10-701 Midterm Exam, Spring 2007

1. Personal info:
   - Name:
   - Andrew account:
   - E-mail address:
2. There should be 16 numbered pages in this exam (including this cover sheet).
3. You can use any material you brought: any book, class notes, your print outs of class materials that are on the class website, including my annotated slides and relevant readings, and Andrew Moore's tutorials. You cannot use materials brought by other students. Calculators are allowed, but no laptops, PDAs, phones or Internet access.
4. If you need more room to work out your answer to a question, use the back of the page and clearly mark on the front of the page if we are to look at what's on the back.
5. Work efficiently. Some questions are easier, some more difficult. Be sure to give yourself time to answer all of the easy ones, and avoid getting bogged down in the more difficult ones before you have answered the easier ones.
6. Note there are extra-credit sub-questions. The grade curve will be made without considering students' extra credit points. The extra credit will then be used to try to bump your grade up without affecting anyone else's grade.
7. You have 80 minutes.
8. Good luck!

Question	Topic	Max. score	Score
1	Short questions	21 + 0.911 extra	
2	SVM and slacks	16	
3	GNB	8	
4	Feature Selection	10	
5	Irrelevant Features	14 + 3 extra	
6	Neural Nets	16 + 5 extra	
7	Learning theory	15	

1 [21 Points] Short Questions

The following short questions should be answered with at most two sentences, and/or a picture. For the **(true/false)** questions, answer true or false. If you answer true, provide a short justification, if false explain why or provide a small counterexample.

1. [1 point] **true/false** A classifier trained on less training data is less likely to overfit.

2. [1 point] **true/false** Given m data points, the training error converges to the true error as $m \to \infty$.

3. [1 point] **true/false** The maximum likelihood model parameters (α) can be learned using linear regression for the model: $y_i = \alpha_1 x_1 x_2^3 + \epsilon_i$ where $\epsilon_i \sim N(0, \sigma^2)$ iid noise.

4. [2 points] **true/false** The maximum likelihood model parameters (α) can be learned using linear regression for the model: $y_i = x_1^{\alpha_1} e^{\alpha_2} + \epsilon_i$ where $\epsilon_i \sim N(0, \sigma^2)$ iid noise.

5. [2 points] **true/false** The maximum likelihood model parameters (α) can be learned using linear regression for the model: $y_i = \log(x_1^{\alpha_1} e^{\alpha_2}) + \epsilon_i$ where $\epsilon_i \sim N(0, \sigma^2)$ iid noise.

6. [2 points] **true/false** In AdaBoost weights of the misclassified examples go up by the same multiplicative factor.

7. [2 points] **true/false** In AdaBoost, weighted training error ϵ_t of the t^{th} weak classifier on training data with weights D_t tends to increase as a function of t.

8. [2 points] **true/false** AdaBoost will eventually give zero training error regardless of the type of weak classifier it uses, provided enough iterations are performed.

9. [2 points] Consider a point that is correctly classified and distant from the decision boundary. Why would SVM's decision boundary be unaffected by this point, but the one learned by logistic regression be affected?

10. [2 points] Why does the kernel trick allow us to solve SVMs with high dimensional feature spaces, without significantly increasing the running time?

11. [2 points] Consider a learning problem with 2D features. How are the decision tree and 1-nearest neighbor decision boundaries related?

12. [2 points] You are a reviewer for the International Mega-Conference on Algorithms for Radical Learning of Outrageous Stuff, and you read papers with the following experimental setups. Would you accept or reject each paper? Provide a one sentence justification. (This conference has short reviews.)

    - **accept/reject** "My algorithm is better than yours. Look at the training error rates!"

    - **accept/reject** "My algorithm is better than yours. Look at the test error rates! (Footnote: reported results for $\lambda = 1.789489345672120002$.)"

    - **accept/reject** "My algorithm is better than yours. Look at the test error rates! (Footnote: reported results for best value of λ.)"

