Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science

6.036 Introduction to Machine Learning

Midterm exam (March 17, 2016)

Your name & ID: ______________________

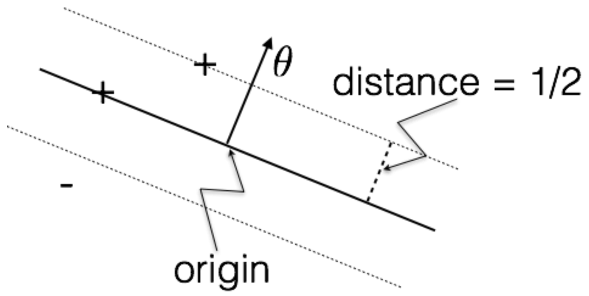
- This is a closed book exam
- You do not need nor are permitted to use calculators
- The value of each question – number of points awarded for full credit – is shown in parenthesis
- The problems are not necessarily in any order of difficulty. We recommend that you read through all the problems first, then do the problems in whatever order suits you best.
- Record all your answers in the places provided

Problem 1	Problem 2	Problem 3	Problem 4	Problem 5	Total
20	7	6	4	23	60

Problem 1

(1.1) **(2 points)** Consider a set of linearly separable examples $(x^{(i)}, y^{(i)})$, $i = 1, \ldots, n$, $x^{(i)} \in \mathcal{R}^d$. If we modify the input vectors by eliminating the first coordinate (e.g., setting it to zero for all $x^{(i)}$), are the resulting set of examples guaranteed to be linearly separable? (Y/N)(   )

(1.2) **(2 points)** Can the perceptron algorithm be viewed as a stochastic gradient descent algorithm, just applied to minimize the zero one loss? (Y/N) (   )



Decision boundary together with the margin boundaries

(1.3) **(6 points)** SVM is an offline algorithm for estimating a linear separator. If we omit the offset parameter, SVM finds θ that minimizes the objective function

$$\left[\frac{1}{n}\sum_{i=1}^{n} \text{Loss}_h(y^{(i)}\theta \cdot x^{(i)})\right] + \frac{\lambda}{2}\|\theta\|^2 \tag{1}$$

Suppose we set $\lambda = 1$ and $n = 3$ as in the figure above. What is the value of the objective function based on the figure?

Based on the figure, is there a solution which has lower loss and smaller $\|\theta\|$?

(1.4) **(2 points)** Pegasos is an on-line stochastic gradient descent (SGD) method for optimizing the SVM objective. Which one of the following update rules is the correct Pegasos update in response to a training example (x, y)? Mark only the correct one or leave blank if all are incorrect.

$$(\quad) \qquad \hat{\theta} = \theta + \eta(yx - \lambda\theta) \tag{2}$$
$$(\quad) \qquad \hat{\theta} = \theta + \eta yx[\![\, 1 - y\theta \cdot x \geq 0 \,]\!] \tag{3}$$
$$(\quad) \qquad \hat{\theta} = \theta + \eta\big(yx[\![\, 1 - y\theta \cdot x \geq 0 \,]\!] - \lambda\theta\big) \tag{4}$$
$$(\quad) \qquad \hat{\theta} = \theta + \eta\big(yx[\![\, 1 - y\theta \cdot x \geq 0 \,]\!]/n - \lambda\theta\big) \tag{5}$$

