Midterm exam CS 189/289, Fall 2015

- You have **80 minutes** for the exam.
- **Total 100 points:**
  1. **True/False:** 36 points (18 questions, 2 points each).
  2. **Multiple-choice questions**: 24 points (8 questions, 3 points each).
  3. **Three descriptive** questions worth 10, 15, 15 points.
- The exam is closed book, closed notes except your one-page crib sheet.
- No calculators or electronic items.
- For true/false questions, fill in the True/False bubble.
- **WARNING** For multiple-choice questions, fill in the bubbles for **ALL CORRECT CHOICES (in some cases, there may be more than one). NO PARTIAL CREDIT: 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	

For staff only	
T/F	/36
Multiple choice	/24
Problem I	/15
Problem II	/15
Problem III	/10
Total	/100

Notation:

X: the training data matrix of dimension (N, d), of N rows representing samples and d columns representing features.

$\mathbf{x}$: an input data vector of dimension (1, d) of components x_i, i=1:d.

$\mathbf{x}^k$: a training example of dimension (1, d) is a row of X, k=1:N.

$\mathbf{w}$: weight vector of a linear model of dimension (1, d) such that

$f(\mathbf{x}) = \mathbf{w}\ \mathbf{x}^T = \mathbf{x}\ \mathbf{w}^T = \sum_{i=1:d} w_i\ x_i$

$\mathbf{y}$: target vector of dimension (N, 1) of components y^k.

α: weight vector of dimension (N, 1) of kernel method $f(\mathbf{x}) = \sum_{k=1:N} \alpha_k\ k(\mathbf{x}, \mathbf{x}^k)$

$k(\mathbf{u}, \mathbf{v})$: a kernel function (a similarity measure between two samples **u** and **v**).

True/False (36 points):

1. Stochastic gradient descent performs less computation per update than batch gradient descent. *

   TRUE 🟢 FALSE ○

2. A function is convex if its Hessian is negative semidefinite. *

   TRUE ○ FALSE 🔴

3. If N < d, the solution to $X\mathbf{w}^T = \mathbf{y}$ is unique. **

   TRUE ○ FALSE 🔴

4. A support vector machine computes $P(y|\mathbf{x})$. **

   TRUE ○ FALSE 🔴

5. Adding a ridge to X^TX guarantees that it is invertible. *

   TRUE 🟢　　　　FALSE ○

6. Grid search is less prone to being trapped in a local minimum than other heuristic search methods. ***

   TRUE 🟢　　　　FALSE ○

7. The bootstrap method involves sampling without replacement. *

   TRUE ○　　　　FALSE 🔴

8. A non linearly-separable training set in a given feature space can always be made linearly-separable in another space.**

   TRUE 🟢　　　　FALSE 🔴

9. Using the kernel trick, one can get non-linear decision boundaries using algorithms designed originally for linear models. *

   TRUE 🟢　　　　FALSE ○

10. Logistic regression cannot be kernelized.*

    TRUE ○　　　　FALSE 🔴

11. Ridge regression, weight decay, and Gaussian processes use the same regularizer: $\|\mathbf{w}\|^2$. *

    TRUE 🟢　　　　FALSE ○

12. Hebb's rule computes the centroid method solution if the target values are $+1/N_1$ and $-1/N_0$ (N_1 and N_0 are the number of examples of each class)**

    TRUE 🟢　　　　FALSE ○

13. Any kernel method can be thought of as a parametric method in a possibly infinite dimensional space.*

TRUE 🟢 FALSE ○

14.Nearest neighbors is a parametric method.*

TRUE ○ FALSE 🔴

15.A symmetric matrix is positive semidefinite if all its eigenvalues are positive or zero. **

TRUE 🟢 FALSE ○

16.Zero correlation between any two random variables implies that the two random variables are independent. ***

TRUE FALSE

17.The Linear Discriminant Analysis (LDA) classifier computes the direction maximizing the ratio of between-class variance over within-class variance. ***

TRUE 🟢 FALSE ○

18.If we repeat an experiment twice and get p-values p1 and p2, the minimum of the two p-values is the p-value of the overall experiment. ***

TRUE ○ FALSE 🔴

Multiple choice questions (30 points)

1. You trained a binary classifier model which gives very high accuracy on the training data, but much lower accuracy on validation data. The following may be true: *

   ● This is an instance of overfitting.

