Homework 3

571: Probabilistic Machine Learning

Due: Feb14

1 Separability

Given a set of points $\{\mathbf{x}_n\}$, we define the *convex hull* to be the set of all points $\mathbf{x}$ given by

$$\mathbf{x} = \sum_n \alpha_n \mathbf{x}_n$$

where $\alpha_n \leq 0$ and $\sum_n \alpha_n = 1$. Consider a second set of points $\{\mathbf{x}'_m\}$ together with their corresponding convex hull. By definition, the two sets of points will be linearly separable if there exists a vector $\mathbf{w}$ and a scalar w_0 such that $\mathbf{w}^\top \mathbf{x}_n + w_0 > 0$ for all $\mathbf{x}_n$, and $\mathbf{w}^\top \mathbf{x}'_m + w_0 < 0$ for all $\mathbf{x}'_m$. Show that if the two sets of points are linearly separable, their convex hulls do not intersect.

2 Logistic regression and gradient descent

(a) Let $\sigma(a) = \frac{1}{1+e^{-a}}$ be the logistic sigmoid function. Show that $\sigma'(a) = \sigma(a)\,(1 - \sigma(a))$.

(b) For a training set $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^n$ (with $y^{(i)} \in \{1, 0\}$), the loss function (commonly referred to as the *cross entropy* loss) for logistic regression is

$$L_{\mathbf{w}}(\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^n) = \sum_{i=1}^n \{-y^{(i)} \log[h_{\mathbf{w}}(\mathbf{x}^{(i)})] - (1 - y^{(i)}) \log[1 - h_{\mathbf{w}}(\mathbf{x}^{(i)})]\}$$

where $h_{\mathbf{w}}(\mathbf{x}) = \sigma(\mathbf{w}^\top \mathbf{x})$. Here $\mathbf{w}$ parametrizes the model. As we have seen in lecture, this loss is the negated *log-likelihood.* Compute

$$\frac{\partial L_{\mathbf{w}}(\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^n)}{\partial \mathbf{w}_j}.$$

(c) Show that the cross entropy loss of logistic regression is convex.

(d) As we see from (c), the loss function of logistic regression is convex and thus can naturally be optimized using gradient descent. Next we implement gradient descent for logistic regression. Please go through the skeleton code in *logistic-regression-2d.ipynb* and fill in the holes in the code (marked with "YOUR CODE HERE"). Make sure that you pass all check points and answer the questions raised in the skeleton code. The skeleton code is in Python 2. If you are using another language, please make sure it has all corresponding functions and can handle all tasks specified in the skeleton code. It is highly recommended to just follow the notebook provided.

3 Boosting

(a) Assume that the weak learning assumption holds (on the training set), show that the boosted model eventually classifies the training set perfectly (i.e. training error equals to zero).
Hint: look at the class notes.

(b) Suppose that the points are weighted differently in the training set. More specifically, the training set is $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ where each point $(\mathbf{x}_i, y_i)$ in the training set has weight w_i and the objective is defined as

$$R^{train}(\boldsymbol{\lambda}) = \sum_{i=1}^{n} w_i e^{-(\boldsymbol{M}\boldsymbol{\lambda})_i}$$

Derive the weighted version of AdaBoost based on this new objective.
Hint: the change to AdaBoost is surprisingly small.

(c) In this problem, you will implement your own boosted decision stumps (trees with two leafs). Follow the skeleton code in *adaboost-3c.ipynb* and fill in the holes in the code (marked with "YOUR CODE HERE"). Make sure that you pass all check points and answer the questions raised in the skeleton code. The skeleton code is in Python 2. If you are using another language, please make sure it has all corresponding functions and can handle all tasks specified in the skeleton code. It is highly recommended to just follow the notebook provided.