10-701 Machine Learning, Fall 2012: Midterm Key

1 Short Questions (50 pts)

Are the following statements True/False? Note that True means "*always* true." Explain your reasoning in only 1 sentence. Each question is worth [2 pts]

1. If the features are independent in a classification problem, i.e. $P(x_1, x_2, ..., x_d) = P(x_1)P(x_2)...P(x_d)$, then the Naive Bayes assumption is satisfied and it is a good choice to classify the data.

   False: Independence does not always imply conditional independence.

   The true reason behind this is: if X_1 and X_2 are independent to each other, and there is another variable Y which is caused by X_1 and X_2 together. It forms a Bayes network $X_1 \rightarrow Y \leftarrow X_2$, but the Naive Bayes assumption expects $X_1 \leftarrow Y \rightarrow X_2$. The latter one is conditionally independent, but the former one is not.

   An easy way to show the falsification of this statement is through the example below: Suppose we have two-dimensional data $(X_1, X_2) \in \mathbb{R}^2$, with label 0 or 1. The distribution of $P(Y)$ is through a uniform distribution, i.e. $P(Y = 0) = P(Y = 1) = 0.5$. In the following figure 1, the data of each class are generated from a mixture Gaussian model (each circle is a Gaussian component). Of course X_1 and X_2 are independent. But Looking at each class, it they are not.

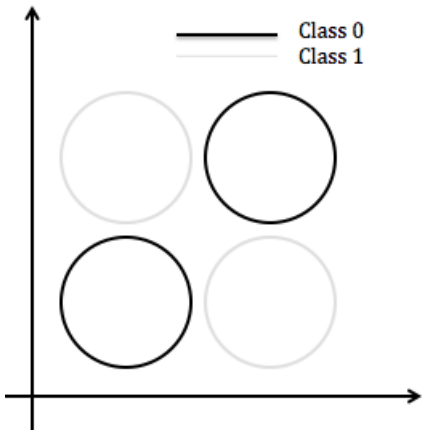


Figure 1: Independence does not imply conditional independence

2. Like the KNN algorithm, SVM will require storing all training instances to label new instances.

False: SVM only needs to store the support vectors instead of all training instances like KNN

3. Hard margin SVM can only have zero training error on linearly separable data.

   False: With kernel trick, data that is not linearly separable can be made linearly separable in higher dimension. Thus, hard margin SVM is able to classify them perfectly if they are linearly separable in higher feature space dimension.

4. Decision trees can only be used for classification.

   False: Can also be used for density estimation and regression.

5. Since instances further away from the decision boundary of SVM are classified with more confidence, these instances are given higher weights α to compute the $w = \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)}$ parameter of SVM.

   False: Due to the design of the optimization objective of SVM, weights α are non-zero only for support vectors, which are instances *nearest* to the decision boundary

6. 4-NN is more likely to over-fit the data than 1-NN.

   False: the opposite

Circle all the correct answers. Explain your reasoning in one or two sentences. Each question is worth [3 pts]

1. In Figure 2, which of the following classifier(s) will obtain zero Leave-One-Out Cross-Validation error on the dataset?

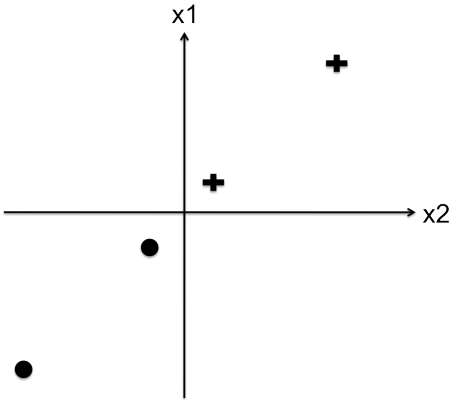


Figure 2: Dataset on 2 dimensional feature space

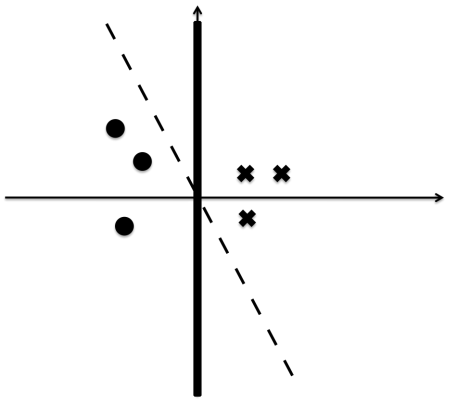


Figure 3: Dataset with two decision boundaries

   (a) Hard margin SVM
   (b) 1-NN
   (c) Logistic regression with regularization on w_2 and $w_0 = 0$
   (d) Logistic regression with regularization on w_1 and $w_0 = 0$

