10-701 Midterm Exam, Spring 2005

1. Write your name and your email address below.

   Name:

   Email address:

2. There should be 15 numbered pages in this exam (including this cover sheet).
3. Write your name at the top of EVERY page in the exam.
4. You may use any and all books, papers, and notes that you brought to the exam, but not materials brought by nearby students. No laptops, PDAs, or Internet access.
5. 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.
6. 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.
7. Note there is one extra-credit question. 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.
8. You have 80 minutes.
9. Good luck!

Question	Number of points	Score
1. Big Picture	10	
2. Short Questions	15	
3. Learning Algorithms	16	
4. Decision Trees	16	
5. Loss Fns. and SVMs	23	
6. Learning Theory	20	
Total	100	
Extra credit		
7. Bias-Variance Trade-off	18	

1 [10 points] Big Picture

Following the example given, add 10 edges to Figure 1 relating the pair of algorithms. Each edge should be labeled with one characteristic the methods share, and one difference. These labels should be short and address basic concepts, such as types of learning problems, loss functions, and hypothesis spaces.

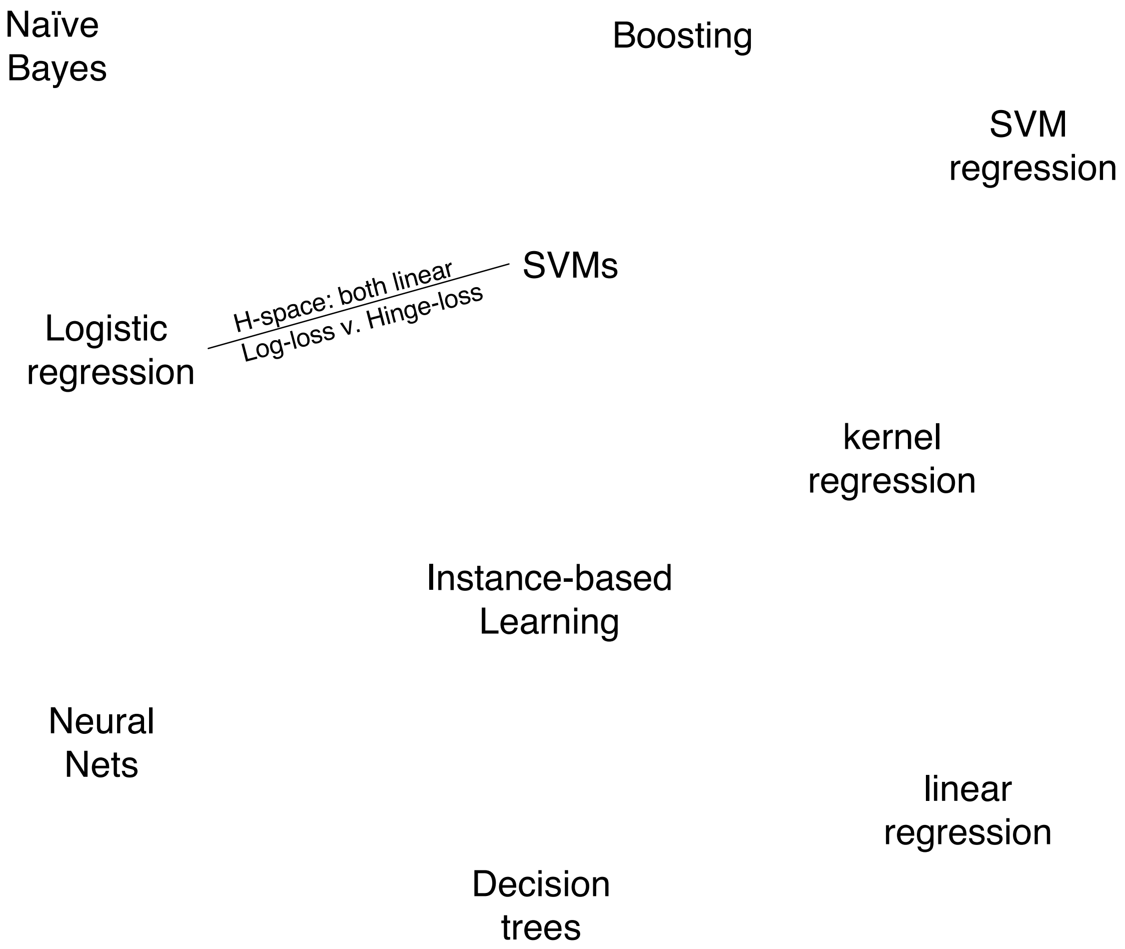


Figure 1: Big picture.