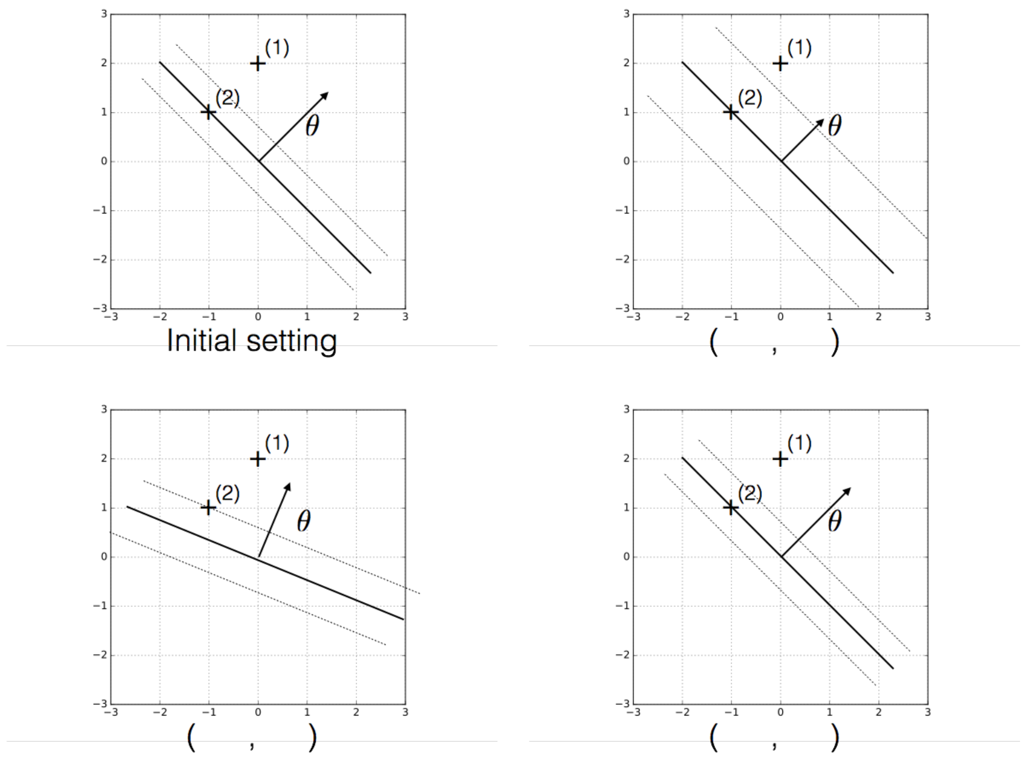
(1.5) **(2 points)** Passive-Aggressive algorithm is more akin to the perceptron algorithm, though it operates with Hinge loss rather than mistakes. Its update rule in response to a training example (x, y) is simply $\hat{\theta} = \theta + \eta yx$ where η comes from minimizing

$$\text{Loss}_h(y\hat{\theta} \cdot x) + \frac{\lambda}{2}\|\hat{\theta} - \theta\|^2 \tag{6}$$

with respect to $\hat{\theta}$. We will use θ as the previous setting of the parameters and $\hat{\theta}$ as the new setting after updating in response to (x, y). Which of the following statements are correct?

$$(\quad) \qquad y\hat{\theta} \cdot x > 1 \text{ if and only if } y\theta \cdot x > 1 \tag{7}$$
$$(\quad) \qquad y\hat{\theta} \cdot x = 1 \text{ if } y\theta \cdot x < 1 \tag{8}$$
$$(\quad) \qquad \text{Loss}_h(y\hat{\theta} \cdot x) \leq \text{Loss}_h(y\theta \cdot x) \text{ always} \tag{9}$$

(1.6) **(6 points)** The top left plot in the figure below shows the initial parameter vector θ along with the associated margin boundaries. We can update the parameters based on either of the two positive points, (1) or (2), and by using either the Passive-Aggressive or the Pegasos algorithm. Please assign the remaining plots to the most plausible combination of (algorithm,point) that was used to generate the figure. You should mark each of the three figures as one of (PA,1), (PA,2), (Peg,1), or (Peg,2).



Problem 2 Given training samples $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$, ridge regression seeks to predict each response $y^{(i)}$ with a linear model $\theta \cdot x^{(i)}$ while encouraging θ to have a small norm. We omit the offset parameter for simplicity. Specifically, θ is estimated by minimizing

$$\left[\frac{1}{n}\sum_{i=1}^{n}(y^{(i)} - \theta \cdot x^{(i)})^2/2\right] + \frac{\lambda}{2}\|\theta\|^2, \tag{10}$$

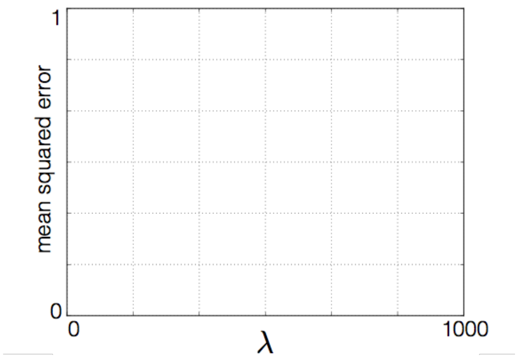
where $\lambda \geq 0$ is the regularization parameter, typically chosen in advance.

