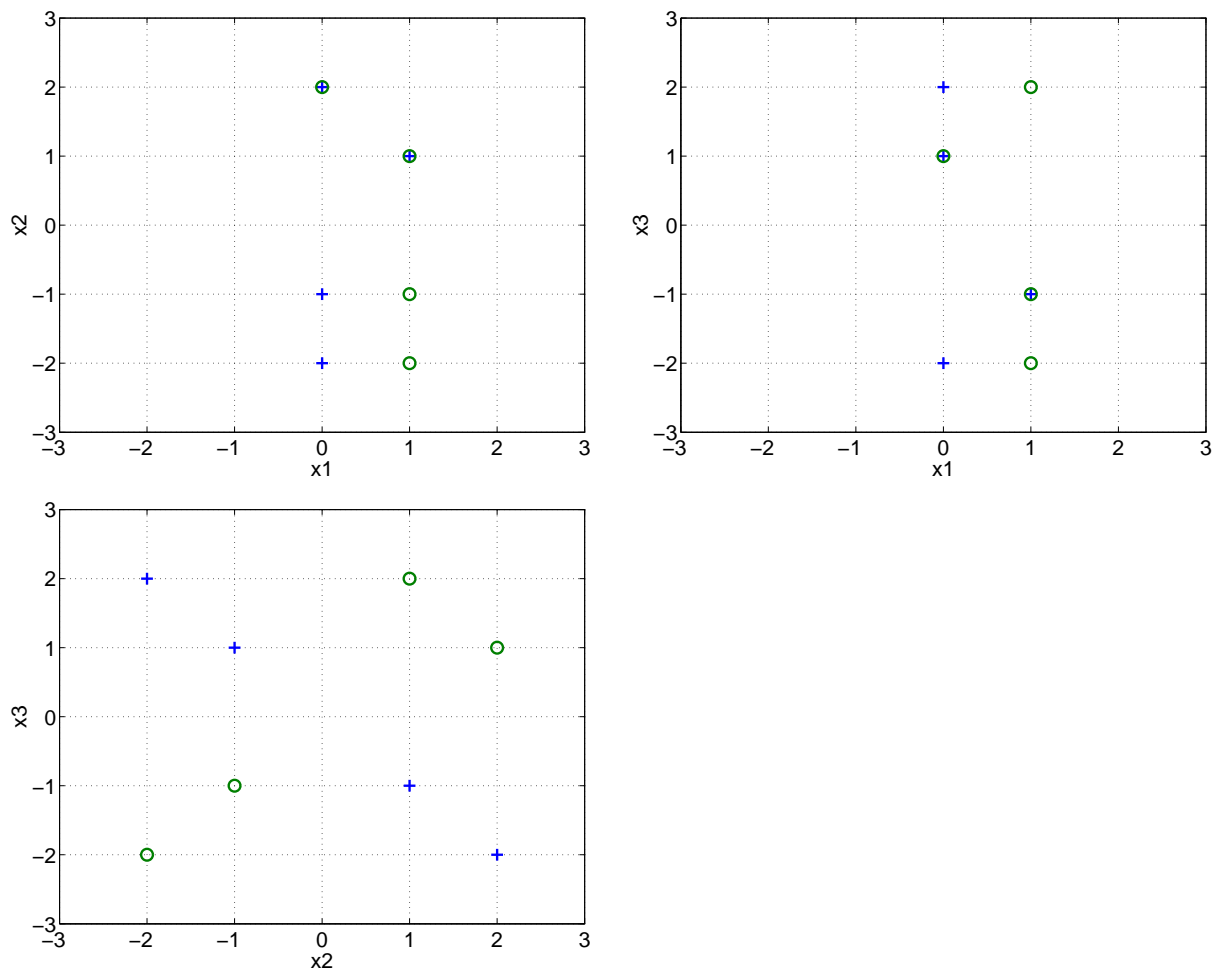


Figure 3: 2-dimensional plots of pairs of features for problem 4. Here '+' corresponds to class label '1' and 'o' to class label '0'.