    - **accept/reject** "My algorithm is better than yours. Look at the test error rates! (Footnote: reported results for best value of λ, chosen with 10-fold cross validation.)"

13. [Extra credit: 0.911 points] You have designed the ultimate learning algorithm that uses physical and metaphysical knowledge to learn and generalize beyond the quantum P-NP barrier. You are now given the following test example:

What label will your algorithm output?

(a) Watch a cartoon.

(b) Call the anti-terrorism squad.

(c) Support the Boston Red Sox.

(d) All labels have equal probability.

2 [16 Points] SVMs and the slack penalty C

The goal of this problem is to correctly classify test data points, given a training data set. You have been warned, however, that the training data comes from sensors which can be error-prone, so you should avoid trusting any specific point too much.

For this problem, assume that we are training an SVM with a **quadratic kernel**– that is, our kernel function is a polynomial kernel of degree 2. You are given the data set presented in Figure 1. The slack penalty C will determine the location of the separating hyperplane. Please answer the following questions *qualitatively*. Give a one sentence answer/justification for each and draw your solution in the appropriate part of the Figure at the end of the problem.

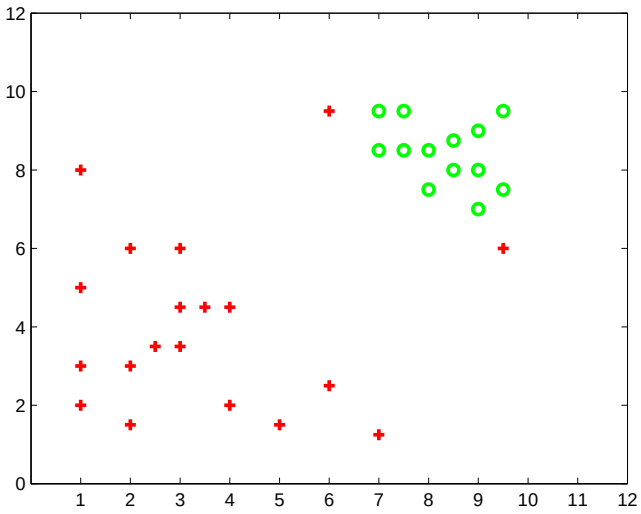


Figure 1: Dataset for SVM slack penalty selection task in Question 2.

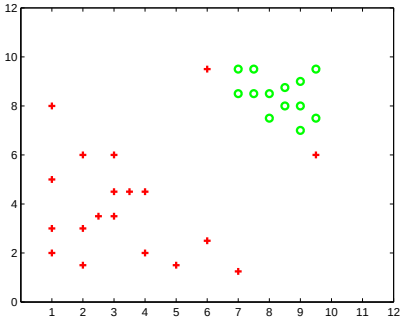
1. [4 points] Where would the decision boundary be for very large values of C (i.e., $C \to \infty$)? (remember that we are using an SVM with a quadratic kernel.) Draw on the figure below. Justify your answer.

2. [4 points] For $C \approx 0$, indicate in the figure below, where you would expect the decision boundary to be? Justify your answer.

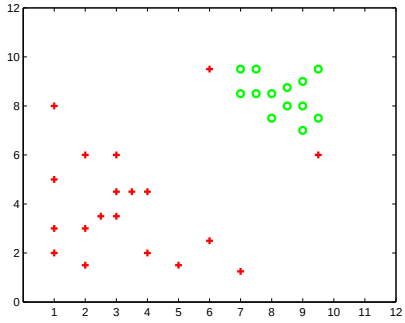
3. [2 points] Which of the two cases above would you expect to work better in the classification task? Why?

4. [3 points] Draw a data point which will not change the decision boundary learned for very large values of C. Justify your answer.