One solution is shown below, there are many others.

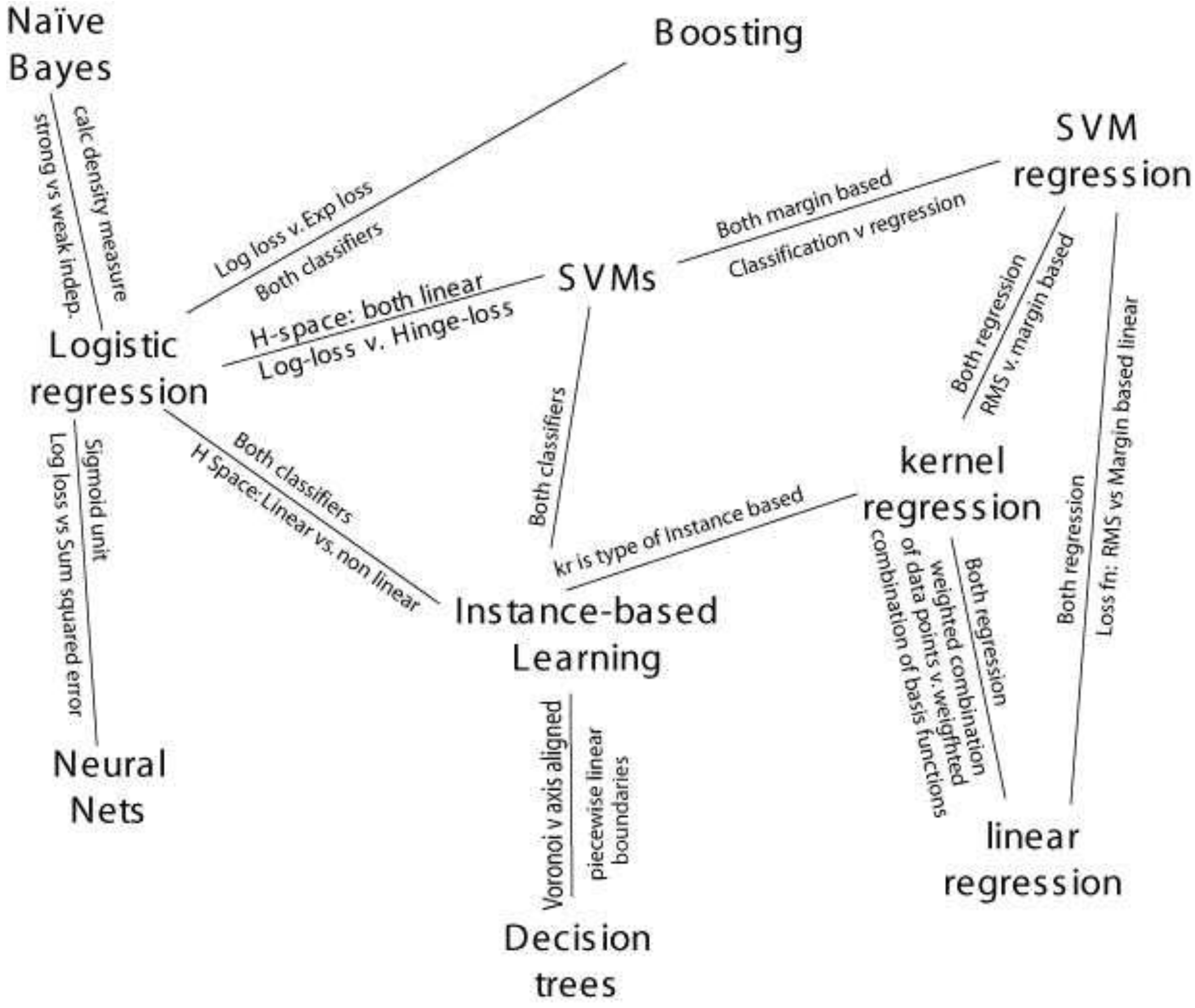


Figure 2: Big picture solutions.

2 [15 points] Short Questions

(a) [3 points] Briefly describe the difference between a *maximum likelihood* hypothesis and a *maximum a posteriori* hypothesis.

Solutions:
ML: maximize the data likelihood given the model, i.e., $\arg\max_W P(Data|W)$
MAP: $\arg\max_W P(W|Data)$

(b) [4 points] Consider a naive Bayes classifier with 3 boolean input variables, X_1, X_2 and X_3, and one boolean output, Y.