6.867 Machine learning

Mid-term exam

October 8, 2003

(2 points) Your name and MIT ID:

J. J. Doe, MIT ID# 000000000

Problem 1

In this problem we use sequential active learning to estimate a linear model

$$y = w_1x + w_0 + \epsilon$$

where the input space (x values) are restricted to be within $[-1, 1]$. The noise term ϵ is assumed to be a zero mean Gaussian with an unknown variance σ^2 . Recall that our sequential active learning method selects input points with the highest variance in the predicted outputs. Figure 1 below illustrates what outputs would be returned for each query (the outputs are not available unless specifically queried).

We start the learning algorithm by querying outputs at two input points, $x = -1$ and $x = 1$, and let the sequential active learning algorithm select the remaining query points.

1. **(4 points)** Give the next two inputs that the sequential active learning method would pick. Explain why.

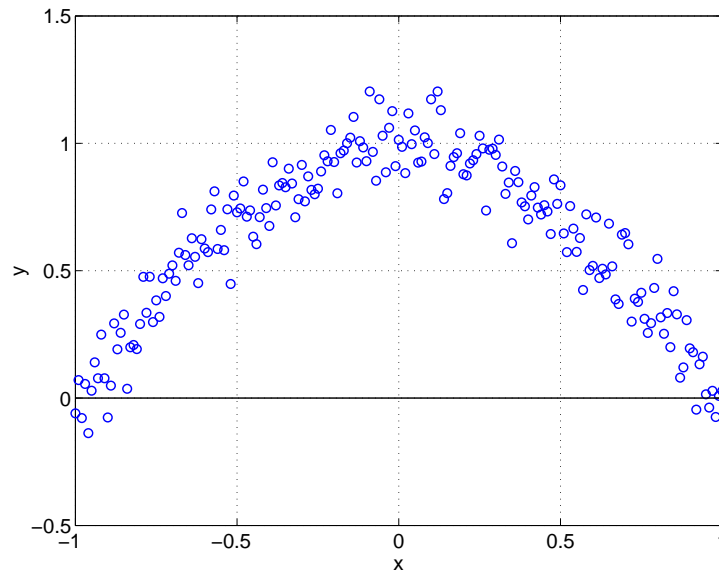


Figure 1: Samples from the underlying relation between the inputs x and outputs y . The outputs are not available to the learning algorithm unless specifically queried.

A linear function is constrained the most by the end points (the variance is always highest at the end points) and thus the next two points are -1 and 1 (order not determined).

Alternatively, you can look more closely at

$$\text{Var}\{\hat{y}(x)\} = \begin{bmatrix} 1 \\ x \end{bmatrix}^T (\mathbf{X}^T \mathbf{X})^{-1} \begin{bmatrix} 1 \\ x \end{bmatrix}$$

where, initially, $\mathbf{X} = [1, -1; 1, 1]$ (MATLAB notation). Since $(\mathbf{X}^T \mathbf{X})^{-1}$ is positive (semi-)definite regardless of the points included in \mathbf{X} , the variance is a convex-up parabola and takes its largest value at (one of) the end points.

2. **(4 points)** In the figure 1 above, draw (approximately) the linear relation between the inputs and outputs that the active learning method would find after a large number of iterations.

Since only the points -1 and 1 are queried, the linear function is determined by the outputs at these points. Hence the line in the figure.

3. **(6 points)** Would the result be any different if we started with query points $x = 0$

and $x = 1$ and let the sequential active learning algorithm select the remaining query points? Explain why or why not.