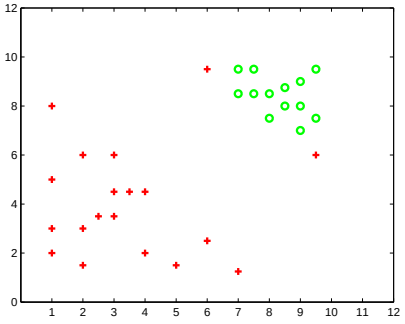
5. [3 points] Draw a data point which will significantly change the decision boundary learned for very large values of C. Justify your answer.



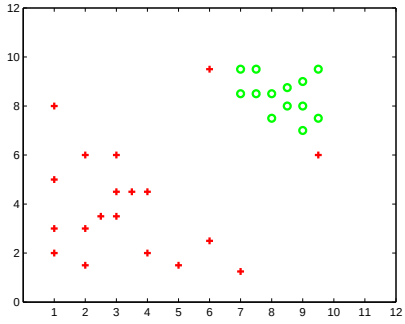
(a) Part 1



(b) Part 2



(c) Part 4



(d) Part 5

Figure 2: Draw your solutions for Problem 2 here.

3 [10 points] Feature selection with boosting

Consider a text classification task, such that the document X can be expressed as a binary feature vector of the words. More formally $X = [X_1, X_2, X_3, ..., X_m]$, where $X_j = 1$ if word j is present in document X, and zero otherwise. Consider using the AdaBoost algorithm with a simple weak learner, namely

$$\begin{aligned} h(X;\theta) &= yX_j \\ \theta &= \{j, y\}\ j \text{ is the word selector ; } y \text{ is the associated class} \\ y &\in \{-1, 1\} \end{aligned}$$

More intuitively, each weak learner is a word associated with a class label. For example if we had a word `football`, and classes {`sports`,`non-sports`}, then we will have two weak learners from this word, namely

- *Predict* **sports** *if document has word* **football**
- *Predict* **non-sports** *if document has word* **football**.

1. [2 points] How many weak learners are there ?

2. This boosting algorithm can be used for feature selection. We run the algorithm and select the features in the *order in which they were identified* by the algorithm.

   (a) [4 points] Can this boosting algorithm select the same weak classifier more than once? Explain.

   (b) [4 points] Consider ranking the features based on their individual mutual information with the class variable y, i.e. $\hat{I}(y; X_j)$. Will this ranking be more informative than the ranking returned by AdaBoost ? Explain.