(2.1) **(2 points)** What is the solution $\hat{\theta}$ that minimizes Eq(10) if $\lambda \to \infty$?

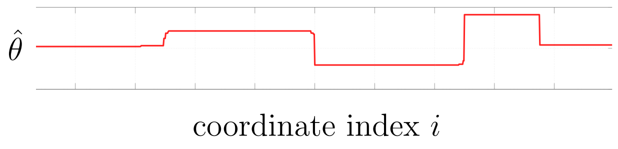
(2.2) **(2 points)** If we assume that $(1/n)\sum_{i=1}^{n}(y^{(i)})^2/2 = 1$, sketch in the figure how the squared training error

$$\frac{1}{n}\sum_{i=1}^{n}(y^{(i)} - \hat{\theta}(\lambda)\cdot x^{(i)})^2/2 \tag{11}$$

behaves as we vary λ. Here $\hat{\theta}(\lambda)$ is the solution to Eq(10) with the chosen λ.



(2.3) **(3 points)** We can bias the estimated linear model in different ways by choosing other kinds of regularization terms. For example, we can replace $\|\theta\|^2 = \sum_{i=1}^{d}\theta_i^2$ with more interesting functions of θ. If we have lots of coordinates (d is large), we can view the parameter vector as a curve (coordinate value as a function of coordinate index). If we wanted the estimated parameter vector to look like



which of the following regularizers should we use?

( ) $|\theta_d - \theta_1|$

( ) $\sum_{i=1}^{d-1}|\theta_{i+1} - \theta_i|$

( ) $\sum_{i=1}^{d}(\theta_i - \sum_{j=1}^{d}\theta_j/d)^2$

Problem 3 Recommender problems can be often solved with low-rank matrix factorization UV^T. We will try to do it here in a very simple context. Suppose the partially observed 3x3 rating matrix Y is given by

$$Y = \begin{bmatrix} 5 & ? & 0 \\ 1 & 1 & ? \\ 1 & ? & 6 \end{bmatrix} \tag{12}$$

(3.1) **(4 points)** We are looking for a rank two factorization so U and V are both 3x2 matrices. We will initialize U as:

$$U = \begin{bmatrix} 1 & 4 \\ 2 & 2 \\ 3 & 5 \end{bmatrix} \tag{13}$$

If we keep U fixed, we can evaluate the best fitting V. Let's focus on the second row of V that pertains to the observations on the 2nd column of Y. Call this row vector v. Write down the objective function for v. What is the value of the objective function if $v = [1, 0]$ and $\lambda = 1$?

(3.2) **(2 points)** Suppose, instead, we initialized U to be all zeros. What will the resulting V be after the first iteration? After convergence?

Problem 4

(4.1) **(4 points)** Figure 1 shows the 'two-corners' dataset. It is clear that the two classes (marked with '+' and '∇') are not linearly separable.

