CS 189 Fall 2015

Introduction to Machine Learning

Final

- Please do not turn over the page before you are instructed to do so.
- You have 2 hours and 50 minutes.
- Please write your initials on the top-right of each odd-numbered page (e.g., write "AE" if you are Alexei Efros). Complete this by the end of your 2 hours and 50 minutes.
- The exam is closed book, closed notes except your one-page cheat sheet.
- No calculators or other electronic devices allowed.
- Mark your answers ON THE EXAM ITSELF IN THE SPACE PROVIDED. If you are not sure of your answer you may wish to provide a *brief* explanation. Do NOT attach any extra sheets.
- The total number of points is 150. There are 15 true/false questions worth 2 points each, 10 multiple choice questions worth 3 points each, and 6 descriptive questions with unequal point assignments.
- For true/false questions, fill in the *True/False* bubble.
- For multiple-choice questions, fill in the bubbles for **ALL** CORRECT CHOICES: There may be more than one correct choice, but there will be at least one correct choice. NO PARTIAL CREDIT: the set of all correct answers must be checked.

First name	
Last name	
SID	
First and last name of student to your left	
First and last name of student to your right	

Q1. [30 pts] True or False

(1) [2 pts] Random forests usually perform better than AdaBoost when your dataset has mislabeled data points.
◯ True ◯ False

(2) [2 pts] The discriminant function computed by kernel methods are a linear function of its parameters, not necessarily a linear function of the inputs.
◯ True ◯ False

(3) [2 pts] The XOR operator can be modeled using a neural network with a single hidden layer (i.e. 3-layer network).
◯ True ◯ False

(4) [2 pts] Convolutional neural networks are rotation invariant.
◯ True ◯ False

(5) [2 pts] Making a decision tree deeper will assure better fit but reduce robustness.
◯ True ◯ False

(6) [2 pts] Bagging makes use of the bootstrap method.
◯ True ◯ False

(7) [2 pts] K-means automatically adjusts the number of clusters.
◯ True ◯ False

(8) [2 pts] Dimensionality reduction can be used as pre-processing for machine learning algorithms like decision trees, kd-trees, neural networks etc.
◯ True ◯ False

(9) [2 pts] K-d trees guarantee an exponential reduction in the time it takes to find the nearest neighbor of an example as compared to the naive method of comparing the distances to every other example.
◯ True ◯ False

(10) [2 pts] Logistic regression is equivalent to a neural network without hidden units and using cross-entropy loss.
◯ True ◯ False

(11) [2 pts] Convolutional neural networks generally have fewer free parameters as compared to fully connected neural networks.
◯ True ◯ False

(12) [2 pts] K-medoids is a kind of agglomerative clustering.
◯ True ◯ False

(13) [2 pts] Whitening the data doesn't change the first principal direction.
◯ True ◯ False

(14) [2 pts] PCA can be kernelized.
◯ True ◯ False

(15) [2 pts] Performing K-nearest neighbors with $K = N$ yields more complex decision boundaries than 1-nearest neighbor.
◯ True ◯ False

Q2. [30 pts] Multiple Choice

(1) [3 pts] Which of the following guidelines is applicable to initialization of the weight vector in a fully connected neural network.

○ Should not set it to zero since otherwise it will cause overfitting

○ Should not set it to zero since otherwise (stochastic) gradient descent will explore a very small space

○ Should set it to zero since otherwise it causes a bias

○ Should set it to zero in order to preserve symmetry across all neurons

(2) [3 pts] Duplicating a feature in linear regression

○ Can reduce the L2-Penalized Residual Sum of Squares.

○ Does not reduce the Residual Sum of Squares (RSS).

○ Can reduce the L1-Penalized Residual Sum of Squares (RSS).

○ None of the above

(3) [3 pts] Which of the following is/are forms of regularization in neural networks.

○ Weight decay

○ L2 regularization

○ L1 regularization

○ Dropout

(4) [3 pts] We are given a classifier that computes probabilities for two classes (positive and negative). The following is always true about the ROC curve, and the area under the ROC curve (AUC):

○ An AUC of 0.5 represents a classifier that performs worse than random.