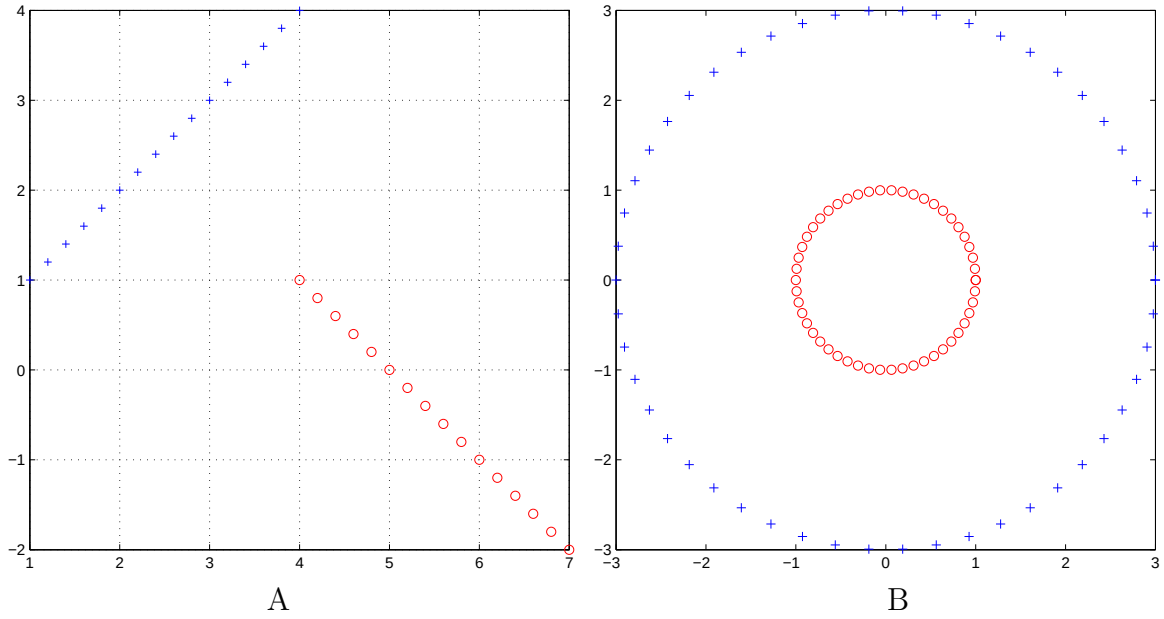


Figure 3: A. **toydata1** in Question 4, B. **toydata2** in Question 4

4 [8 points] Gaussian Naive Bayes classifier

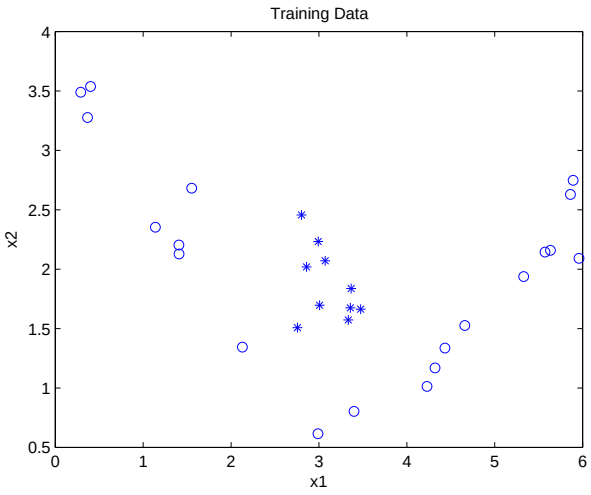
Consider the datasets **toydata1** in figure **3(A)** and **toydata2** in figure **3(B)**.

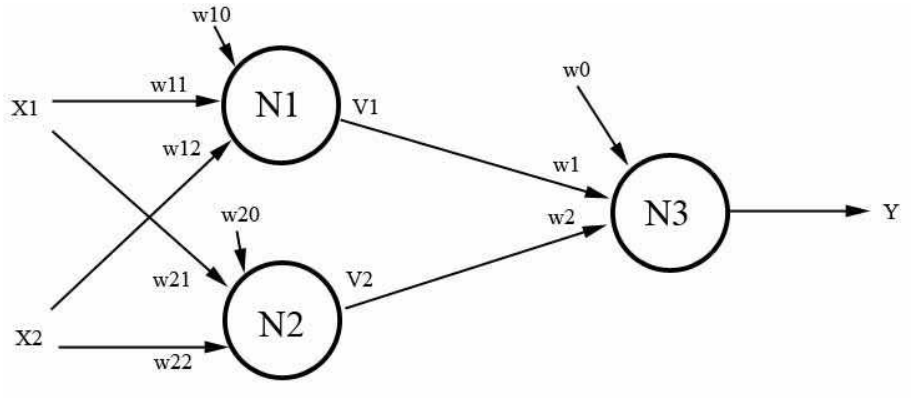
- In each of these datasets there are two classes, '+' and 'o'.
- Each class has the same number of points.
- Each data point has two real valued features, the X and Y coordinates.

For each of these datasets, draw the decision boundary that a Gaussian Naive Bayes classifier will learn.

5 [16 Points] Neural Networks

Consider the following classification training data (where "*" = true or 1 and "O" = false or 0) and neural network model that uses the **sigmoid** response function $\left(g(t) = \frac{1}{1+e^{-t}}\right)$.



