CS 189: Introduction to Machine Learning

Spring 2018

•

Homework 11

Due on Friday, April 13th, 2018 at 10pm

•

Solutions by

Firstname Lastname

StudentID

In collaboration with

None

Problem 1: Getting Started

Read through this page carefully. You may typeset your homework in latex or submit neatly handwritten/scanned solutions. Please start each question on a new page. Deliverables:

1. Submit a PDF of your writeup, **with an appendix for your code**, to assignment on Gradescope, "HW11 Write-Up". If there are graphs, include those graphs in the correct sections. Do not simply reference your appendix.
2. If there is code, submit all code needed to reproduce your results, "HW11 Code".
3. If there is a test set, submit your test set evaluation results, "HW11 Test Set".

After you've submitted your homework, watch out for the self-grade form.

a. Who else did you you work with on this homework? In case of course events, just describe the group. How did you work on this homework? Any comments about the homework?

Solution

b. Please copy the following statement and sign next to it. We just want to make it *extra* clear so that no one inadvertently cheats.

I certify that all solutions are entirely in my words and that I have not looked at another student's solutions. I have credited all external sources in this write up.

Solution

Problem 2: SVM with custom margins

In the lecture, we covered the soft margin SVM. The objective to be optimized over the training set $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$ is

$$\min_{\mathbf{w},b,\xi_i} \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i \tag{1}$$

$$s.t. \quad y_i(\mathbf{w}^\top \mathbf{x}_i - b) \geqslant 1 - \xi_i \quad \forall i \tag{2}$$

$$\xi_i \geqslant 0 \quad \forall i \tag{3}$$

In this problem, we are interested in a modified version of the soft margin SVM where we have a custom margin for each of the n data points. In the standard soft margin SVM, we pay a penalty of ξ_i for each of the data point. In practice, we might not want to treat each training point equally, since with prior knowledge, we might know that some data points are more important than the others. There is some connection to weighted least squares. We formally define the following optimization problem:

$$\min_{\mathbf{w},b,\xi_i} \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\phi_i\xi_i \tag{4}$$

$$s.t. \quad y_i(\mathbf{w}^\top \mathbf{x}_i - b) \geqslant 1 - \xi_i \quad \forall i \tag{5}$$

$$\xi_i \geqslant 0 \quad \forall i \tag{6}$$

Note that the only difference is that we have a weighting factor $\phi_i > 0$ for each of the slack variables ξ_i in the objective function. ϕ_i are some constants given by the prior knowledge, thus they can be treated as known constants in the optimization problem. Intuitively, this formulation weights each of the violations (ξ_i) differently according to the prior knowledge (ϕ_i).

a. For the standard soft margin SVM, we have shown that the constrained optimization problem is equal to the following unconstrained optimization problem, i.e. regularized empirical risk minimization problem with hinge loss:

$$\min_{\mathbf{w},b} \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\max(1 - y_i(\mathbf{w}^\top \mathbf{x}_i - b), 0) \tag{7}$$

What's the corresponding unconstrained optimization problem for the SVM with custom margins?

Solution

b. The dual of the standard soft margin SVM is:

$$\max_{\alpha} \alpha^\top \mathbf{1} - \frac{1}{2}\alpha^\top \mathbf{Q}\alpha \tag{8}$$

$$s.t. \sum_{i=1}^{n}\alpha_i y_i = 0 \tag{9}$$

$$0 \leqslant \alpha_i \leqslant C \quad i = 1, \cdots, n \tag{10}$$

where $\mathbf{Q} = (\operatorname{diag}\mathbf{y})\mathbf{X}\mathbf{X}^T(\operatorname{diag}\mathbf{y})$

What's the dual form of the SVM with custom margin? Show the derivation steps in detail.

Solution

c. **From the dual formulation above, how would you kernelize the SVM with custom margins? What role does the ϕ_i play in the kernelized version?**

Solution

Problem 3: Nearest Neighbors, from A to Z

For this problem, we will use data from the UN to have some fun with the nearest neighbors approach to learning. A lot of the code you will need has been provided for you.

The data we are using is called the "World Values Survey." It consists of survey data collection over several years from almost all countries. The survey asked "Which of these are most important for you and your family?" There were 16 possible responses, including needs like "Freedom from Discrimination and Persecution" and "Better Transport and Roads." The data reported is the fraction of responses in each country that chose each option.