   ○ This is an instance of underfitting.

   ● The training was not well regularized.

   ● The training and testing examples are sampled from different distributions.

2. Okham in the 14th century is credited to have stated that one should “shave off unnecessary parameters of a model”. Which of the following implement that principle: **

   ● Regularization.

   ○ Maximum likelihood estimation.

   ● Shrinkage.

   ○ Empirical risk minimization.

   ● Feature selection.

3. Good practices to avoid overfitting include: **

   ● Using a two part cost function which includes a regularizer to penalize model complexity.

   ○ Using a good optimizer to minimize error on training data.

   ● Building a structure of nested subsets of models and train learning machines in each subset, starting from the inner subset, and stopping when the cross-validation error starts increasing.

   ○ Discarding 50% of randomly chosen samples.

4. Wrapper methods are hyper-parameter selection methods that:**

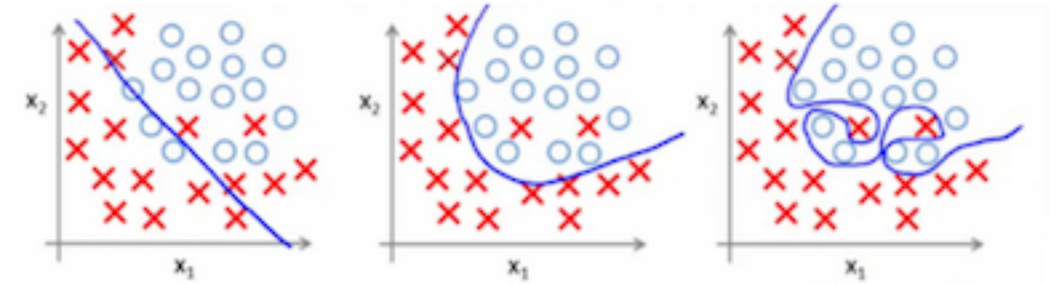
   ○ Should be used whenever possible because they are computationally efficient.

   ○ Should be avoided unless there are no other options because they are always prone to overfitting.

   ● Are useful mainly when the learning machines are "black boxes".

   ○ Should be avoided altogether.

5. Three different classifiers are trained on the same data. Their decision boundaries are shown below. Which of the following statements are true?



   ● The leftmost classifier has high robustness, poor fit.

   ○ The leftmost classifier has poor robustness, high fit.

   ● The rightmost classifier has poor robustness, high fit.

   ○ The rightmost classifier has high robustness, poor fit.

6. What are support vectors: ***

   ○ The examples farthest from the decision boundary.

   ● The only examples necessary to compute f(**x**) in an SVM.

   ○ The class centroids.

   ● All the examples that have a non-zero weight α_k in a SVM.

7. Which of the following can only be used when training data are linearly-separable? *

   - ● Linear hard-margin SVM.
   - ○ Linear Logistic Regression.
   - ○ Linear Soft margin SVM.
   - ○ The centroid method.
   - ○ Parzen windows.

8. The number of test examples needed to get statistically significant results should be: ***

   - ○ Larger if the error rate is larger.
   - ● Larger if the error rate is smaller.
   - ○ It does not matter.