The same argument applies here and we would start querying at $x=-1$ and $x=1$ in an alternating fashion. Note that while the variance is a function of inputs only, the resulting linear function surely depends on the outputs as well. Initially, therefore, there will be a difference due to the single non-extremal query point $x=0$ but as more end points are selected, the difference will vanish

Problem 2

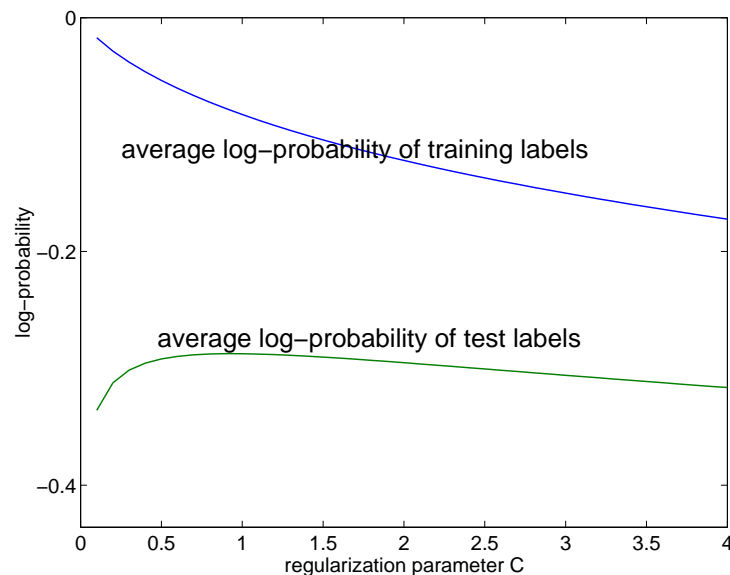


Figure 2: Log-probability of labels as a function of regularization parameter C

Here we use a logistic regression model to solve a classification problem. In Figure 2, we have plotted the mean log-probability of labels in the training and test sets after having trained the classifier with quadratic regularization penalty and different values of the regularization parameter C .

1. **(T/F – 2 points)** In training a logistic regression model by maximizing the likelihood of the labels given the inputs we have multiple locally optimal solutions.

F

The log-probability of labels given examples implied by the logistic regression model is a concave (convex down) function with respect to the weights. The (only) locally optimal solution is also globally optimal (if there are only a few examples, there might be a “flat” region with many equally optimal solutions, as in, e.g., question 3.4).

2. **(T/F – 2 points)** A stochastic gradient algorithm for training logistic regression models with a fixed learning rate will find the optimal setting of the weights exactly.

F

A fixed learning rate means that we are always taking a finite step towards improving the log-probability of any single training example in the update equation. Unless the examples are somehow “aligned”, we will continue jumping from side to side of the optimal solution, and will not be able to get arbitrarily close to it. The learning rate has to approach to zero in the course of the updates for the weights to converge.

3. **(T/F – 2 points)** The average log-probability of training labels as in Figure 2 can never increase as we increase C .

T

Stronger regularization means more constraints on the solution and thus the (average) log-probability of the training examples can only get worse.

4. **(4 points)** Explain why in Figure 2 the test log-probability of labels decreases for large values of C .

As C increases, we give more weight to constraining the predictor, and thus give less flexibility to fitting the training set. The increased regularization guarantees that the test performance gets closer to the training performance, but as we over-constrain our allowed predictors, we are not able to fit the training set at all, and although the test performance is now very close to the training performance, both are low.

5. **(T/F – 2 points)** The log-probability of labels in the test set would decrease for large values of C even if we had a large number of training examples.

T

The above argument still holds, but the value of C for which we will observe such a decrease will scale up with the number of examples.

6. **(T/F – 2 points)** Adding a quadratic regularization penalty for the parameters when estimating a logistic regression model ensures that some of the parameters (weights associated with the components of the input vectors) vanish.

F

A regularization penalty for feature selection must have non-zero derivative at zero. Otherwise, the regularization has no effect at zero, and weight will tend to be slightly non-zero, even when this does not improve the log-probabilities by much.

Problem 3

Consider a training set consisting of the following eight examples:

Examples labeled “0”	Examples labeled “1”
3,3,0	2,2,0
3,3,1	1,1,1
3,3,0	1,1,0
2,2,1	1,1,1

The questions below pertain to various feature selection methods that we could use with the logistic regression model.

1. **(2 points)** What is the mutual information between the third feature and the target label based on the training set?

0

The third feature has the same distribution conditioned on both labels, hence it is statistically independent of the target label and the mutual information between them is zero.