We would like to use these 16 features of each country (the citizen's responses to the survey) to predict that country's HDI (Human Development Index). In reality, the HDI is a complex measure which takes into account lots of data about a country, including factors like life expectancy, education, per capita income, etc. Intuitively though, you might expect citizens of countries with different HDI to have different priorities. For that reason, predicting the HDI from survey data might be a reasonable endeavor.

Note that throughout the problem we will be using RMSE, which stands for Root Mean Squared Error.

a. (Bonus): **Fill out the "Berkeley's S2018 Values Survey."** The purpose of this is so that you have a sense of how the data was generated, a useful first step in any ML problem. Just for fun, at the end of this problem we will attempt to predict what the HDI of Berkeley would be if it were its own country.

Solution

b. First, we should do some basic data exploration. **Compute the correlation of each feature with HDI. Which feature is the most positively correlated with HDI? Which feature is the most negatively correlated with HDI? Which feature is the least correlated with HDI (closest to 0)?**

Solution

c. **For each of these three features identified in b (most positively correlated, most negatively correlated, least correlated), plot "HDI versus [Feature]."** You will create three plots in total. **What do you observe?**

Solution

d. Let's visualize the data a bit more. **Plot the data in its first two PCA dimensions, colored by HDI.** The code to do this has been provided for you.

Solution

e. Now, let's use our first ML technique. **Use the code provided to train and cross-validate ridge regression to predict a country's HDI from its citizens' world values survey responses. What is the best RMSE?**

Solution

f. Let's try another ML technique. **Use the code provided to train and cross-validate LASSO regression to predict a country's HDI from its citizens' world values survey responses. What is the best RMSE?**

Solution

g. **Examine the model returned by LASSO regression (that is, the 16 feature weights). Does LASSO regression indeed give more 0 weights?**

Solution

h. In lecture, we covered k-Nearest Neighbors for classification problems. We decided that the class of a test point would be the plurality of the classes of the k nearest training points. That algorithm makes sense when the outputs are discrete, so we can vote. Here, the outputs are continuous. **How would you adapt the k Nearest Neighbors algorithm for a regression problem?**

Solution

i. **Which countries are the 7 nearest neighbors of the USA (in order)?**

Solution

j. The most important meta-parameter of k nearest neighbors is k itself. **Plot the RMSE of kNN regression versus k, where k is the number of neighbors. What is the best value of k? What is the RMSE?**

Solution

k. **Explain your plot in (j) in terms of bias and variance.** This is tricky, so take some time to think about it. Think about the spirit of bias and variance more than their precise definitions.

Solution

l. We do not need to give every neighbor an equal weight. Maybe closer neighbors are more relevant. For the sake of this problem, let's weight each neighbor by the inverse of its distance to the test point. **Plot the RMSE of kNN regression with distance weighting versus k, where k is the number of features. What is the best value of k? What is the RMSE?**

Solution

m. One of the challenges of k Nearest Neighbors is that it is very sensitive to the scale of the features. For example, if one feature takes on values 0 or 0.1 and another takes on values 0 or 10, then the nearest neighbors approach will almost certainly pick nearest neighbors according to the second feature. **Which countries are the 7 nearest neighbors of the USA after scaling (in order)? Compare your result to i.**

Solution

n. **Add scaling to your k nearest neighbors pipeline (continue to use distance weighting). Plot RMSE versus k. What is the best value for k? What is the RMSE?**

Solution

o. (Bonus): **Rather than scaling each feature to have unit variance, explore ways of scaling the features non-uniformly. How much does this help, if at all?**

Solution

p. You have been given a set of test features: countries where the responses to the world values survey are given but the HDI is not known. **Using the best model developed so far, predict the HDI values of the countries in the test set. Submit your predictions on Gradescope.**

Solution

q. So far we have dealt with the regression problem. Let's take a brief look at classification. A naive classifier is a classifier which disregards the features and just classifies everything as belonging to a single class. **In any classification problem with k classes, at least what accuracy are we guaranteed to get with the best naive classifier?** (Hint: there are k possible naive classifiers. Use the pigeonhole principle).