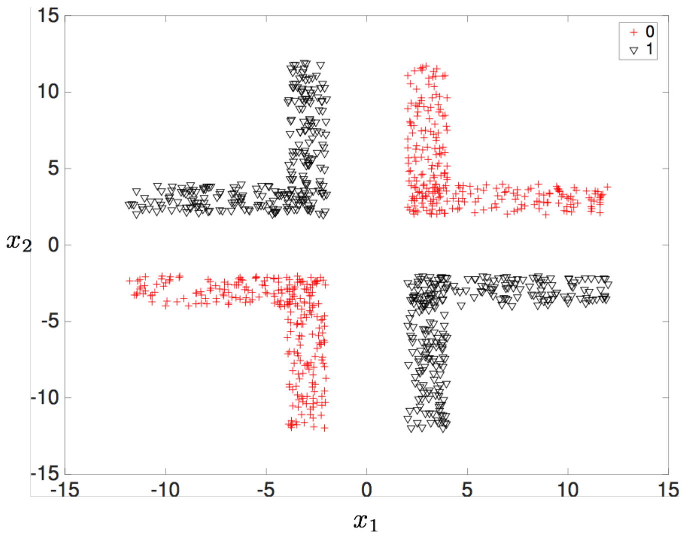


Figure 1: Two-corners dataset

The points labeled '+' live in the first and third quadrants, whereas the '∇' points live in the second and fourth quadrants. Perhaps we can make them separable if we introduce non-linear coordinates. We list a few possible choices below. Mark all the choices that would make this data linearly separable

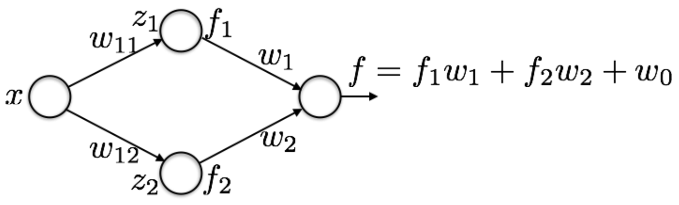
(a) ( ) $[x_1, x_2, x_1x_2]$

(b) ( ) $[x_1^2, x_2^2, \frac{x_1+x_2}{2}]$

(c) ( ) $[x_1, x_2, \tanh(x_1 + x_2)]$

(d) ( ) $[x_1 + x_2, x_1x_2, 1]$

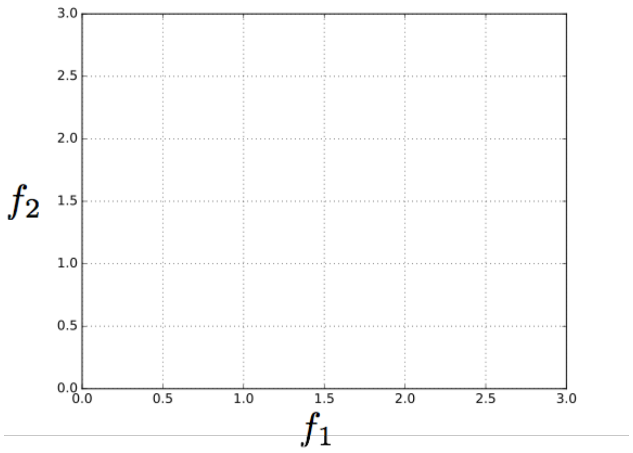
Problem 5 Consider a simple feed-forward neural network model that takes $x \in \mathbb{R}$ as an input and has two ReLU hidden units.



$$\begin{aligned} z_1 &= x\, w_{11} + w_{01} \\ z_2 &= x\, w_{12} + w_{02} \end{aligned}, \quad \begin{bmatrix} w_{11} & w_{01} \\ w_{12} & w_{02} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 2 & -2 \end{bmatrix} \tag{14}$$

(5.1) **(2 points)** When is the output of the first hidden unit, i.e., f_1, exactly zero as a function of $x \in \mathbb{R}$?

(5.2) **(6 points)** Given the parameters in Eq(14), sketch in the figure below how examples $x \in [-2, 2]$ map to the 2-dimensional feature coordinates (f_1, f_2). In other words, map the interval $[-2, 2]$ in the x-space to the 2-dimensional space of hidden unit activations.



(5.3) **(2 points)** Are the training examples $(x = -1, y = -1)$, $(x = 1, y = 1)$, and $(x = 2, y = -1)$ linearly separable in the (f_1, f_2) coordinates? (Y/N) **(** **)**

(5.4) **(4 points)** Suppose $w_1 = 1$, $w_2 = 1$, and $w_0 = 0$. We use $\text{Loss}_h(yf) = \max\{0, 1 - yf\}$ to measure the loss given the network output f and true label y. The parameters will potentially change if we perform a gradient descent step in response to (x, y) where $y = -1$. Provide the range of values of x such that w_{02} *decreases* after the update: $x \in ($ $)$

