ELEG/CISC 867 Advanced Machine Learning

Homework 2 University of Delaware
Handout: April 23, 2019 Due: May 2, 2019

Problem 1 (Polynomial Regression; 10 pts). Consider a one-dimensional m-degree polynomial regression learning task, where $\mathcal{X} = \mathbb{R}$, $\mathcal{Y} = \mathbb{R}$, and the hypothesis class $\mathcal{H}_{m\text{-poly}}$ is the set of all m-degree polynomials

$$\mathcal{H}_{m\text{-poly}} = \left\{ p_{\mathbf{a}}(x) : \mathbf{a} = (a_0, a_1, \ldots, a_m) \in \mathbb{R}^{m+1} \right\},$$

where

$$p_{\mathbf{a}}(x) = a_0 + a_1 x + \cdots + a_m x^m.$$

Show that one can learn the class $\mathcal{H}_{m\text{-poly}}$ by reducing the problem to an $m+1$-dimensional linear regression problem. In particular, for this new linear regression problem, what are the domain set $\bar{\mathcal{X}}$, target set $\bar{\mathcal{Y}}$, hypothesis class $\bar{\mathcal{H}}$, and training data $\bar{z}^n = \{(\bar{\mathbf{x}}_i, \bar{y}_i)\}_{i=1}^n$? [Hint: Every hypothesis in $\mathcal{H}_{m\text{-poly}}$ can be written as a generalized linear regression predictor.]

Problem 2 (Logistic Regression; 15 pts). Let $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\| \leq r\}$ and $\mathcal{Y} = \{\pm 1\}$. Let $\mathcal{H} = \{\mathbf{w} \in \mathbb{R}^d : \|\mathbf{w}\| \leq B\}$ and let the loss function be $\ell(\mathbf{w}, (\mathbf{x}, y)) = \log(1 + e^{-y\mathbf{w}^T\mathbf{x}})$. This corresponds to a logistic regression problem with the log(istic) loss, where we assume that the instances are in a ball of radius r and we restrict the hypothesis to be $h_{\mathbf{w}}(\mathbf{x}) = \frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}}$ where the norm of $\mathbf{w}$ is bounded by B. Follow the steps below to show that this learning problem is convex-Lipschitz/smooth-bounded.

(a) Demonstrate that $\mathcal{H}$ is a convex set and $\ell(\mathbf{w}, (\mathbf{x}, y))$ is a convex function in $\mathbf{w}$ for any data example $(\mathbf{x}, y)$. This allows us to conclude that the problem is a convex learning problem. [5 pts]

(b) Show that $\ell(\mathbf{w}, (\mathbf{x}, y))$ is r-Lipschitz for any $(\mathbf{x}, y)$. This combined with the fact that $\|\mathbf{w}\| \leq B$ implies that the problem is a convex-Lipschitz-bounded learning problem with parameter r, B. [5 pts]

(c) Show that $\ell(\mathbf{w}, (\mathbf{x}, y))$ is $r^2/4$-smooth and nonnegative for any $(\mathbf{x}, y)$. This combined with the fact that $\|\mathbf{w}\| \leq B$ implies that the problem is a convex-smooth-bounded learning problem with parameter $r^2/4, B$. [5 pts]

Problem 3 (Surrogate Loss Function; 30 pts). In many learning tasks, the natural loss function is not convex and hence implementing the ERM rule is hard. To circumvent this hardness, one popular approach is to replace the nonconvex loss function by a surrogate loss function which i) is convex and ii) upper bounds the original loss. We now demonstrate this concept in the context of learning halfspaces with 0-1 loss. In particular, let $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Y} = \{\pm 1\}$. Let the hypothesis class be the set of homogenous halfspaces, i.e.

$$\mathcal{H} = \{\mathbf{x} \mapsto \operatorname{sgn}(\mathbf{w}^T\mathbf{x}) : \mathbf{w} \in \mathbb{R}^d\}$$

and let the loss be the 0-1 loss given by

$$\ell_{0-1}(\mathbf{w}, (\mathbf{x}, y)) = \mathbb{I}(\operatorname{sgn}(\mathbf{w}^T\mathbf{x}) \neq y).$$

(a) Check that the 0-1 loss can be equivalently expressed as

$$\ell_{0-1}(\mathbf{w}, (\mathbf{x}, y)) = \mathbb{I}(y\mathbf{w}^T\mathbf{x} \leq 0). \quad \text{[5 pts]}$$

(b) Define the hinge loss function as

$$\ell_{\text{hinge}}(\mathbf{w}, (\mathbf{x}, y)) = \max\{0, 1 - y\mathbf{w}^T\mathbf{x}\}.$$