- How many parameters must be estimated to train such a naive Bayes classifier? (you need not list them unless you wish to, just give the total)

  Solutions:
  For a naive Bayes classifier, we need to estimate P(Y=1), $P(X1 = 1|y = 0), P(X2 = 1|y = 0)$, $P(X3 = 1|y = 0)$, $P(X1 = 1|y = 1), P(X2 = 1|y = 1), P(X3 = 1|y = 1)$. *Other probabilities can be obtained with the constraint that the probabilities sum up to 1.*
  So we need to estimate 7 parameters.

- How many parameters would have to be estimated to learn the above classifier if we do *not* make the naive Bayes conditional independence assumption?

  Solutions:
  Without the conditional independence assumption, we still need to estimate P(Y=1). For Y=1, we need to know all the enumerations of (X1,X2,X3), i.e., 2^3 *of possible (X1,X2,X3). Consider the constraint that the probabilities sum up to 1, we need to estimate* $2^3 - 1 = 7$ *parameters for Y=1.*
  Therefore the total number of parameters is $1 + 2(2^3 - 1) = 15$.

[8 points] True or False? If true, explain why in *at most two sentences.* If false, explain why or give a brief counterexample in *at most two sentences.*

- **(True or False?)** The error of a hypothesis measured over its training set provides a pessimistically biased estimate of the true error of the hypothesis.

  Solutions:
  False. The training error is optimisticly biased since it's biased while usually smaller than the true error.

- **(True or False?)** If you are given m data points, and use half for training and half for testing, the difference between training error and test error decreases as m increases.

  Solutions:
  True. As we have more and more data, training error increases and testing error decreases. And they all converge to the true error.

- **(True or False?)** Overfitting is more likely when the set of training data is small

  Solutions:
  True. With small training dataset, it's easier to find a hypothesis to fit the training data exactly,i.e., overfit.

- **(True or False?)** Overfitting is more likely when the hypothesis space is small

  Solutions:
  False. We can see this from the bias-variance trade-off. When hypothesis space is small, it's more biased with less variance. So with a small hypothesis space, it's less likely to find a hypothesis to fit the data very well,i.e., overfit.

3 [16 points] Learning Algorithms

Consider learning a target function of the form $f : \Re^2 \rightarrow \{A, B, C\}$ that is, a function with 3 discrete values defined over the 2-dimensional plane. Consider the following learning algorithms:

- Decision trees
- Logistic regression
- Support Vector Machine
- 1-nearest neighbor

Note each of these algorithms can be used to learn our target function f, though doing so might require a common extension (e.g., in the case of decision trees, we need to utilize the usual method for handling real-valued input attributes).

For each of these algorithms,

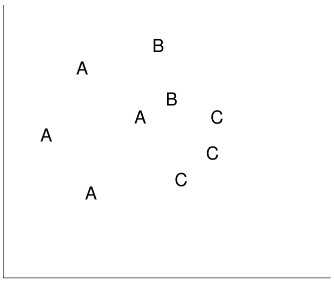
A. Describe any assumptions you are making about the variant of the algorithm you would use

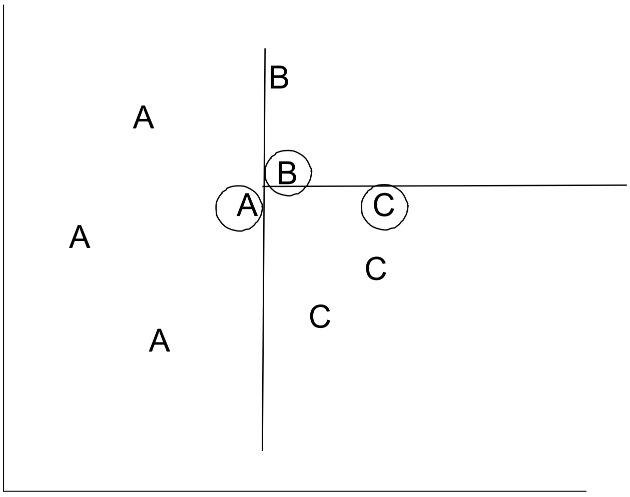
B. Draw in the decision surface that would be learned given this training data (and describing any ambiguities in your decision surface)

C. Circle any examples that would be misclassified in a leave-one-out evaluation of this algorithm with this data. That is, if you were to repeatedly train on n-1 of these examples, and use the learned classifier to label the left out example, will it be misclassified?