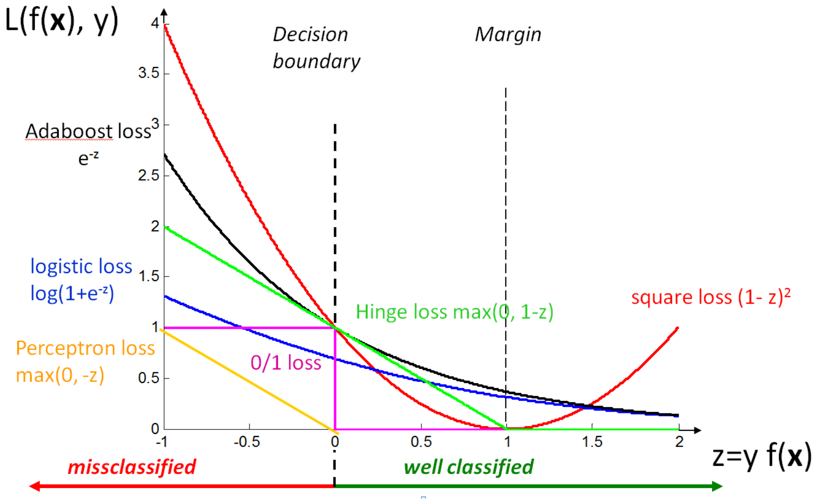
Three descriptive problems

Problem I: Gradient descent (15 points).

Given N training data points $\{(\mathbf{x}^k, y^k)\}$, k=1:N, $\mathbf{x}^k$ in R^d, and labels in y^k in {-1,1}, we seek a linear discriminant function $f(\mathbf{x}) = \mathbf{w}.\mathbf{x}$ optimizing the loss function $L(z) = e^{-z}$, for $z=y\, f(\mathbf{x})$.

Question I.1 (3 points) Is L(z) a large margin loss function? Justify your answer (a graphical justification may be useful).

Answer: Yes. This is because the loss penalizes even examples that are well classified, but the penalty decreases as you go away from the decision boundary.



Question I.2 (4 points) Derive the stochastic gradient descent update $\Delta\mathbf{w}$ for L(z):

Answer: For a learning rate $\eta>0$, and for $z = y\, f(\mathbf{x}) = y \sum_{i=1:d} w_i x_i$

$$\begin{aligned}\Delta w_i &= - \eta\, \partial L/\partial w_i \\ &= - \eta\, \partial L/\partial z\ \partial z/\partial w_i \\ &= \eta\, e^{-z}\, y\, x_i \\ \Delta \mathbf{w} &= \eta\, e^{-z}\, y\, \mathbf{x}\end{aligned}$$

Question I.3 (3 point) We call $R_{emp}(\mathbf{w}) = \sum_{k=1:N} L(z^k)$, where $z^k = y^k f(\mathbf{x}^k)$, the "empirical risk". Derive the batch gradient update $\Delta\mathbf{w}$ for the empirical risk:

Answer: $\Delta\mathbf{w} = \eta \sum_{k=1:N} \exp(-z^k)\, y^k\, \mathbf{x}^k$

Question I.4 (3 point) Suppose you also want to include a penalty term $\lambda \|\mathbf{w}\|^2$ to the risk functional that you wish to minimize. Derive the batch gradient update for the regularized risk $R_{reg}(\mathbf{w}) = R_{emp}(\mathbf{w}) + \lambda \|\mathbf{w}\|^2$:

Answer: $\Delta\mathbf{w} = \eta \sum_{k=1:N} \exp(-z^k)\, y^k\, \mathbf{x}^k - 2\,\eta\,\lambda\,\mathbf{w}$

or $\mathbf{w} \leftarrow (1 - 2\,\eta\,\lambda\,\mathbf{w}) + \eta \sum_{k=1:N} \exp(-z^k)\, y^k\, \mathbf{x}^k$

Question I.5 (2 point) How do you estimate λ (answer in at most 3 words)?

Answer: By cross-validation.

Problem II. Classification concept review (15 points).

Question II.1. **Centroid method.** Now consider a 2-class classification problem in a 2-dimensional feature space x=[x1, x2] with target variable y=±1. The training data comprises 7 samples as shown in Figure 1 (4 black diamonds for the positive class and 3 white diamonds for the negative class).