   Answer: c and d

2. In Figure 3, which decision boundary will a hard margin SVM choose?

   (a) The dashed line
   (b) The *vertical* axis (drawn in thick ink)
   (c) Both are equally good
   (d) None of the above

   Answer: b

For $i = 1, 2, ..., n$, let $x_i = 1 + i/n$, and $y_i = 2 + \epsilon_i$ where $\epsilon_i \sim N(0, 0.01)$. Consider estimating the expectation of Y given X = x (a scalar) using the following 3 different models:

i. $E(Y|x) = \beta_0 + \beta_1 x$

ii. $E(Y|x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2$

ii. $E(Y|x) = 1.99$.

Answer the following questions with these models in mind.

3. Which model has the minimum bias

   (a) $E(Y|x) = \beta_0 + \beta_1 x$
   (b) $E(Y|x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2$
   (c) $E(Y|x) = 1.99$

   Answer: (a) and (b), they both have 0 bias.

4. What is the variance of the 3rd model?

   (a) undefined
   (b) 0
   (c) ∞

   Answer: (b), because it is a constant

5. For large n which model has the smallest squared difference risk? Note the variance of $E(Y|x) = \beta_0 + \beta_1 x$ is more than $\frac{\sigma^2}{n}$

   (a) $E(Y|x) = \beta_0 + \beta_1 x$

   (b) $E(Y|x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2$

   (c) $E(Y|x) = 1.99$

   Answer: (a), because when n is large, (a)'s and (b)'s risk will both go to 0; but (a) has less variance than (b),so (a) would have less risk

For the following questions, give short answers (1-3 sentences).

1. Let $f_\theta(x;\theta)$ where $x \in \mathbb{R}$ be the probability density function (pdf) of the uniform distribution over the range 0 to θ. Precisely, $f_\theta(x;\theta) = \frac{1}{\theta}$ if $0 \leq x \leq \theta$ while $f_\theta(x;\theta) = 0$ otherwise.

   Let x_1 to x_n be an independent identically distributed (i.i.d.) sample from f_θ for some unknown true parameter value $\theta > 0$.

   [4 pts] Derive the maximum likelihood estimator (MLE) of θ.

   Answer: $\hat{\theta} = max_i\ x_i$ is the MLE because it is the value that will maximize the likelihood of the data. Any other value less than $\hat{\theta}$ will cause the likelihood of the data to be zero. Any other value larger than $\hat{\theta}$ will cause the likelihood to be less than when $\hat{\theta}$ is used.

2. **[2 pts]** In a three-way classification problem ($Y \in \{0, 1, 2\}$) with k binary features, how many parameters do we need to estimate for the full Bayes classifier? How many parameters for the Naive Bayes classifier?

   Answer: $P(X_1...X_k|Y)$ has $3(2^k - 1)$ parameters; $P(Y)$ has 2. In sum, there are $3 \times 2^k - 1$ for full Bayes. For Naive Bayes it is 3k + 2 in minimal

3. **[4 pts]** Which of the three binary classification problems shown in Figure 4 can be solved by Gaussian Naive Bayes, Logistic Regression, decision trees, and SVM (with proper kernel)?

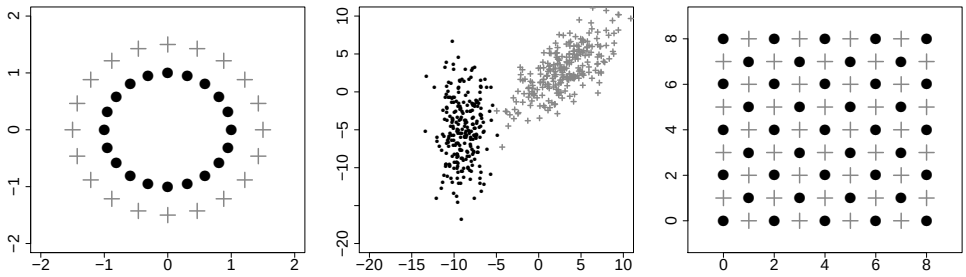


Figure 4

Answer: figure(1): NB, DT, SVM, figure(2): DT, SVM, LR, figure(3) : DT

4. **[2 pts]** If the data is generated by $y = ae^{bx}$ plus some Gaussian noise. Can you use linear regression to estimate the coefficients, a and b? If you can, how? If you cannot, what is the problem?