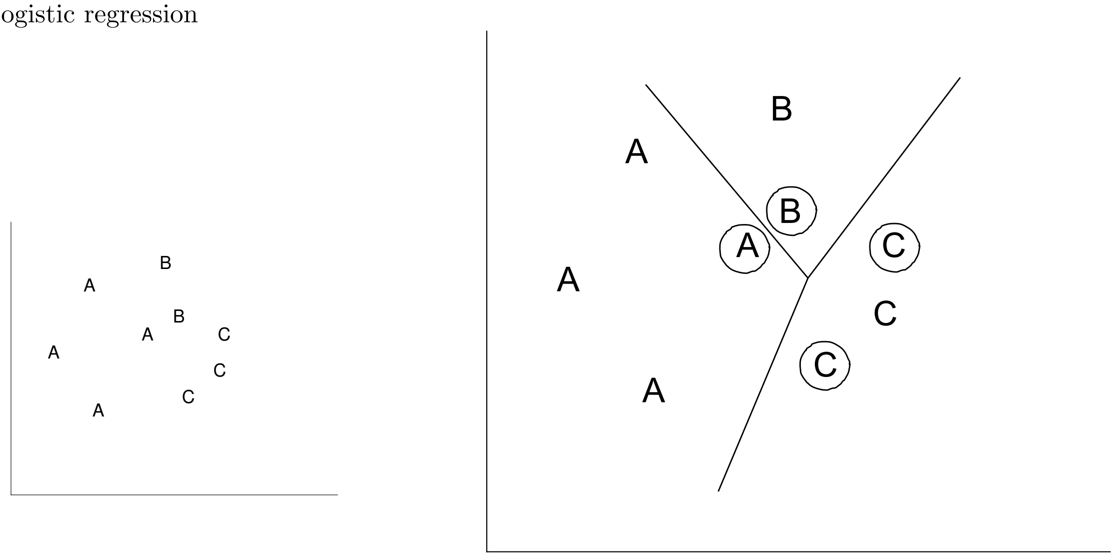
Solutions:
the assuptions are as follows:
Decision trees: Handle real valued attributes by discretizing;
Logistic regression: Handle non-binary classification;
SVM: Use one against all approach and a linear kernel;
1-NN: x-axis features and y-axis features are non-weighted.
Please see the figures on the right for decision surface and misclassified examples by leave-one-out evaluation.