Solution

r. We will split countries into two groups: high HDI (more than 0.7) and low HDI (less than 0.7). **Plot the countries by their first two PCA dimensions again, but now color them by class.**

Solution

s. Examine the graph generated in (r). **How well do you think a linear SVM would do in classification?**

Solution

t. We will use an SVM classifier to predict whether a country's HDI is "high" or "low" based on the responses of their citizens to the World Values Survey. **Use the code provided to train and cross-validate an SVM classifier using a linear kernel. What is the accuracy of the classifier?**

Solution

u. We are going to modify the classifier from (t). **Add a PCA step and Scaling step to the SVM pipeline. Your hyper-parameter search should now be over all possible dimensions for the PCA reduction. Does the accuracy improve?**

Solution

v. Change the kernel in t from linear to "radial basis function" (rbf). For this part, do not use PCA or Scaling. **What is the accuracy?**

Solution

w. Now we are going to use k Nearest Neighbors for the same task. That is, we would like to predict whether a country's HDI is "high" or "low" based on the responses of their citizens to the World Values Survey. **Train and cross-validate a k Nearest Neighbors classifier using distance weighting. What is its accuracy? Does scaling help?**

Solution

x. (Bonus): Towards the end of the week, we will post the "Berkeley's S2018 Values Survey." **If this course were an independent country, what do you predict its HDI would be?**

Solution

y. (Bonus): **Describe how you would use kNN to revisit the sensor location problem from previous homework. How well do you think it will work?**

Solution

z. (Bonus): **What did you learn from this problem? Do you have any useful feedback for the problem author?**

Solution

Problem 4: Stability: A Unified Approach to Generalization for Classification and Regression

In this problem, we will study how well machine learning algorithms can generalize from the dataset they have been trained on (the training set) to an unseen dataset (the test set). Assume all data is generated from some distribution $\mathcal{D}$ and we sample a training set $S \sim \mathcal{D}^n$ where $\mathcal{D}^n$ is the distribution of n datapoints drawn i.i.d. from $\mathcal{D}$. In this problem we consider estimators of the form

$$\mathbf{w}(S) = \arg\min_{\mathbf{w}} L_S(\mathbf{w}) + \lambda \|\mathbf{w}\|^2 \tag{11}$$

where $L_S(\mathbf{w})$ is the empirical loss function

$$L_S(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{w}, z_i) \qquad \text{where } z_i = (\mathbf{x}_i, y_i)$$

evaluated on the training data set $S = (z_1, z_2, \ldots, z_n)$. We are interested in the loss

$$L_{\mathcal{D}}(\mathbf{w}) = \mathbb{E}_{z\sim\mathcal{D}}[\ell(\mathbf{w}, z)] \tag{12}$$

on a yet unseen test dataset drawn from $\mathcal{D}$.

We call an estimator ϵ_n-**stable** if

$$\mathbb{E}_{S\sim\mathcal{D}^n, z'\sim\mathcal{D}, i\sim U(n)}[\ell(\mathbf{w}(S^{(i)}(z')), z_i) - \ell(\mathbf{w}(S), z_i)] \leqslant \epsilon_n$$

where $S = (z_1, z_2, \ldots, z_n)$ and $S^{(i)}(z') = (z_1, z_2, \ldots, z_{i-1}, z', z_{i+1}, \ldots, z_n)$ and $U(n)$ is the uniform distribution on $\{1, \ldots, n\}$.

We will show that if an estimator is ϵ_n-**stable**, then the test error is close to the training error in expectation, namely

$$\mathbb{E}_{S\sim\mathcal{D}^n}[L_{\mathcal{D}}(\mathbf{w}(S))] \leqslant \mathbb{E}_{S\sim\mathcal{D}^n}[L_S(\mathbf{w}(S))] + \epsilon_n \tag{13}$$

In the first part of the problem, we will establish this result and in the second part, we will apply it to show the generalization properties of ridge regression and soft margin SVMs.

a. We first link stability to generalization via the fundamental property

$$\mathbb{E}_{S\sim\mathcal{D}^n}[L_{\mathcal{D}}(\mathbf{w}(S)) - L_S(\mathbf{w}(S))] = \mathbb{E}_{S\sim\mathcal{D}^n, i\sim U(n), z'\in\mathcal{D}}[\ell(\mathbf{w}(S^{(i)}(z'), z_i) - \ell(\mathbf{w}(S), z_i)] \tag{14}$$

where $S = (z_1, \ldots, z_n)$ be an iid sequence of samples and z' be another iid sample and $U(n)$ be the uniform distribution over $\{1, 2, \ldots, n\}$. **Show equation (14) and conclude that (13) holds if A is ϵ-stable.**

Conceptual hint: When computing the expectation over both training points and a test point, switching any one training point and the test point gives the same result.