○ We generate an ROC curve by varying the discriminative threshold of our classifier.

○ The ROC curve allows us to visualize the tradeoff between true positive and false positive classifications.

○ The ROC curve monotonically increases.

(5) [3 pts] The K-means algorithm:

○ Requires the dimension of the feature space to be no bigger than the number of samples

○ Has the smallest value of the objective function when $K = 1$

○ Minimizes the within class variance for a given number of clusters

○ Converges to the global optimum if and only if the initial means are chosen as some of the samples themselves

○ None of the above

(6) [3 pts] Suppose when you are training your convolutional neural network, you find that the training loss just doesn't go down after initialization. What could you try to fix this problem?

○ Change the network architecture

○ Change learning rates

○ Ensure training data is being read correctly

○ Find a better model

○ Normalize the inputs to the network

○ Add a regularization term

(7) [3 pts] Logistic regression:

- ◯ Minimizes cross-entropy loss
- ◯ Has a simple, closed form analytical solution
- ◯ Models the log-odds as a linear function
- ◯ Is a classification method to estimate class posterior probabilities

(8) [3 pts] Select all the true statements.

- ◯ The first principal component is unique up to a sign change.
- ◯ The last principal component is unique up to a sign change.
- ◯ All principal components are unique up to a sign change.
- ◯ If some features are linearly dependent, at least one singular value is zero.
- ◯ If some features are correlated, at least one singular value is zero.

(9) [3 pts] Select all the choices that make the following statement true:
In (a), the training error does not increase as (b) increases.

- ◯ a: K-means,
b: number of iterations
- ◯ a: Training neural nets with back propagation using *batch* gradient decent,
b: number of iterations
- ◯ a: Training neural nets with back propagation using *stochastic* gradient decent,
b: number of iterations
- ◯ a: Regression Trees with square loss,
b: depth of the tree
- ◯ a: Random Forest Classifier,
b: number of trees in the forest
- ◯ a: Least squares,
b: number of features

(10) [3 pts] Neural networks:

- ◯ Optimize a convex objective function
- ◯ Can only be trained with stochastic gradient descent
- ◯ Can use a mix of different activation functions
- ◯ Can be made to perform well even when the number of parameters/weights is much greater than the number of data points.

Q3. [15 pts] Nearest Neighbors and Bayes risk

In this problem, we want to investigate whether given enough training examples, the Bayes decision rule gives more accurate results that nearest neighbors.

A life insurance company needs to estimate whether a client is at risk of dying in the year to come, based on his age and blood pressure. We call $\mathbf{x} = [A, B]$ (A=Age, B=Blood pressure) the two dimensional input vector and y the outcome ($y = 1$ if the client dies and $y = -1$ otherwise). The insurance company has a lot of data, enough to estimate accurately with Parzen windows the posterior probability $P(y = 1|\mathbf{x})$. This is represented in a diagram in Figure 1.