2. **(2 points)** Which feature(s) would a filter feature selection method choose? You can assume here that the mutual information criterion is evaluated between a single feature and the label.

1,2

Both of these features have high, and equal, mutual information with the target label.

3. **(2 points)** Which two feature(s) would a greedy wrapper process choose?

1,3
or 2,3

First, one of the first two features will be chosen, as this allows us to easily classify correctly six of the eight features. Once one of these features is chosen, the other one does not help us. However, the third feature can help us discriminate between $(2,2,1)$ and $(2,2,0)$ — including it will enable us to classify all examples correctly, and will thus be included.

4. **(4 points)** Which features would a regularization approach with a 1-norm penalty $\sum_{i=1}^3 |w_i|$ choose? Explain briefly.

Since the first two features are identical, shifting weight between them has no effect on the log-probabilities. The 1-norm penalty is oblivious to shifting weight in this way. Thus, it will not favor setting the weight of one of the first two features to zero, and such a setting will be one of many equally optimal solutions. Most optimal solutions would thus include both features, but depending on how we optimize the objective function, we might choose the rare solutions in which one of the weights is set to zero.

Including the third features improve the average log-probabilities, but it also increases the penalty. Whether or not it will be included depends on the regularization parameter C that controls the relative importance of the average log-probabilities and the regularization penalty. To check what values of C will allow inclusion of the third feature, we can check the derivative of the objective function at $w_3 = 0$: The derivative of the log probability of the sample $(2,2,1)$ will be $g'(0)x_3 = \frac{1}{2}$. The derivatives for samples with $x_3 = 0$ will be zero, and the derivatives for the other four samples will cancel each other. Thus the total derivative of the log-probabilities term will be $\frac{1}{2}$. This will be more significant than the regularization only if $C < \frac{1}{2}$.

Problem 4

1. **(6 points)** Figure 3 shows the first decision stump that the AdaBoost algorithm finds (starting with the uniform weights over the training examples). We claim that the weights associated with the training examples after including this decision stump will be $[1/8, 1/8, 1/8, 5/8]$ (the weights here are enumerated as in the figure). Are these weights correct, why or why not?

Do not provide an explicit calculation of the weights.

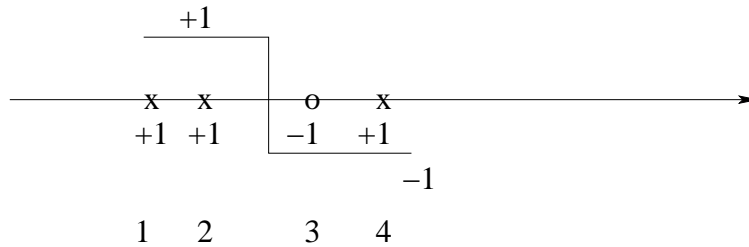


Figure 3: The first decision stump that the boosting algorithm finds.

The boosting algorithm adjusts the weights so that the current hypothesis is at chance level relative to the new weights. Since there is only one miss-classified example, it has to receive weight = $1/2$. So the weights are not correct.

2. **(T/F – 2 points)** The votes that AdaBoost algorithm assigns to the component classifiers are optimal in the sense that they ensure larger “margins” in the training set (higher majority predictions) than any other setting of the votes.

F

The votes in the boosting algorithm are optimized sequentially and never “reoptimized” after all the hypotheses have been generated. These votes cannot therefore be optimal in the sense of achieving the largest majority predictions for the training examples.

3. **(T/F – 2 points)** In the boosting iterations, the training error of each new decision stump and the training error of the combined classifier vary roughly in concert

F

While the training error of the combined classifier typically decreases as a function of boosting iterations, the error of the individual decision stumps typically increases since the example weights become concentrated at the most difficult examples.