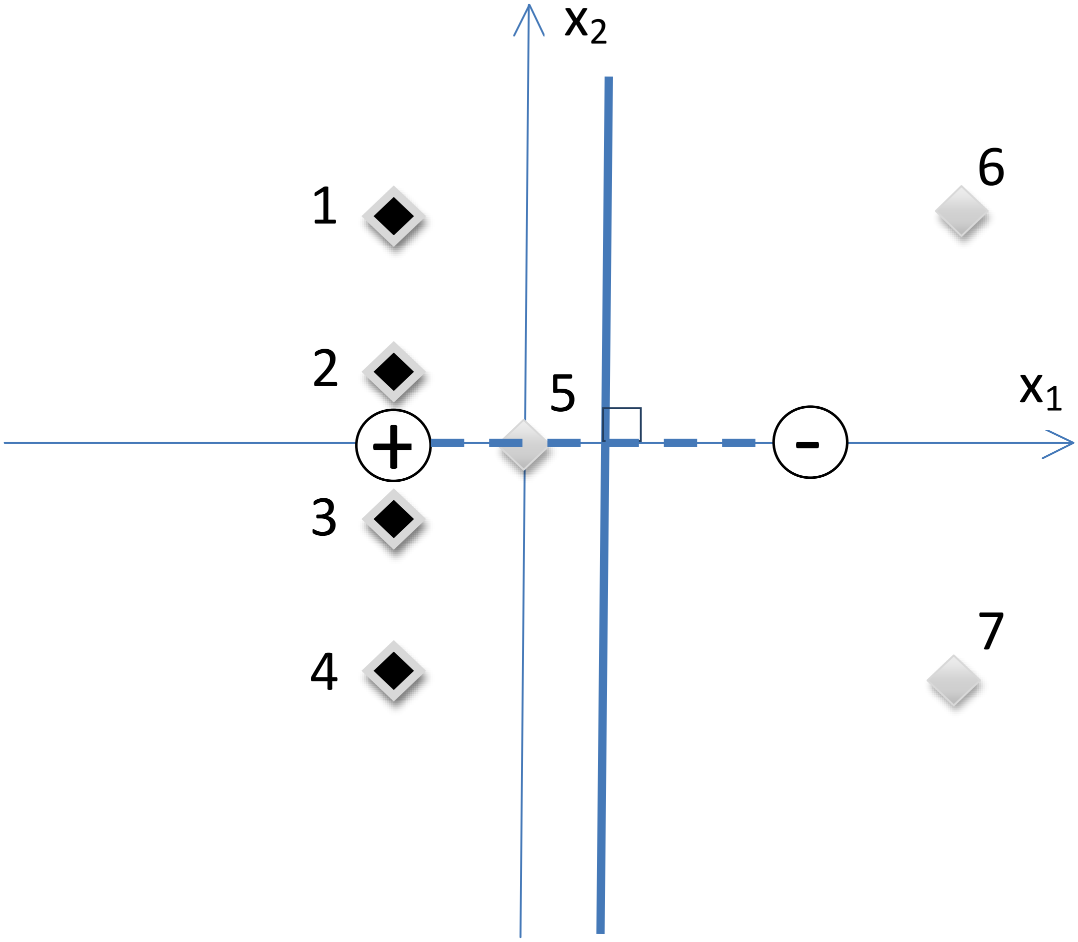


Figure 1: Data for Problem II. **Centroid method** question.

Question II.1.A (2 points): Draw on Figure 1 the **centroids of the two classes** (mark them with a circles (+) for the positive class and a circled (-) for the negative class). **Join the centroids with a thick dashed line**. Draw the **decision boundary** of the centroid method with a **thick solid line.**

Question II.1.B (1 point) What is the **training error rate**? 1/7

Question II.1.C (2 points) Is there any sample such that upon its removal, the decision boundary changes in a manner that the removed sample goes to the other side (Answer "yes" or "no")? NO

Question II.1.D (2 point) What is the **leave-one-out error rate**? 1/7

Question II. 2: **Support Vector Machine (SVM).** Consider again the same training data as in Question II.1, replicated in Figure 2, for your convenience. The "maximum margin classifier" (also called linear "hard margin" SVM) is a classifier that leaves the largest possible margin on either side of the decision boundary. The samples lying on the margin are called support vectors.