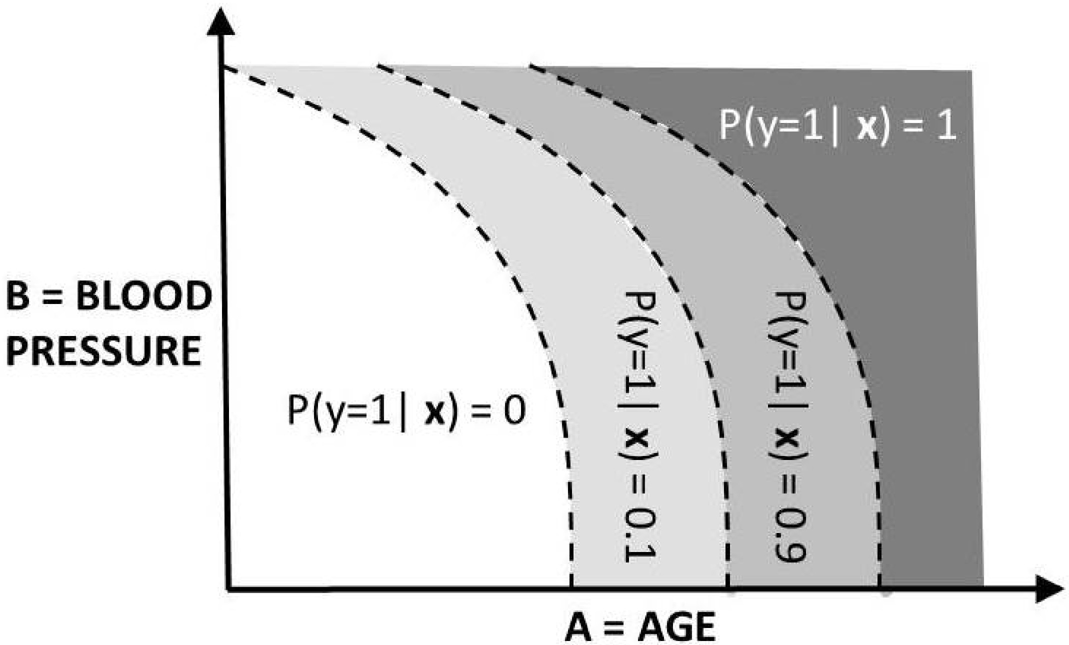


Figure 1: Draw your answer to the Nearest Neighbor and Bayes risk problem. Note that the distribution $P(x)$ is assumed to be uniform across the area shown in the figure.

Note: No worries, nobody died, this is a fictitious example. For simplicity we have 4 regions in which $P(y = 1|\mathbf{x})$ is constant, and the distribution $P(x)$ is assumed to be uniform across the area shown in the figure.

Let us name the different regions:

$$
\begin{aligned}
R1 :& P(y = 1|\mathbf{x}) = 0 \\
R2 :& P(y = 1|\mathbf{x}) = 0.1 \\
R3 :& P(y = 1|\mathbf{x}) = 0.9 \\
R4 :& P(y = 1|\mathbf{x}) = 1
\end{aligned}
$$

(1) [2 pts] Draw on the figure the Bayes optimum decision boundary with a thick line.

(2) [2 pts] What is the Bayes risk in each of the four regions (the Bayes risk is the probability of error of the optimum Bayes classifier).

(3) [4 pts] Assume we have lots and lots of samples due to which we can assume that the nearest neighbor of any sample lies in the same region as that sample. Now consider any sample, say, $\mathbf{x}$ which falls in region R_i. For $i \in \{1, 2, 3, 4\}$, find the probability that $\mathbf{x}$ and its nearest neighbor belong to different classes (that is, have different labels):

(4) [2 pts] What is the nearest neighbor error rate ENN in each region:

(5) [5 pts] Now let us generalize the previous results to the case where the posterior probabilities are:

$$
\begin{aligned}
R1 :& P(y = 1|\mathbf{x}) = 0 \\
R2 :& P(y = 1|\mathbf{x}) = p \\
R3 :& P(y = 1|\mathbf{x}) = (1 - p) \\
R4 :& P(y = 1|\mathbf{x}) = 1
\end{aligned}
$$

where p is a number between 0 and 0.5.
After recalculating the results of the previous questions, give an upper bound and a lower bound of ENN in terms of EBayes.

Q4. [11 pts] Curse of dimensionality

When the dimension of input space d is large, the performance of the nearest neighbor algorithm (and other local methods such as "Parzen windows") tends to deteriorate. This phenomenon is known as the "curse of dimensionality". In the following questions, we will assume that we have a training set of fixed size N and that all features are uniformly distributed on $[0, 1]$. Associated with each test example $\mathbf{x}$ is a predicted response y corresponding to the average of the responses associated to the training examples that are near $\mathbf{x}$.

(1) [1 pt] Suppose we have only one feature ($d = 1$) and we want to make prediction using only training examples that are within 10% of the input range. For instance, to predict the response y of $x = 0.6$, we will use the training examples that fall in the range $[0.55, 0.65]$ only. On average, what fraction of the training examples will we use to make each prediction?

