Practice problems to prepare for Exam 1

COMS 4771 Spring 2016

Problem 1 (Convex optimization)**.** In this problem, we'll consider an optimization formulation for *linear regression*—this is the supervised learning problem where the output space is $\mathcal{Y} = \mathbb{R}$ (rather than a categorical value like $\mathcal{Y} = \{0, 1\}$). Instead of classification error, the goal here is to learn a function $f \colon \mathbb{R}^d \to \mathbb{R}$ so as to minimize the expected *squared error* with respect to a distribution P over $\mathbb{R}^d \times \mathbb{R}$: $\mathbb{E}[(f(\boldsymbol{X}) - Y)^2]$ (the expectation is taken with respect to the random couple $(\boldsymbol{X}, Y)$ which has distribution P). We consider the case where f is a *linear function* represented by a weight vector $\boldsymbol{w} \in \mathbb{R}^d$. Let S be an i.i.d. sample from P; consider the following optimization problem:

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} \quad \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2 + \frac{1}{|S|} \sum_{(\boldsymbol{x}, y) \in S} (\langle \boldsymbol{w}, \boldsymbol{x} \rangle - y)^2 .$$

(Above, $\lambda > 0$ is assumed to be a positive scalar.)

(a) Show that this optimization problem is a convex optimization problem by computing the Hessian of the objective function and showing that it is positive semidefinite. Hint: you may find the following facts useful: (a) the sum of two positive semidefinite matrices is positive semidefinite and (b) for any vector v, the matrix $vv^\top$ is positive semidefinite.

(b) Derive a gradient descent algorithm for solving the optimization problem (following the recipe from lecture). No need to specify the initial point $\boldsymbol{w}^{(1)}$ nor the step sizes $\eta_t > 0$.

(c) Suppose we add the following constraints to the optimization problem:

$$w_i^2 \;\leq\; 1 \quad \text{for all } i = 1, 2, \ldots, d \,.$$

Is the optimization problem still convex? Briefly explain why or why not.

(d) Same as (c), but with the following constraints instead (assuming d is even):

$$w_i \;=\; w_{i+1} \quad \text{for all } i = 1, 2, \ldots, d-1 \,.$$

(Hint: can you write equality constraints as a pair of inequality constraints?)

(e) Same as (c), but with the following constraints instead:

$$w_i^2 \;=\; 1 \quad \text{for all } i = 1, 2, \ldots, d \,.$$

Problem 2 (Linear classifiers)**.** Which of the following classifiers are *linear classifiers* in $\mathbb{R}^d$?

- The function $f \colon \mathbb{R}^d \to \{0, 1\}$ given by $f(\boldsymbol{x}) = \mathbb{1}\left\{ \frac{1}{1+\exp(-\langle \boldsymbol{w}, \boldsymbol{x} \rangle)} - \frac{1}{2} > 0 \right\}$ for some vector $\boldsymbol{w} \in \mathbb{R}^d$.
- The 1-NN classifier based on the following training data ($d = 2$):

	Feature 1	Feature 2	Label
Example 1	-1	-1	-1
Example 2	$+1$	$+1$	$+1$
Example 3	-1	$+3$	-1
Example 4	$+1$	$+5$	$+1$

- A classifier trained using Kernelized Online Perceptron with the kernel $K(\boldsymbol{x}, \tilde{\boldsymbol{x}}) := \langle \boldsymbol{x}, \boldsymbol{A}\tilde{\boldsymbol{x}} \rangle$, where $\boldsymbol{A}$ is a $d \times d$ symmetric positive definite matrix.

Problem 3 (Maximum likelihood estimation)**.**

- Consider the model $\mathcal{P}$ of distributions over the positive integers $\mathbb{N}$ with probability mass functions given by

$$P_\theta(x) = (1-\theta)^{x-1}\theta, \quad x \in \mathbb{N}$$

for parameter $\theta \in (0, 1)$ called the *success parameter*.