   Answer: transform e^x as x', and fit regression y = $\beta_1 x' + \beta_0$ as usual. The other correct answer could be no, one may need to state that using transformation, the noise becomes non-gaussian and then linear regression is not minimizing the risk. Both answers are acceptable.

5. **[2 pts]** In Linear regression, suppose we have n data points with k feature dimensions. If one dimension of the data, say $x^{(i)}$ (note that this is an n-dimensional vector), is a linear combination of other dimensions, i.e. $x^{(i)} = \sum_{j=1, j\neq i}^{k} \alpha_j x^{(j)}$. Will it be a problem in linear regression?

   Answer: this is the problem of colinearity, where $X^T X$ is not invertible.

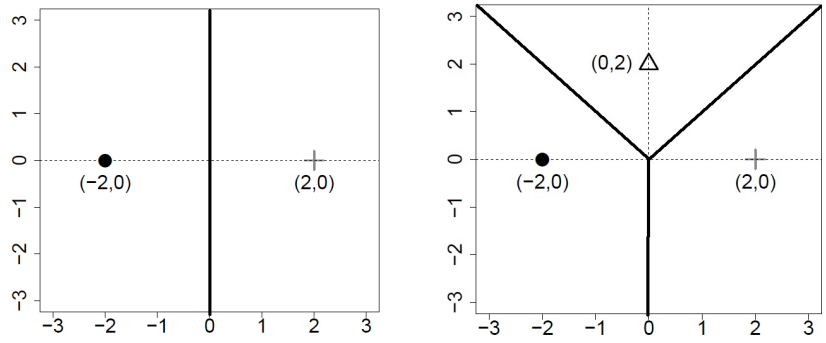
   [2 pts] If it is a problem, how can you fix it without changing the basic model or using extra data?

   Answer: to solve the problem, we can use regularization like lasso or ridge regression.

6. **[2 pts]** Find a kernel function so that SVM can perfectly classify the XOR problem in 2-dimensional space (where the positive examples are (1,1) and (-1,-1), and the negative examples are (-1,1) and (1,-1)).

   Answer: there are many possible answers such as (x,y) = (x, y, xy); also, (x,y) = (xy) can also solve the problem nicely.

7. **[2 pts]** For the two datasets shown in Figure 5 show the 1-NN decision boundary. Assume that each of the points belong to different class. Figure a has two points from two different classes and figure b has three points each belonging to three different class.



(a) (b)

Figure 5

8. **[3 pts]** Given the training data, propose a method to determine the best value of k ("best" in the sense that based on the same training data and distance function, the k-NN algorithm for this particular k is expected to produce minimal test error on a *different* data set drawn from the *same* distribution).

   Answer: do cross validation for all values of k; choose the value of k that minimizes validation error.

2 Generative vs Discriminative Classifier (20 pts)

Consider the binary classification problem where class label $Y \in \{0, 1\}$ and each training example has 2 binary attribute $X_1, X_2 \in \{0, 1\}$.

In this problem, we will always assume X_1 and X_2 are conditional independent given Y, that the class priors are $P(Y = 0) = P(Y = 1) = 0.5$, and the conditional probabilities are as follows:-

$P(X_1\|Y)$	$X_1 = 0$	$X_1 = 1$
$Y = 0$	0.7	0.3
$Y = 1$	0.2	0.8

$P(X_2\|Y)$	$X_2 = 0$	$X_2 = 1$
$Y = 0$	0.9	0.1
$Y = 1$	0.5	0.5

The expected rate is the probability that the classifier provides an incorrect prediction for an observation: if Y is the true label, let $\hat{Y}(X_i, X_2)$ be the predicted class label, then the expected rate error is

$$P_{\mathcal{D}}(Y = 1 - \hat{Y}(X_i, X_2)) = \sum_{X_1=0}^{1} \sum_{X_2=0}^{1} P_{\mathcal{D}}(X_1, X_2, Y = 1 - \hat{Y}(X_i, X_2)) \tag{1}$$

- **[8 pts]** Write down the naive bayes prediction for all of the four possible configuration of X_1, X_2. This table should help in finding the solution.