(2) [1 pt] Now suppose that we have two features ($d = 2$) and we want to predict using only training examples that are within 10% of the input range in both dimensions. For instance, to predict the response y of $\mathbf{x} = (0.6, 0.35)$, we will use the training examples that fall in the range $[0.55, 0.65]$ for the first feature and $[0.3, 0.4]$ for the second one. On average, what fraction of the training examples will we use to make each prediction?

(3) [4 pts] Generalize your response for the general case of any dimension d. Argue that a drawback of methods based on nearest neighbors is that, when d is large, there are very few training examples near any test example.

(4) [5 pts] Now suppose that we wish to make a prediction of a test example $\mathbf{x}$ by creating a d-dimensional hypercube centered around $\mathbf{x}$ that contains on average 10% of the training examples. For d=1, 2, 3, and 100, what is the length of each side of the hypercube? Explain the implication of your answer on the performance of the nearest neighbors algorithm.

Q5. [22 pts] Decision Trees and Random Forests

Consider constructing a decision tree on data with d features and n training points where each feature is real-valued and each label takes one of m possible values. The splits are two-way, and are chosen to maximize the information gain. We only consider splits that form a linear boundary parallel to one of the axes. In parts (a), (b) and (c) we will consider a standalone decision tree and not a random forest, so no randomization.

(1) [4 pts] Prove or give a counter-example: For every value of $m > 3$, there exists some probability distribution on m objects such that its entropy is less than -1.

(2) [4 pts] Prove or give a counter-example: In any path from the root split to a leaf, the same feature will never be split on twice.

(3) [4 pts] Prove or give a counter-example: The information gain at the root is at least as much as the information gain at any other node.

(4) [4 pts] One may be concerned that the randomness introduced in random forests may cause trouble, for instance, some features or samples may not be considered at all. We will investigate this phenomenon in the next two parts.

Consider n training points in a feature space of d dimensions. Consider building a random forest with t binary trees, each having exactly h internal nodes. Let f be the number of features randomly selected at each node. In order to simplify our calculations, we will let $f = 1$. For this setting, compute the probability that a certain feature (say, the first feature) is never considered for splitting.

(5) [3 pts] Now let us investigate the concern regarding the random selection of the samples. Suppose each tree employs n bootstrapped training samples. Compute the probability that a particular sample (say, the first sample) is never considered in any of the trees.

(6) [3 pts] Compute the values of the probabilities you obtained in the previous two parts for the case when there are $n = 2$ training points, $d = 2$ dimensions, $t = 10$ trees of depth $h = 4$ (you may leave your answer in a fraction and exponentiated form, e.g., as $\left(\frac{51}{100}\right)^2$. What conclusions can you draw from your answer with regard to the concern mentioned in the beginning of the problem?

Q6. [12 pts] Elastic net regularization

A powerful method for regularizing linear regression is called elastic net regularization, which combines ridge regression (L2 regularization) and Lasso (L1 regularization).

Observe that linear regression can be probabilistically modeled as $P(y^{(k)}|\mathbf{x}^{(k)}, \mathbf{w}, \sigma^2) \sim \mathcal{N}(\mathbf{w}^T\mathbf{x}^{(k)}, \sigma^2)$. This means $P(y^{(k)}|\mathbf{x}^{(k)}, \mathbf{w}, \sigma^2) =$

$$\frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{(y^{(k)} - \mathbf{w}^T\mathbf{x}^{(k)})^2}{2\sigma^2})$$

It is then possible to show that ridge regression is equivalent to MAP estimation with a Gaussian prior, and Lasso is equivalent to MAP estimation with a Laplace prior.

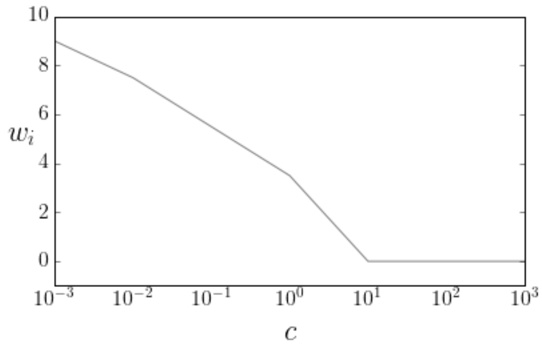
Let us assume a different prior distribution. Assume each weight w_j is i.i.d, drawn from a distribution such that $P(w_j) = q\exp(-\alpha_1|w_j| - \alpha_2 w_j^2)$, where q, α_1, α_2 are fixed constants. Our training set is $(\mathbf{x}^{(k)}, y^{(k)}), ...(\mathbf{x}^{(n)}, y^{(n)})$.