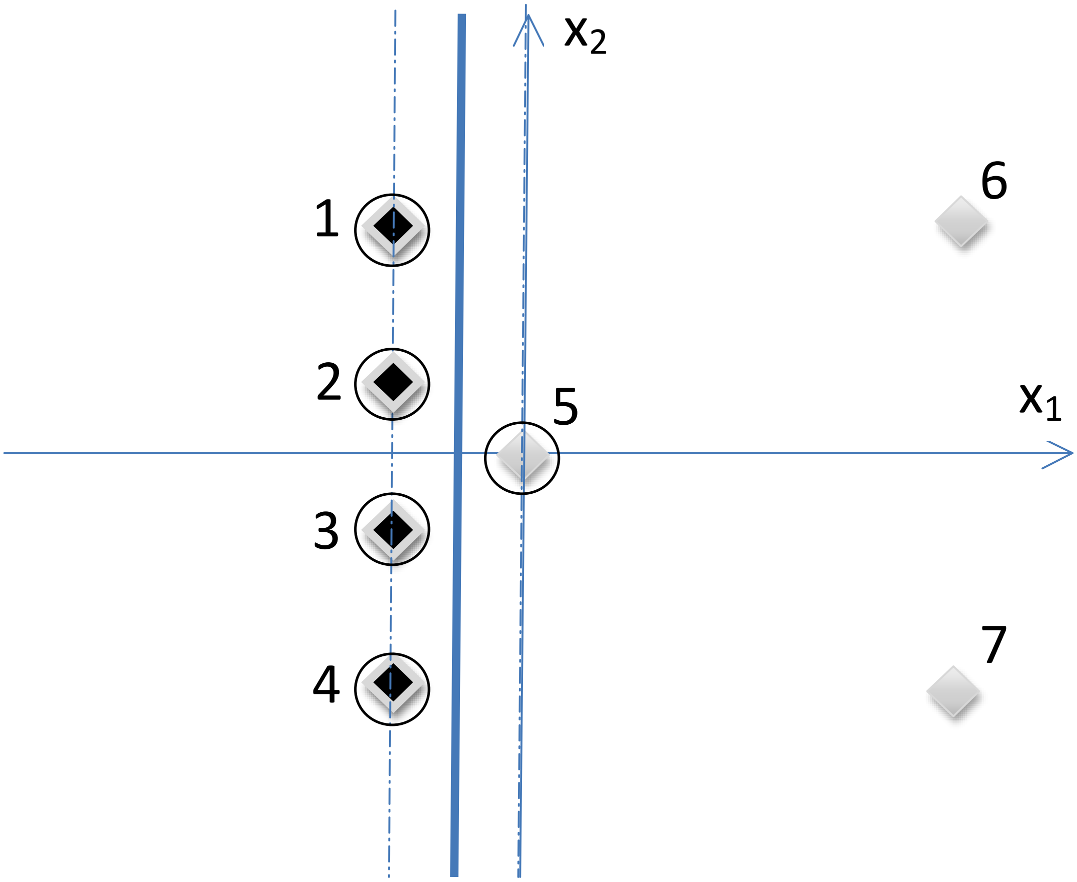


Figure 1: Data for Problem II. **SVM method** question.

Question II.2.A (2 points): Draw on Figure 2 the **decision** boundary obtained by the **linear hard margin SVM** method with a **thick solid line**. Draw the **margins** on either side with **thinner dashed lines**. **Circle the support vectors.**

Question II.2.B (1 points) What is the **training error rate**? Zero.

Question II.2.C (1 point) The removal of which sample will change the decision boundary? Number 5.

Question II.2.D (2 points) What is the leave-one-out error rate? 1/7

Question II.2.E (1 point) A method is more robust if the difference between training error and leave-one-out error is smaller. **Which method (centroid or SVM) is more robust**? The centroid method.

Question II.2.F (1 point) A method has a better fit is it has fewer training error. **Which method has the best fit**? The SVM method.

Problem III. Newton-Raphson for least-square regression (10 points)

In this problem, we will derive an optimization algorithm which we did not study in class, called the ***Newton-Raphson*** algorithm. The algorithm makes updates in a manner that often allows reaching the solution faster than regular gradient descent.