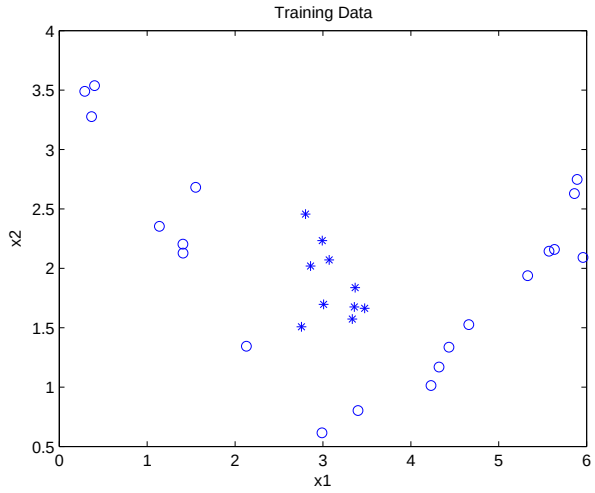


5.1 Weight choice

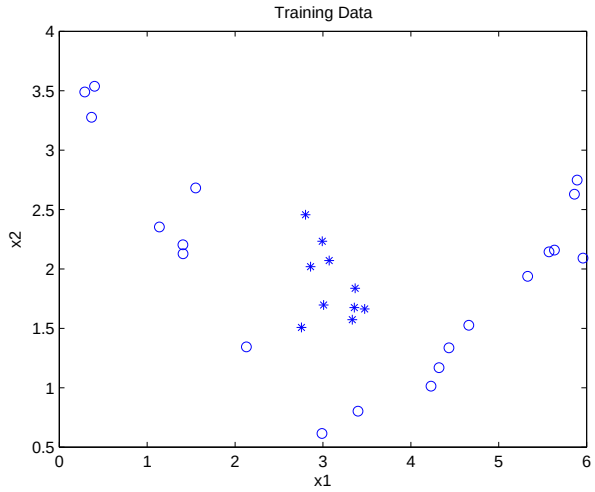
[8 points] We would like to set the weights (w) of the neural network so that it is capable of correctly classifying this dataset. Please plot the decision boundaries for N_1 and N_2 (e.g., for neuron N_1, the line where $w10 + w11 * X1 + w12 * X2 = 0$) on the first two graphs. In the third graph, which has axes V_2 and V_1, plot $\{V_1(x_1, x_2), V_2(x_1, x_2)\}$ for a few of the training points and provide a decision boundary so that the neural net will correctly classify the training data.

All graphs are on the following page!

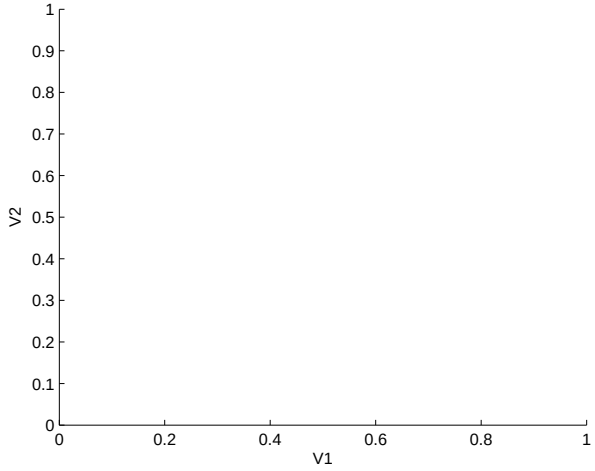
N1 (2 points)



N2 (2 points)



N3 (4 points)



5.2 Regularized Neural Networks

[8 points]

One method for preventing the neural networks' weights from overfitting is to add regularization terms. You will now derive the update rules for the regularized neural network. **Note:** $Y = out(x)$

Recall that the non-regularized gradient descent update rule for w_1 is:

$$w_1^{t+1} \leftarrow w_1^t + \eta \sum_j \left[\left(y^{(j)} - out(x^{(j)}) \right) out(x^{(j)})(1 - out(x^{(j)})) * V_1(x^{(j)}) \right] \tag{1}$$

[4 points] Derive the update rule for w_1 in the regularized neural net loss function which penalizes based on the square of each weight. Use λ to denote the magic regularization parameter.