(1) [6 pts] Show that the MAP estimate for $\mathbf{w}$ is equivalent to minimizing the following risk function, for some choice of constants λ_1, λ_2:

$$R(\mathbf{w}) = \sum_{k=1}^{n}(y^{(k)} - \mathbf{w}^T\mathbf{x}^{(k)})^2 + \lambda_1\|\mathbf{w}\|_1 + \lambda_2\|\mathbf{w}\|_2^2$$

(2) [2 pts] Suppose we scale both λ_1 and λ_2 by a positive constant c. The graph below represents the value of a single one of the weights, w_i graphed against the value of c. Out of the following set of values for λ_1, λ_2, which best corresponds to the graph (select exactly one option)?

○ $\lambda_1 = 1, \lambda_2 = 0$ ○ $\lambda_1 = 0, \lambda_2 = 1$

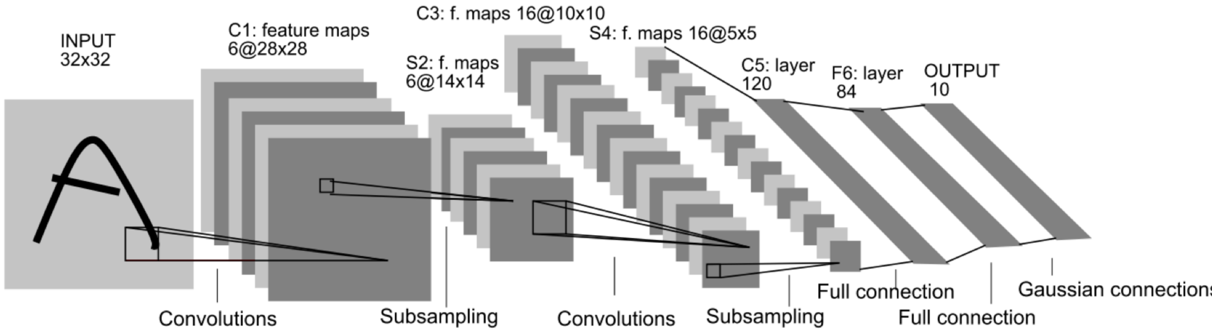


(3) [2 pts] Explain why your choice in **(b)** results in the graph.

(4) [2 pts] What is the advantage of using Elastic net regularization over using a single regularization term of $||\mathbf{w}||_p$, where $1 < p < 2$?

Q7. [14 pts] Neural Networks

(1) [6 pts] Here is the historical *LeNet* Convolutional Neural Network architecture of Yann LeCun et al. for digit classification that we've discussed in class. Here, the INPUT layer takes in a 32x32 image, and the OUTPUT layer produces 10 outputs. The notation 6@28x28 means 6 matrices of size 28x28.



If the parameters of a given layer are the weights that connect to its inputs,

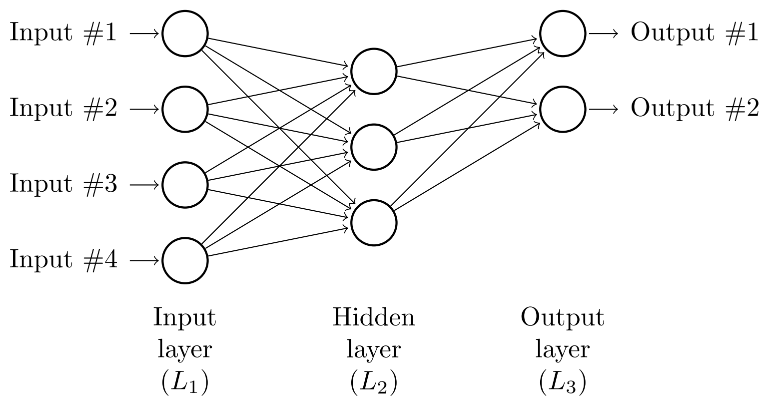
- Given that the input size is 32x32, and the Layer 1 size is 28x28, what's the size of the convolutional filter in the first layer (i.e. how many inputs is each neuron connected to)?