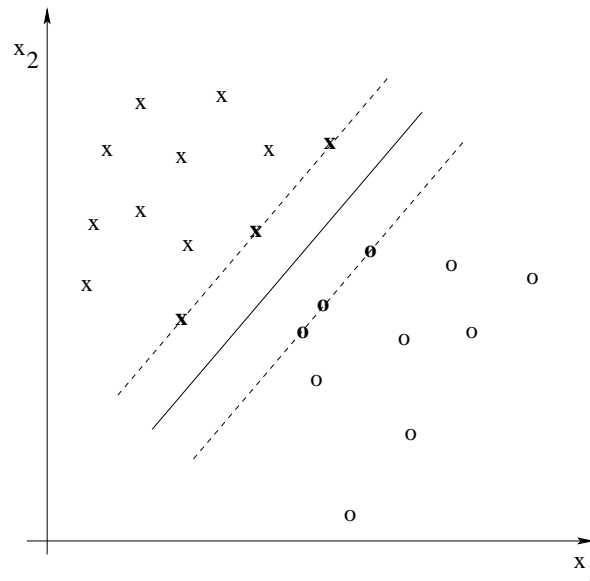


Figure 4: Training set, maximum margin linear separator, and the support vectors (in bold).

Problem 5

1. **(4 points)** What is the leave-one-out cross-validation error estimate for maximum margin separation in figure 4? (we are asking for a number)

0

Based on the figure we can see that removing any single point would not change the resulting maximum margin separator. Since all the points are initially classified correctly, the leave-one-out error is zero.

2. **(T/F – 2 points)** We would expect the support vectors to remain the same in general as we move from a linear kernel to higher order polynomial kernels.

F

There are no guarantees that the support vectors remain the same. The feature vectors corresponding to polynomial kernels are non-linear functions of the original input vectors and thus the support points for maximum margin separation in the feature space can be quite different.

3. **(T/F – 2 points)** Structural risk minimization is guaranteed to find the model (among those considered) with the lowest expected loss

F

We are guaranteed to find only the model with the lowest upper bound on the expected loss.

4. **(6 points)** What is the VC-dimension of a mixture of two Gaussians model in the plane with equal covariance matrices? Why?

A mixture of two Gaussians with equal covariance matrices has a linear decision boundary. Linear separators in the plane have VC-dim exactly 3.

Problem 6

Using a set of 100 labeled training examples (two classes), we train the following models:

GaussI A Gaussian mixture model (one Gaussian per class), where the covariance matrices are both set to I (identity matrix).

GaussX A Gaussian mixture model (one Gaussian per class) without any restrictions on the covariance matrices.

LinLog A logistic regression model with linear features.

QuadLog A logistic regression model, using all linear and quadratic features.

1. **(6 points)** After training, we measure for each model *the average log probability of labels given examples in the training set*. Specify all the equalities or inequalities that must *always* hold between the models relative to this performance measure. We are looking for statements like “model 1 \leq model 2” or “model 1 = model 2”. If no such statement holds, write “none”.

GaussI \leq **LinLog** (both have logistic posteriors, and **LinLog** is the logistic model maximizing the average log probabilities)

GaussX \leq **QuadLog** (both have logistic posteriors with quadratic features, and **QuadLog** is the model of this class maximizing the average log probabilities)

LinLog \leq **QuadLog** (logistic regression models with linear features are a subclass of logistic regression models with quadratic functions— the maximum from the superclass is at least as high as the maximum from the subclass)

GaussI \leq **QuadLog** (follows from above inequalities)

(**GaussX** will have higher average log joint probabilities of examples and labels, then will **GaussI**. But have higher average log joint probabilities does not necessarily translate to higher average log conditional probabilities)

2. (4 points) Which equalities and inequalities must *always* hold if we instead use the mean classification error in the training set as the performance measure? Again use the format “model 1 \leq model 2” or “model 1 = model 2”. Write “none” if no such statement holds.

None. Having higher average log conditional probabilities, or average log joint probabilities, does not necessarily translate to higher or lower classification error. Counterexamples can be constructed for all pairs in both directions.

Although there is no inequalities which is always correct, it is commonly the case that **GaussX** \leq **GaussI** and that **QuadLog** \leq **LinLog**. Partial credit of up to two points was awarded for these inequalities.