(5.5) **(3 points)** The architecture and training details are sometimes quite relevant to how well a particular neural network model works. Suppose we use ReLU activation functions for all the hidden units (one layer). Briefly explain what will happen if we initialized all the weights to zero, and all the offset parameters to -1, and ran the stochastic gradient descent method to learn the parameters

(5.6) **(6 points)** We experimented with three different architecture/training combinations:

( ) m hidden units, no regularization

( ) $2m$ hidden units, no regularization

( ) $2m$ hidden units, dropout regularization

Student who carried out the experiments observed that

Model A: training error 0.2, validation error 0.35

Model B: training error 0.25, validation error 0.3

Model C: training error 0.1, validation error 0.4

Please assign the models A,B, and C to their best fitting setups above.

6.036 Official Cheat Sheet

- A (hyper-)plane is a set of points $x \in \mathcal{R}^d$ such that $\theta \cdot x + \theta_0 = 0$. Vector θ is normal to the plane. The signed distance of any point x from the plane is $(\theta \cdot x + \theta_0)/\|\theta\|$. The value of distance is positive on the side where θ points to, and negative on the other side.
- A linear classifier with offset: $h(x; \theta) = \text{sign}(\theta \cdot x + \theta_0)$
- Training error (classification error): $\epsilon_n(h) = \frac{1}{n} \sum_{i=1}^{n} [[y^{(i)} \neq h(x^{(i)})]]$ The same formula can be used to calculate the test error.
- Loss functions:

$$z = y(\theta \cdot x + \theta_0) \text{ (agreement)}$$

$$\text{Loss}_{0,1}(z) = [[z \leq 0]]$$

$$\text{Loss}_{hinge}(z) = \max\{1 - z, 0\}$$

- Passive-aggressive algorithm (no offset): At step k, in response to (x, y), find $\theta^{(k+1)}$ that minimizes $\lambda\|\theta - \theta^{(k)}\|^2/2 + Loss_{hinge}(y\theta \cdot x)$
- SVM: Find a maximum-margin classifier by minimizing $\frac{1}{n} \sum_i \text{Loss}_{hinge}(y^{(i)}(\theta \cdot x^{(i)} + \theta_0)) + \frac{\lambda}{2}||\theta||^2$ which can be done using stochastic gradient descent (Pegasos).
- Linear regression: predict value $\theta \cdot x + \theta_0$ by minimizing $\frac{\lambda}{2}||\theta||^2 + \frac{1}{n} \sum_{i=1}^{n} (y^{(i)} - \theta \cdot x^{(i)} - \theta_0)^2/2$
- Low-rank matrix factorization for collaborative filtering: Minimize $J(U, V) = \sum_{(a,i) \in D} (Y_{ai} - [UV^T]_{ai})^2/2 + \frac{\lambda}{2} \sum_{a=1}^{n} \sum_{j=1}^{k} U_{aj}^2 + \frac{\lambda}{2} \sum_{i=1}^{m} \sum_{j=1}^{k} V_{ij}^2$ Can be solved iteratively by fixing one matrix and using linear regression to find the other.
- Kernels: $K(x, x') = \phi(x) \cdot \phi(x')$

Kernel	form
Linear	$x \cdot x'$
Quadratic	$x \cdot x' + (x \cdot x')^2$
Radial basis	$\exp(-\|\|x - x'\|\|^2/2)$

- Kernel Perceptron (with offset): Cycles through each point t=1,..n and checks if $y^{(t)}(\sum_{i=1}^{n} \alpha_i y^{(i)} [K(x^{(i)}, x^{(t)}) + 1]) \leq 0$. If true, $\alpha_t = \alpha_t + 1$.
- Neural Nets:
  - output of neuron with activation function f is $Z = \sum_{j=1}^{m} f(z_j) V_j + V_0$
  - a unit in layer l with net input $z_j^l = \sum_i f_i^{l-1} w_{ij}^l + w_{0j}^l$ will output $f_j^l = f(z_j^l)$
  - common activation functions include ReLU ($f(z) = \max\{0, z\}$), tanh, and the identity function
  - backpropagation: $\delta_i^{l-1} = f'(z_i^{l-1}) \sum_j w_{ij}^l \delta_j^l$
- Good luck! ⌣̈