Decision trees

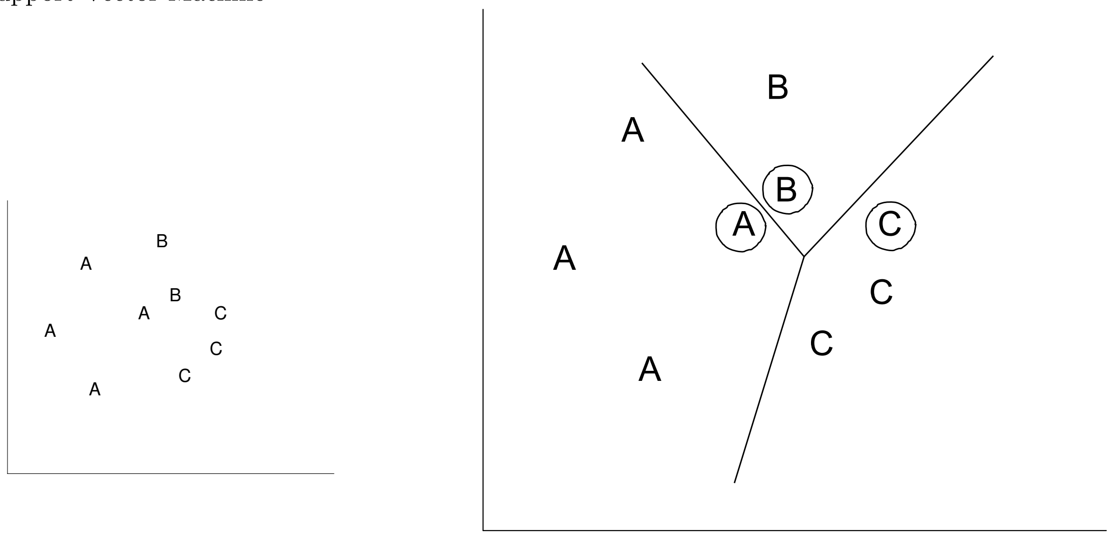




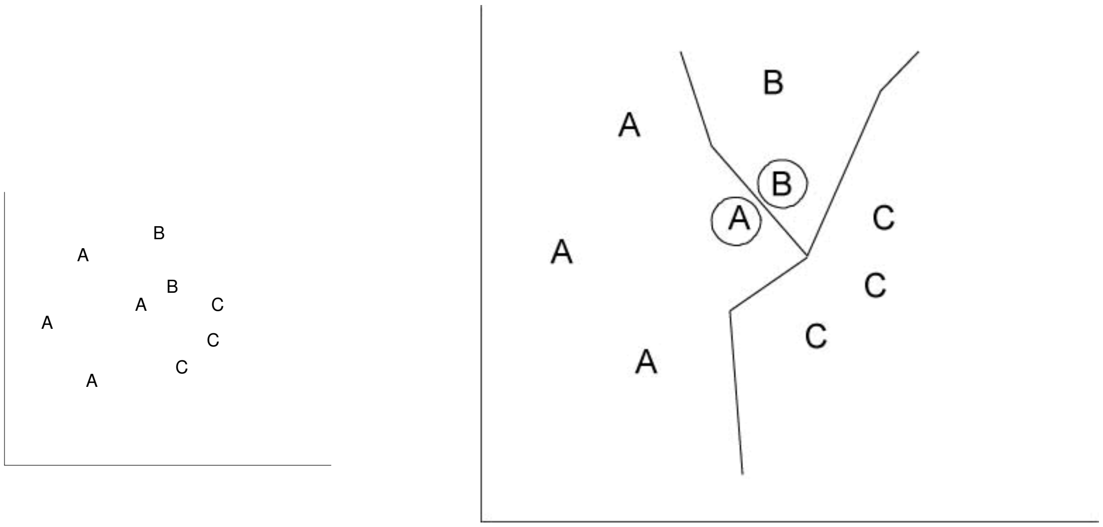
Logistic regression



Support Vector Machine



1-nearest neighbor

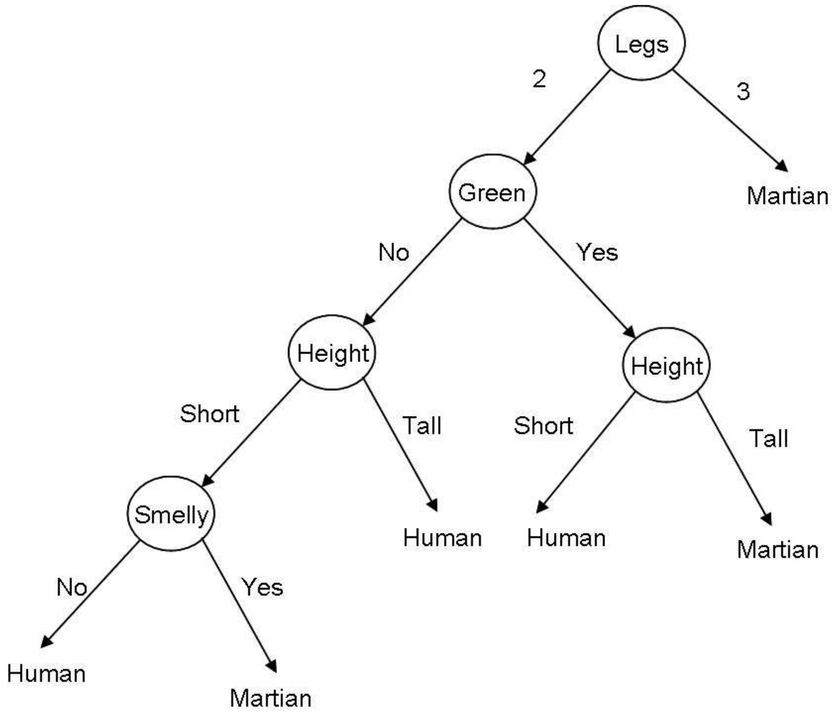


4 [16 points] Decision Trees

NASA wants to be able to discriminate between Martians (M) and Humans (H) based on the following characteristics: Green $\in \{N, Y\}$, Legs $\in \{2, 3\}$, Height $\in \{S, T\}$, Smelly $\in \{N, Y\}$. Our available training data is as follows:

	Species	Green	Legs	Height	Smelly
1)	M	N	3	S	Y
2)	M	Y	2	T	N
3)	M	Y	3	T	N
4)	M	N	2	S	Y
5)	M	Y	3	T	N
6)	H	N	2	T	Y
7)	H	N	2	S	N
8)	H	N	2	T	N
9)	H	Y	2	S	N
10)	H	N	2	T	Y

a)[8 points] Greedily learn a decision tree using the ID3 algorithm and draw the tree.
See the following figure for the ID3 decision tree:

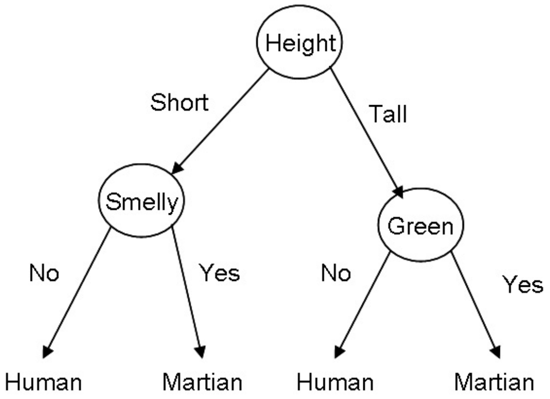


b) i) [3 points] Write the learned concept for Martian as a set of conjunctive rules (e.g., if (green=Y and legs=2 and height=T and smelly=N), then Martian; else if ... then Martian; ...; else Human).

Only the disjunction of conjunctions for Martians was required.
(Legs=3) $\vee$
(Legs=2 $\wedge$ *Green=Yes* $\wedge$ *Height=Tall)* $\vee$
(Legs=2 $\wedge$ *Green=No* $\wedge$ *Height=Short* $\wedge$ *Smelly=Yes)*

ii) [5 points] The solution of part b)i) above uses up to 4 attributes in each conjunction. Find a set of conjunctive rules using only 2 attributes per conjunction that still results in zero error in the training set. Can this simpler hypothesis be represented by a decision tree of depth 2? Justify.

We allowed a little variation on this one because the question could be interpreted as allowing conjunctions with up to two terms. In fact, only two two-term conjunctions are necessary:
(Green=Yes $\wedge$ *Height=Tall)* $\vee$ *(Smelly=Yes* $\wedge$ *Height=Short)*
These conjunctive rules share the height term, so a depth-2 tree is possible. See the figure below.



Notice how ID3 finds a tree that is much longer than the optimal tree. This is due to the greediness of the ID3 algorithm.

5 [23 points] Loss functions and support vector machines

In homework 2, you found a relationship between ridge regression and the maximum a posteriori (MAP) approximation for Bayesian learning in a particular probabilistic model. In this question, you will explore this relationship further, finally obtaining a relationship between SVM regression and MAP estimation.

(a) Ridge regression usually optimizes the squared (L_2) norm:

$$\widehat{\mathbf{w}}_{L_2} = \arg\min_{\mathbf{w}} \sum_{j=1}^{N} (t_j - \sum_i w_i h_i(x_j))^2 + \lambda \sum_i w_i^2. \tag{1}$$

The L_2 norm minimizes the squared residual $(t_j - \sum_i w_i h_i(x_j))^2$, thus significantly weighing outlier points. (An outlier is a data point that falls far away from the prediction $\sum_i w_i h_i(x_j)$.) An alternative that is less susceptible to outliers is to minimize the "sum of absolute values" (L_1) norm:

$$\widehat{\mathbf{w}}_{L_1} = \arg\min_{\mathbf{w}} \sum_{j=1}^{N} |t_j - \sum_i w_i h_i(x_j)| + \lambda \sum_i w_i^2. \tag{2}$$

(i)[2 points] Plot a sketch of the L_1 loss function, do not include the regularization term in your plot. (The x-axis should be the residual $t_j - \sum_i w_i h_i(x_j)$ and the y-axis is the loss function.)

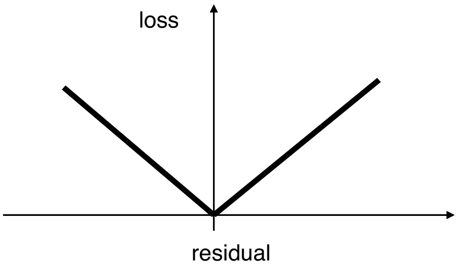


Figure 3: L_1 loss.

(ii)[2 points] Give an example of a case where outliers can hurt a learning algorithm.

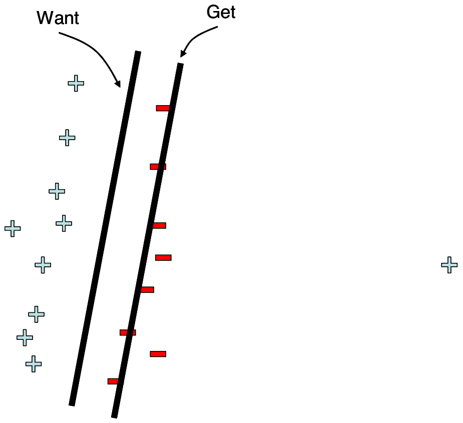


Figure 4: Outlier example.