Plot the 0-1 loss and the hinge loss with respect to $y\mathbf{w}^T\mathbf{x}$. [5 pts]

(c) Demonstrate that the hinge loss i) is convex, and ii) upper bounds the 0-1 loss, i.e. $\ell_{0-1}(\mathbf{w}, (\mathbf{x}, y)) \leq \ell_{\text{hinge}}(\mathbf{w}, (\mathbf{x}, y))$ for any $\mathbf{w}$ and $(\mathbf{x}, y)$; therefore, the hinge loss is a convex surrogate for the 0-1 loss. [10 pts]

(d) Show that if we learn the problem with respect to the surrogate loss function ℓ_{hinge}, the error of a hypothesis $\hat{\mathbf{w}}$ can be decomposed as

$$L_{0-1}(\hat{\mathbf{w}}, P) \leq \underbrace{\min_{\mathbf{w}\in\mathcal{H}} L_{0-1}(\mathbf{w}, P)}_{L_{\text{app}}} + \underbrace{\min_{\mathbf{w}\in\mathcal{H}} L_{\text{hinge}}(\mathbf{w}, P) - \min_{\mathbf{w}\in\mathcal{H}} L_{0-1}(\mathbf{w}, P)}_{L_{\text{opt}}}$$
$$+ \underbrace{L_{\text{hinge}}(\hat{\mathbf{w}}, P) - \min_{\mathbf{w}\in\mathcal{H}} L_{\text{hinge}}(\mathbf{w}, P)}_{L_{\text{est}}}.$$

Here, L_{app} is the approximation error that measures how well the hypothesis class $\mathcal{H}$ performs under the 0-1 loss, L_{est} is the estimation error that measures the difference between the smallest error achievable within $\mathcal{H}$ and the error associated with $\hat{\mathbf{w}}$ under the hinge loss, and L_{opt} is the optimization error that measures the difference between the approximation error with respect to the surrogate hinge loss and the approximation error with respect to the original 0-1 loss. [10 pts]

Problem 4 (Boosting; 15 pts). Recall that the AdaBoost algorithm at each iteration updates the weighting distribution on the training data in such a way to "force" the weak learner to focus on the problematic examples in the next iteration. In this question we will find some rigorous justification for this argument. In particular, show that the error of h_t with respect to the distribution $\mathbf{D}^{(t+1)}$ is exactly $1/2$, i.e.,

$$\sum_{i=1}^{n} \mathbf{D}^{(t+1)}(i) \cdot \mathbb{I}(h_t(x_i) \neq y_i) = \frac{1}{2}, \ \forall t \in [1:T].$$

Problem 5 (Stochastic Gradient Descent; 40 pts). Consider a logistic regression problem, where $\mathcal{X} = \mathbb{R}^2$ and $\mathcal{Y} = \{\pm 1\}$, the hypothesis class $\mathcal{H} = \{\mathbf{w} : \mathbf{w} \in \mathbb{R}^2\}$, and the loss function is given by $\ell(\mathbf{w}, (\mathbf{x}, y)) = \log(1 + e^{-y\mathbf{w}^T\mathbf{x}})$. Also assume that the data generating distribution $p(\mathbf{x}, y)$ is given by

$$p(\mathbf{x}, y) = p(y)p(\mathbf{x}|y)$$

in which $p(y) = 0.5$ for any $y \in \{\pm 1\}$ and $p(\mathbf{x}|y)$ is a Gaussian distribution whose mean is $y\boldsymbol{\mu}$ and covariance matrix is the identity matrix I. Write a program (in Python, Matlab, or R) to do the following few steps; submit the results along with the source code.

(a) Assume $\boldsymbol{\mu} = (4, 4)$. Generate 1,000 i.i.d. pairs of $(\mathbf{X}, Y)$ according to the above described distribution $p(\mathbf{x}, y)$; this constitutes the training data $Z^n = \{(\mathbf{X}_i, Y_i)\}_{i=1}^n$, where $n = 1000$. [10 pts]

(b) Use SGD to learn the logistic regression model. Plot the training error vs. the number of iterations. [You might want to try various step sizes

$$\eta \in \{\ldots, 0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1, \ldots, \}$$

to obtain a good plot.] [10 pts]

(c) Now generate a test dataset of 2,000 points. Use the trained model to compute the test error, i.e. the empirical risk on the test dataset. [10 pts]

(d) Repeat parts (a)–(c) with $\boldsymbol{\mu} = (0.5, 0.5)$. How have the training error and test error been changed? Why? [10 pts]