X_1	X_2	$P(X_1, X_2, Y = 0)$	$P(X_1, X_2, Y = 1)$	$\hat{Y}(X_1, X_2,)$
0	0	0.7 × 0.9×0.5	0.2×0.5×0.5	0
0	1	0.7×0.1×0.5	0.2×0.5×0.5	1
1	0	0.3×0.9×0.5	0.8×0.5×0.5	1
1	1	0.3×0.1×0.5	0.8×0.5×0.5	1

- **[4 pts]** Compute the expected error rate of this naive bayes classifier which predicts Y given both the attributes $\{X_1, X_2\}$. Assume that the classifier is learned with infinite training data.

  Ans The expected error rate is given by:

$$0.2 \times 0.5 \times 0.5 + 0.7 \times 0.1 \times 0.5 + 0.3 \times 0.9 \times 0.5 + 0.3 \times 0.1 \times 0.5 = 0.235$$

- **[2 pts]** Which of the following has smaller expected error rate?
  - the naive Bayes classifier which predicts Y given X_1 only.
  - the naive Bayes classifier which predicts Y given X_2 only.

  Ans: The first one since the expected error rate is smaller.

- **[3 pts]** Now suppose we create a new attribute X_3 which is an exact replica of X_2. Will the expected error rate of the naive Bayes which predicts Y given all the attributes (X_1, X_2, X_3) be more, equal or less than calculated in part 2? Assume that the classifier is learned from infinite training data. Explain your reason.

  Ans: Expected error rate is more since the conditional independence (CI) assumption is not satisfied. The error can also be calculated and it comes out to 0.3.

- **[3 pts]** Does Logistic regression suffer form the same problem? Why?

  Ans: Logistic regression does not suffer from this because it does not make such CI assumption

3 Probablistic Interpretation of KNN (12 pts)

In this question, we are going to see that k-NN can be thought of as the Bayes optimal classifier based on a certain density estimator.

Suppose we have n training points $x_1, ..., x_n \in \mathbb{R}^d$, each with a corresponding label $y_i \in \{0, 1\}$ (we could generalize to more than 2 classes, but we won't do that here for simplicity). For some point $x \in \mathbb{R}^d$ (not necessarily one of the training points), let $S(x, k)$ be the *smallest* d dimensional ball centered at x that contains k training points, and let $V(x, k)$ be the volume of that ball. (For example, $V(x_i, 1) = 0$ for any $i = 1, ..., n$; can you see why?)

Consider the following "density" estimator:

$$\widehat{p}_k(x) = \frac{k}{nV(x, k)}.$$

Intuitively, we define $\widehat{p}_k$ so that *if* it were constant in the ball $S(x_0, k)$ for some x_0 (it is *not* actually constant), then the total probability (not density) assigned to that ball could be $\widehat{p}_k(x_0)V(x_0, k) = k/n$, i.e. the fraction of all training points that fall in that ball. Note that this is not a true density estimator, since it does not integrate to 1, but we ignore that here.

Let n_0 and n_1 be the number of i for which $y_i = 0$ and $y_i = 1$, respectively, in the training data (i.e. $n_0 + n_1 = n$). For any $x \in \mathbb{R}^d$ and $j = 0, 1$, let $k_j(x, k)$ be the number of points with label j in the ball $B(x, k)$ (i.e. $k_0(x, k) + k_1(x, k) = k$).

1. **[5 pts]** Come up with class-conditional density estimators $\widehat{p}_k(x|Y = j)$ analogous to the density estimator above, using only $k_j(x, k)$, n_j, and $V(x, k)$ (do not define any new quantities). Also write down expressions for the class probabilities $P(Y = j)$. (**Hint:** make sure that $\widehat{p}_k(x) = \widehat{p}_k(x|Y = 0)P(Y = 0) + \widehat{p}_k(x|Y = 1)P(Y = 1)$).

   Answer:

   $$\hat{p}_k(x|Y = j) = \frac{k_j(x, k)}{n_j V(x, k)}, \quad \hat{p}(Y = j) = \frac{n_j}{n}$$

2. **[5 pts]** Derive the posterior probability of $P(Y = j|X = x)$, $j =\in \{0, 1\}$, at a point x using Bayes' Rule. Simplify as much as possible.

   Answer:

   $$P(Y = j|X = x) = \frac{\hat{p}_k(x|Y = j)\hat{p}(Y = j)}{\hat{p}_k(x)} = \frac{\frac{k_j(x,k)}{n_j V(x,k)} \times \frac{n_j}{n}}{\frac{k}{nV(x,k)}} = \frac{k_j(x, k)}{k}$$

3. **[2 pt]** Show that using the Bayes optimal decision rule based on $P(Y = j|X = x)$ is the same as applying k-NN.