(iii)[2 points] Why do you think L_1 is less susceptible to outliers than L_2?

L_2 penalizes the square of the residual, so an outlier with residual r will have a loss of r^2. On the other hand, L_1 will have a loss of only $|r|$. Therefore, if $|r| > 1$, this outlier will have a larger influence on the L_2 loss than L_1, and, thus, a greater effect on the solution.

(vi)[2 points] Are outliers always bad and we should always ignore them? Why? (Give one short reason for ignoring outliers, and one short reason against.)

Outliers are often "bad" data, caused by faulty sensors or errors entering values; in such cases, the outliers are not part of the function we want to learn and should be ignore. On the other hand, an outlier could be just an unlikely sample from the true distribution of the function of interest; in these cases, the data point is just another sample and should not be ignored.

(v)[4 points] As with ridge regression in Equation 1, the regularized L_1 regression in Equation 2 can also be viewed a MAP estimator. Explain why by describing the prior $P(\mathbf{w})$ and the likelihood function $P(t \mid \mathbf{x}, \mathbf{w})$ for this Bayesian learning problem. Hint: The p.d.f. of the Laplace distribution is:

$$P(x) = \frac{1}{2b} e^{-|x-\mu|/b}.$$

As with ridge regression, the prior over each parameter is zero-mean Gaussian with variance $1/\lambda$:

$$P(w_i) \sim \mathcal{N}(0; 1/\lambda).$$

The parameters have independent priors:

$$P(\mathbf{w}) = \prod_i P(w_i).$$

The likelihood function is Laplacian with mean $\mathbf{x} \cdot \mathbf{w}$:

$$P(t \mid \mathbf{x}, \mathbf{w}) = \frac{1}{2} e^{-|t-\mathbf{x}\cdot\mathbf{w}|}.$$

(b) As mentioned in class, SVM regression is a margin-based regression algorithm that takes two parameters, $\epsilon > 0$ and $C \geq 0$, as input. In SVM regression, there is no penalty for points that are within ϵ of the hyperplane. Points that are further than ϵ are penalized using the hinge loss. Formally, the SVM regression QP is:

$$\begin{array}{ll} \widehat{\mathbf{w}}_{SVM} = \min_{\mathbf{w},\xi,\bar{\xi}} & \frac{1}{2}\mathbf{w}.\mathbf{w} + C\sum_{j=1}^{m}(\xi_j + \bar{\xi}_j) \\ \text{s.t.} & t_j - \sum_i w_i h_i(x_j) \leq \epsilon + \xi_j \\ & \sum_i w_i h_i(x_j) - t_j \leq \epsilon + \bar{\xi}_j \\ & \xi_j \geq 0, \quad \bar{\xi}_j \geq 0, \quad \forall j \end{array}$$

(i)[4 points] Plot a sketch of the loss function function used by SVM regression. Again, the x-axis should be the residual $t_j - \sum_i w_i h_i(x_j)$ and the y-axis the loss function. However, do not include the $\frac{1}{2}\mathbf{w}.\mathbf{w}$ term in this plot of the loss function.

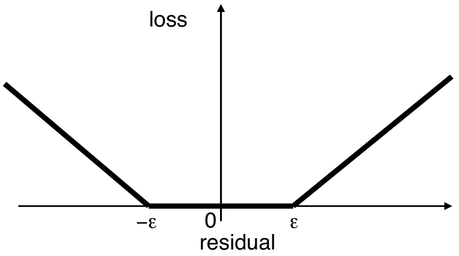


Figure 5: Margin loss.

(ii)[2 points] Compared to L_2 and L_1, how do you think SVM regression will behave in the presence of outliers?

The margin loss is very similar to L_1, so the margin loss will be less susceptible to outliers than L_2. When compared to L_1, the margin loss has an small region ($[-\epsilon, \epsilon]$) with zero penalty, thus, seemingly, margin loss should be less susceptible to outliers than L_1. However, ϵ is usually much smaller than the outlier residual, thus, L_1 and the margin loss will usually have very similar behavior.

(iii)[5 points] SVM regression can also be view as a MAP estimator. What is the prior and the likelihood function for this case?