Suppose we start with an initial value of a (1, d **w**; lets call this initial value as $\mathbf{w}^{(0)}$. We know that the first order Taylor apr tion of $\nabla_{\mathbf{w}} R(\mathbf{w}^{(1)})$, at the point $\mathbf{w}^{(0)}$ is:

$$\nabla_{\mathbf{w}} R(\mathbf{w}^{(1)}) = \nabla_{\mathbf{w}} R(\mathbf{w}^{(0)}) + (\mathbf{w}^{(1)} - \mathbf{w}^{(0)})\, \nabla_{\mathbf{w}}^{2} R(\mathbf{w}^{(0)})$$

Question III.1 (3 points). We want to minimize $R(\mathbf{w}^{(1)})$ using this approximation of $\nabla_{\mathbf{w}} R(\mathbf{w}^{(1)})$. Find the update equation for the value of $\mathbf{w}^{(1)}$. This is called the Newton-Raphson update. Notes: This is not a trick question, you just have to solve for $\mathbf{w}^{(1)}$ after equaling $\nabla_{\mathbf{w}}R(\mathbf{w}^{(1)})$ to 0. You can assume that the (d, d) Hessian matrix $\nabla_{\mathbf{w}}{}^2R(\mathbf{w}^{(0)})$ is invertible.

Answer: Let us call $H = \nabla_{\mathbf{w}}{}^2R(\mathbf{w}^{(0)})$ the Hessian matrix.

$\nabla_{\mathbf{w}}R(\mathbf{w}^{(1)}) = 0 \Leftrightarrow \nabla_{\mathbf{w}}R(\mathbf{w}^{(0)}) + (\mathbf{w}^{(1)} - \mathbf{w}^{(0)})\ H = 0$

$\Leftrightarrow (\mathbf{w}^{(1)} - \mathbf{w}^{(0)})\ H = - \nabla_{\mathbf{w}}R(\mathbf{w}^{(0)})$

$\Leftrightarrow \mathbf{w}^{(1)} = \mathbf{w}^{(0)} - \nabla_{\mathbf{w}}R(\mathbf{w}^{(0)})\ H^{-1}$ (if H is invertible)

Question III.2 (4 points). Consider now the linear regression problem: We are given a training data matrix X of dim (N, d) and a target vector **y** of dim(N, 1) and want to find a weight vector **w** of dim (1, d) such that $f(\mathbf{x}) = \mathbf{x}\ \mathbf{w}^T$ approximates **y** best, in the least square sense. The risk functional is: $R(\mathbf{w}) = (X\mathbf{w}^T - \mathbf{y})^T\ (X\mathbf{w}^T - \mathbf{y})$. We will assume that we are in the "regression case" N>d and that the Hessian is invertible. Find the Newton-Raphson update for $\mathbf{w}^{(1)}$.

Answer: $\nabla_{\mathbf{w}}R = 2\ (\mathbf{w}X^TX - \mathbf{y}^TX)$

$\nabla_{\mathbf{w}}{}^2R = H = 2\ X^TX$

$\mathbf{w}^{(1)} = \mathbf{w}^{(0)} - \nabla_{\mathbf{w}}R(\mathbf{w}^{(0)})\ H^{-1}$

$= \mathbf{w}^{(0)} - (\mathbf{w}^{(0)}X^TX - \mathbf{y}^TX)\ (X^TX)^{-1}$

$= - \mathbf{y}^TX(X^TX)^{-1}$

Question III.3 (3 points). Recall the solution to the problem we found in class using the normal equations or the solution found by solving for $\nabla_{\mathbf{w}}R(\mathbf{w}) = 0$ directly. Compare with the solution obtained in question (2). How many iterations of the Newton-Raphson update do we need to perform for linear regression?

Answer: One iteration.

$\nabla_{\mathbf{w}}R = 2\ (\mathbf{w}X^TX - \mathbf{y}^TX) = 0 \Leftrightarrow \mathbf{w}X^TX = \mathbf{y}^TX \Leftrightarrow \mathbf{w} = \mathbf{y}^TX(X^TX)^{-1}$, if $H = X^TX$ is invertible.

Newton-Raphson update: $\mathbf{w}^{(1)} = - \mathbf{y}^TX(X^TX)^{-1}$ identical.