- How many independent parameters (weight and bias) are in layer C1?

- How many independent parameters (weight and bias) are in layer C3?

- How many independent parameters (weight and bias) are in layer F6?

(2) [8 pts] Consider a three layer fully-connected network with n_1, n_2, n_3 neurons in three layers respectively. Inputs are fed into the first layer. The loss is mean squared error E, and the non-linearity is a sigmoid function. Let the label vector be t of size n_3. Let each layer output vector be y_i and input vector be z_i, both of size n_i. Let the weight between layer i and layer $i+1$ be W_{ii+1}. The j-th element in y_i is defined by y_i^j, same for z_i^j. The weight connecting k-th and l-th neuron in $i, i+1$ layers is defined by W_{ii+1}^{kl} (You don't need to consider bias in this problem).



Here is a summary of our notation:

- σ denotes the activation function for L_2 and L_3, $\sigma(x) = \frac{1}{1+e^{-x}}$. There is no activation applied to the input layer.
- $z_i^{(j)} = \sum_{k=1}^{P} W_{i-1i}^{kj} x_{i-1}^{(k)}$
- $y_i^{(j)} = \sigma\left(\sum_{k=1}^{P} W_{i-1i}^{kj} x_{i-1}^{(k)}\right)$

Now solve the following problems.

- Find $\frac{\partial E}{\partial z_3^j}$ in terms of y_3^j.

- Find $\frac{\partial E}{\partial y_2^k}$ in terms of elements in W_{23} and $\frac{\partial E}{\partial z_3^j}$.

- Find $\frac{\partial E}{\partial W_{23}^{kj}}$ in terms of y_2^k, y_3^j and t^j.

- If the input to a neuron in max-pooling layer is x and the output is $y = max(x)$, derive $\frac{\partial y}{\partial x_i}$.

Q8. [16 pts] The dimensions are high! And possibly hard too.

In this problem, we will derive a famous result called the "Johnson-Lindenstrauss" lemma. Suppose you are given n arbitrary vectors $x^1, \ldots, x^n \in \mathbb{R}^{d\times 1}$. Let $k = 320 \log n$. Now consider a matrix $A \in \mathbb{R}^{k\times d}$ that is obtained randomly in the following manner: every entry of the matrix is chosen independently at random from $\mathcal{N}(0,1)$. Define vectors $z^1, \ldots, z^n \in \mathbb{R}^{k\times 1}$ as $z^i = \frac{1}{\sqrt{k}} A x^i$ for every $i \in \{1, \ldots, n\}$.

(1) [4 pts] For any given $i \in \{1, \ldots, n\}$, what is the distribution of the random vector Ax^i? Your answer should be in terms of the vector x^i.

(2) [4 pts] For any distinct $i, j \in \{1, \ldots, n\}$, derive a relation between $\mathbb{E}[||A(x^i - x^j)||_2^2]$ and the value of $||x^i - x^j||_2^2$? More points for deriving the relation using your answer from part (1) above.

(3) [4 pts] It can be shown that for any fixed vector v, the random matrix A has the property that

$$\frac{3}{4}||v||_2^2 \leq ||Av||_2^2 \leq \frac{5}{4}||v||_2^2$$

with probability at least $1 - \frac{1}{n^4}$. Using this fact, show that with probability at least $1 - \frac{1}{n^2}$, every pair (z^i, z^j) simultaneously satisfies $\frac{3}{4}||x^i - x^j||_2^2 \leq ||z^i - z^j||_2^2 \leq \frac{5}{4}||x^i - x^j||_2^2$. (Think of how you would bound probabilities of multiple events. Only requires a very basic fact about probability and a little thought.)

(4) [4 pts] Describe, in at most two sentences, the usefulness of this result. (Think of n and d as having very large values, for instance, several billions).