As with ridge regression, the prior over each parameter is zero-mean Gaussian, but now with variance $2C$:

$$P(w_i) \sim \mathcal{N}(0; 2C).$$

The parameters have independent priors:

$$P(\mathbf{w}) = \prod_i P(w_i).$$

The likelihood function is constant in $[-\epsilon, \epsilon]$ and Laplacian with mean $\mathbf{x} \cdot \mathbf{w}$ elsewhere:

$$P(t \mid \mathbf{x}, \mathbf{w}) = \begin{cases} \frac{1}{2+2\epsilon}, & \textit{for } |t - \mathbf{x} \cdot \mathbf{w}| \leq \epsilon; \\ \frac{1}{2+2\epsilon} e^{-(|t-\mathbf{x}\cdot\mathbf{w}|-\epsilon)}, & \textit{for } |t - \mathbf{x} \cdot \mathbf{w}| > \epsilon. \end{cases}$$

6 [20 points] Learning Theory

This question asks you to consider the relationship between the VC dimension of a hypothesis space H and the number of queries the learner must make (in the worst case) to assure that it exactly learns an arbitrary target concept in H.

More precisely, we have a learner with a hypothesis space H containing hypotheses of the form $h : X \to \{0, 1\}$. The target function $c : X \to \{0, 1\}$ is one of the hypotheses in H. Training examples are generated by the learner posing a query instance $x_i \in X$, and the teacher then providing its label $c(x_i)$. The learner continues posing query instances until it has determined exactly which one of its hypothesis in H is the target concept c.

Show that in the worst case (i.e., if an adversary gets to choose $c \in H$ based on the learner's queries thus far, and wishes to maximize the number of queries), then the number of queries needed by the learner will be at least VC(H), the VC dimension of H. Put more formally, let $MinQueries(c, H)$ be the minimum number of queries needed to guarantee learning target concept c exactly, when considering hypothesis space H. We are interested in the worst case number of queries, $WorstQ(H)$, where

$$WorstQ(H) = max_{c \in H}[MinQueries(c, H)]$$

You are being asked to prove that

$$WorstQ(H) \geq VC(H)$$

You will break this down into two steps:

(a) [8 points] Consider the largest subset of instances $S \subset X$ that can be shattered by H. Show that regardless of its learning algorithm, in the worst case the learner will be forced to pose each instance $x \in S$ as a separate query.

Because S is shattered by H, there will be at least one subset $H^* \subset H$, where each $h \in H^*$ assigns one of the $2^{|S|}$ possible labelings to S. Suppose the adversary chooses a target function c such that $c \in H^*$.

The problem statement says the learner must pose queries until it determines exactly which one of its hypothesis in H is the target concept. Let us assume the learner poses fewer than $|S|$ queries. We will show the learner cannot in this case have converged to just a single consistent candidate hypothesis. Let $x_i \in S$ be one of the instances from S it has not used as a query, and let $A \subset S$ be the set of all instances from S the learner has queried. Because H^* shatters S there are at least two hypotheses $h_1 \in H^*$ and $h_2 \in H^*$ such that both h_1 and h_2 label A correctly, but for which $h_1(x_i) \neq h_2(x_i)$. Therefore, the learner will not have determined which one of the hypotheses in H (or even in H^*) is the target cocept.

(b) [5 points] Use the above to argue that $WorstQ(H) \geq VC(H)$.

We just showed in part (a) that $WorstQ(H) \geq |S|$. By definition, $VC(H) = |S|$. Therefore, $WorstQ(H) \geq VC(H)$.

(c) [7 points] Is there a better case? In other words, if the learner knows that a friend (not an adversary) will be choosing $c \in H$, and that the friend wishes to *minimize* the number of learning queries, is it possible for the friend to choose a c that allows the learner to avoid querying all of the points in S? More formally, if we define

$$BestQ(H) = min_{c \in H}[MinQueries(c, H)]$$

then is the following statement true or false?

$$BestQ(H) \geq VC(H)$$

Justify your answer.

False. In fact, the answer will depend on the exact X and H, and is therefore false in general. To see why, consider the figure below, where X contains exactly 6 instances, and H contains exactly 5 hypotheses. In this diagram,the circle associated with each h indicates which members of X it labels positive. Notice VC(H)=2 in this figure, because the four hypotheses in the lower left of H shatter points $x1$ and $x2$.

Suppose here that the learner first asks for the label of $x3$, and the teacher/friend responds that the label of $x3$ is positive. There is only one hypothesis in H that labels $x3$ positive, so the learner has exactly learned the target function from one example, despite the fact that $VC(H) = 2$.

Notice an adversary could, in this case, respond that the label of $x3$ is negative, thereby forcing the learner to continue to consider the 4 hypotheses that shatter $x1$ and $x2$.

While $BestQ(H) \geq VC(H)$ does not hold for this X and H, it will hold in other cases. For example, if we add hypotheses to H in this example so that it shatters the entire instance space X, then we will have $BestQ(H) = VC(H)$.