  (a) Derive a formula for the maximum likelihood estimator for the success parameter given data $\{x_i\}_{i=1}^n$ (treated as an iid sample).

  (b) Let $X \sim P_\theta$ for some $P_\theta \in \mathcal{P}$; here θ is an unknown success parameter. Let $\hat{\theta}$ denote the maximum likelihood estimate of the success parameter given the training data $S = \{X\}$ (well, *datum*). What is the expectation of $\hat{\theta}$? (You may leave your answer in terms of θ and a summation over $\mathbb{N}$ if you like.)

- Consider the model $\mathcal{P}$ of probability distributions over the non-negative reals $\mathbb{R}_+$ with probability densities on the given by

$$p_\lambda(x) = \lambda e^{-\lambda x}, \quad x \in \mathbb{R}_+$$

for parameter $\lambda > 0$ called the *rate parameter*.

  (c) Derive a formula for the maximum likelihood estimator for the rate parameter given data $\{x_i\}_{i=1}^n$ (treated as an iid sample).

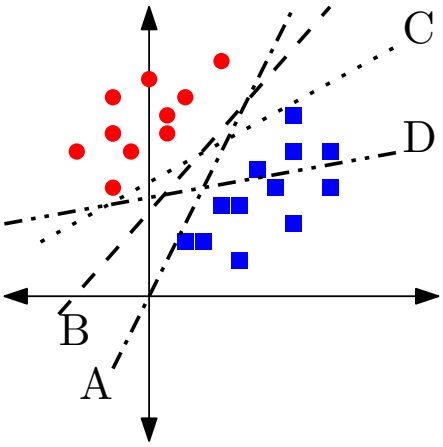
  (d) Let $X \sim p_\lambda$ for some $p_\lambda \in \mathcal{P}$; here λ is an unknown rate parameter. Let $\hat{\lambda}$ denote the maximum likelihood estimate of the rate parameter given the training data $S = \{X\}$ (well, *datum*). What is the expectation of $\hat{\lambda}$? (You may leave your answer in terms of λ and an integral over $\mathbb{R}_+$ if you like.)

Problem 4 (Kernelization)**.** Explain how to "kernelize" *one step* of the gradient descent algorithm from Problem 1(b). That is, assume $K \colon \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is a positive definite kernel with feature map $\boldsymbol{\phi} \colon \mathbb{R}^d \to \mathbb{R}^D$, and assume you have a black box subroutine for computing $K(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ for any $\boldsymbol{x}, \tilde{\boldsymbol{x}} \in \mathbb{R}^d$. Give pseudocode for evaluating $\langle \boldsymbol{w}^{(2)}, \boldsymbol{\phi}(\boldsymbol{z}) \rangle$ for any $\boldsymbol{z} \in \mathbb{R}^d$, where $\boldsymbol{w}^{(2)} \in \mathbb{R}^D$ is the result of one step of the gradient descent algorithm for solving the optimization problem

$$\min_{\boldsymbol{w} \in \mathbb{R}^D} \quad \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2 + \frac{1}{|S|}\sum_{(\boldsymbol{x}, y) \in S} \left(\langle \boldsymbol{w}, \boldsymbol{\phi}(\boldsymbol{x}) \rangle - y\right)^2.$$

Assume that $\boldsymbol{w}^{(1)} = \boldsymbol{0}$ and $\eta_t \equiv \eta$ for some fixed $\eta \in (0, 1/\lambda)$.

Problem 5 (Linear classifiers, again)**.** The figure below depicts the decision boundaries of four linear classifiers (labeled A, B, C, D), and the locations of some labeled training data (positive points are squares, negative points are circles).



Consider the following algorithms for learning linear classifiers which could have produced the depicted classifiers given the depicted training data:

1. (Batch) Perceptron
2. Online Perceptron (making one pass over the training data)
3. ERM for homogeneous linear classifiers
4. An algorithm that exactly solves the SVM problem

What is the most likely correspondence between these algorithms and the depicted linear classifiers? Explain your matching.