   Answer:

   $$j = \arg\max_j P(Y = j|X = x) = \arg\max_j \frac{k_j(x, k)}{k} \propto \arg\max_j k_j(x, k)$$

   It is same as choosing the majority class in k nearest neighbors as j.

4 Linear Regression (18 pts)

Let $x_1, ..., x_n \in \mathbb{R}^d$ fixed. Assume that $y_i = \beta^T x_i + \epsilon_i$ with $\epsilon_i \sim \mathcal{N}(0,1)$. We saw in class that under these assumptions, if we place a prior distribution on β as $\beta_j \sim \text{Laplace}(0,t)$ (i.i.d.) for $j = 1, ..., d$, then the maximum a posteriori estimator of β can be obtained by solving the L1 regularized least squares regression problem:

$$\widehat{\beta} = \arg\min_{\beta} \frac{1}{2} \sum_{i=1}^{n} (y_i - \beta^T x_i)^2 + \lambda \|\beta\|_1 \tag{2}$$

where $\lambda = 1/t$.

1. **[6 pts]** Given a list of numbers $\lambda_1, ..., \lambda_d > 0$, we could replace the $\lambda\|\beta\|_1$ term in L1 regularized regression by $\sum_{j=1}^{d} \lambda_j |\beta_j|$, making the combined objective:

$$\widehat{\beta} = \arg\min_{\beta} \frac{1}{2} \sum_{i=1}^{n} (y_i - \beta^T x_i)^2 + \sum_{j=1}^{d} \lambda_j |\beta_j|. \tag{3}$$

   Under what prior distributions on the β_j would the MAP estimator of $\beta = (\beta_1, ..., \beta_d)$ take on the same form as (3)?

   Hint: Do not modify the assumptions that $y_i = \beta^T x_i + \epsilon_i$ with $\epsilon_i \sim \mathcal{N}(0,1)$.

   Hint 2: The density of the centered (i.e. mean 0) Laplace distribution with parameter t is as follows (in case you need it):

$$p(z; 0, t) = \frac{1}{2t} e^{-|z|/t}.$$

   Answer:
   We can take $\beta_j \sim \text{Laplace}(0, 1/\lambda_j)$ independent of each other. The term in the overall negative log likelihood of the posterior corresponding to the priors on the β_j will then clearly take on the required form (along with terms constant in β_j):

$$-\log \prod_{j=1}^{d} p(\beta_j; 0, 1/\lambda_j) = \sum_{j=1}^{d} \lambda_j |\beta_j| + \log(2/\lambda_j).$$

2. **[3 pt]** In what case might it be a good idea to replace λ with $\lambda_1, ..., \lambda_d$, as done in (3)? Useful fact – the variance of a Laplace$(0,t)$ random variable is $2t^2$.

   Answer:
   Many answers are possible. For example if some features have much larger variance than others, it does not make sense to penalize their coefficients on the same scale. Prior information on the relative importance of different features can also play a role.

3. **[6 pts]** Given $\lambda_1, ..., \lambda_d > 0$, come up with values for $\widetilde{x}_1, ..., \widetilde{x}_n \in \mathbb{R}^d$ and λ so that

$$\min_{\beta} \frac{1}{2} \sum_{i=1}^{n} (y_i - \beta^T x_i)^2 + \sum_{j=1}^{d} \lambda_j |\beta_j| = \min_{\beta} \frac{1}{2} \sum_{i=1}^{n} (y_i - \beta^T \widetilde{x}_i)^2 + \lambda \|\beta\|_1. \tag{4}$$

   (Notice these are min's, **not** arg min's.)

   Answer:
   It is sufficient to take $\lambda = 1$ and $\widetilde{x}_{ij} = x_{ij}/\lambda_j$ for $i = 1, ..., n$ and $j = 1, ..., d$.

4. **[3 pt]** In light of your answer for part 2, what is the intuitive meaning of the transformation from $x_1, ..., x_n$ to $\widetilde{x}_1, ..., \widetilde{x}_n$?

   Answer:
   Normalization! If we chose different λ_j to account for different variances, dividing each feature by λ_j simply equalized their standard deviation (recall the variance of the Laplace$(0, 1)$ was $2t^2$).