[4 points] Now, re-express the regularized update rule so that the only difference between the regularized setting and the unregularized setting above is that the old weight w_1^t is scaled by some constant. Explain how this scaling prevents overfitting.

5.3 Neural Net Simplification [Extra Credit (5 points)]

Please provide a feed-forward neural network with a smaller architecture (i.e., fewer neurons and weights) that is able to correctly predict the entire training set. Justify your answer.

6 [14 Points] The Effect of Irrelevant Features

1. (a) [3 points] Provide a 2D dataset where 1-nearest neighbor (1-NN) has lower leave-one-out cross validation error (LOO error) than SVMs.

   (b) [3 points] Provide a 2D dataset where 1-NN has higher LOO error than SVMs.

2. [8 points] You will now generate a dataset to illustrate SVMs' robustness to irrelevant features. In particular, create a 2D dataset with features X_1 and X_2, here X_2 will be the irrelevant feature, such that:

   - If you only use X_1, 1-NN will have lower LOO error than SVMs,
   - but if you use both X_1 and X_2, the SVM LOO error will remain the same, but LOO error for 1-NN will increase significantly.

   You will receive extra credit if the 1-NN LOO error before adding the irrelevant feature is zero, but the error becomes 100% after adding the feature.

3. [Extra Credit (3 points)] SVMs tend to be robust to irrelevant features. Suppose we run SVMs with features $X_1, \ldots, X_n$, and then add a irrelevant feature X_{n+1} that cannot help increase the margin. How will SVMs automatically ignore this feature? Justify your answer formally.

7 [15 points] Learning Theory

Consider binary data-points X in n dimensions, with binary labels Y, i.e. $X \in \{0,1\}^n$; $Y \in \{0,1\}$. We wish to learn a mapping $X \rightarrow Y$ using a few different hypothesis classes, but are concerned about the tradeoff between the expressivity of our hypothesis space and the number of training examples required to learn the true mapping probably approximately correctly.

1. Consider the following hypothesis class H: decision stumps that choose a value for Y based on the value of one of the attributes of X. For example, there are two hypotheses in H that involve feature i:

$$h_i(X) = \begin{cases} 1 & \text{if } X_i = 1 \\ 0 & \text{otherwise} \end{cases} \qquad h_{\neg i}(X) = \begin{cases} 0 & \text{if } X_i = 1 \\ 1 & \text{otherwise;} \end{cases}$$

   (a) [3 points] What is the size of this hypothesis class?

   (b) [3 points] For given ϵ, δ how many training examples are needed to yield a decision stump that satisfies the Haussler-PAC bound?

2. Now let us define another hypothesis class H', where each hypothesis is a majority over a set of simple decision stumps. Specifically, for each feature i, we either use h_i or $h_{\neg i}$, and the output is the result of a majority vote (in the case of a tie, we predict 1). For example, if we have 5 features, and we choose the stumps $\{h_{\neg 1}, h_2, h_3, h_4, h_{\neg 5}\}$, then the resulting hypothesis is:

$$h'(X) = \begin{cases} 1 & \text{if } h_{\neg 1}(X) + h_2(X) + h_3(X) + h_4(X) + h_{\neg 5}(X) \geq \frac{5}{2} \\ 0 & \text{otherwise} \end{cases}$$

   (a) [4 points] What is the size of this hypothesis class?

   (b) [2 points] For given ϵ, δ how many training examples are needed to yield a hypothesis that satisfies the Haussler-PAC bound?

3. [3 points] What can we say about the amount of extra samples necessary to learn this voting classifier? Is this a concern?
   Briefly explain the tradeoff between the expressive power of the hypothesis space and the number of training samples required for these two classifier.