Technical hint: Use that $\frac{1}{n}\sum_{i=1}^{n} f(z_i) = \mathbb{E}_{i\sim U(n)} f(z_i)$.

Solution

b. As a simple example, let's walk through the steps of the proof in the simple example where we estimate the mean

$$\mu(S) = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{15}$$

of a dataset $S = (x_1, x_2, \ldots, x_n)$ with scalars $x_1, \ldots, x_n \in \mathbb{R}$. The loss function in this case is

$$\ell(\mu, x_i) = \frac{1}{2}(\mu - x_i)^2$$

and the regularization parameter is $\lambda = 0$. **First, show that**

$$\ell(\mu(S^{(i)}(z')), x_i) - \ell(\mu(S), x_i) \leqslant \frac{C \cdot B \cdot \max_i |x_i|}{n}$$

where $|\mu(S)|, |\mu(S^{(i)}(z'))| \leqslant B$ and C is some numerical constant. **Conclude that the generalization error is bounded by**

$$\mathbb{E}_{S\sim\mathcal{D}^n}[L_{\mathcal{D}}(\mu(S)) - L_S(\mu(S))] \leqslant \frac{C \cdot B \cdot \max_i |x_i|}{n}$$

Solution

c. We now consider **ridge regression**. The loss function is (11) with $\ell(\mathbf{w}, z) = \frac{1}{2}(\mathbf{w}^\top \mathbf{x}_i - y_i)^2$. **First, show that $\ell(\mathbf{w}, z)$ is β-smooth, i. e.**

$$\|\nabla \ell(\mathbf{v}, z) - \nabla \ell(\mathbf{w}, z)\| \leqslant \beta \|\mathbf{v} - \mathbf{w}\|$$

with $\beta = \max_i \|x_i\|$.

Solution

d. **Derive a generalization bound (13) for ridge regression.**

You may use *without proof* that for a β-smooth loss function, we have

$$\mathbb{E}[\ell(\mathbf{w}(S^{(i)}(z')), z_i) - \ell(\mathbf{w}(S), z_i)] \leqslant \frac{C\beta}{\lambda n} \tag{16}$$

for some constant C.

Solution

e. Show that if ℓ is ρ-Lipschitz in the first argument, i.e.

$$\ell(\mathbf{w}(S^{(i)}(z')), z_i) - \ell(\mathbf{w}(S), z_i) \leqslant \rho \|\mathbf{w}(S^{(i)}(z')) - \mathbf{w}(S)\| \tag{17}$$

then the learning algorithm (11) is ϵ-stable with

$$\ell(\mathbf{w}(S^{(i)}(z')), z_i) - \ell(\mathbf{w}(S), z_i) \leqslant \frac{2\rho^2}{\lambda n}.$$

Hint: First show

$$\lambda \|\mathbf{w}(S^{(i)}(z')) - \mathbf{w}(S)\|^2 \leqslant \frac{\ell(\mathbf{w}(S^{(i)}), z_i) - \ell(\mathbf{w}(S), z_i)}{n} + \frac{\ell(\mathbf{w}(S), z') - \ell(\mathbf{w}(S^{(i)}), z')}{n}. \tag{18}$$

Solution

f. We now consider the **soft margin SVMs**. The associated loss function is the hinge loss $\ell(\mathbf{w}, z_i) = \max(0, 1 - y_i \mathbf{w}^\top \mathbf{x}_i)$. **First show that $\ell(\mathbf{w}, z_i)$ is ρ-Lipschitz with $\rho = \max_i \|x_i\|$.**

Solution

g. **Derive a generalization bound (13) for soft margin SVMs.**

Solution

Problem 5: Your Own Question

Write your own question, and provide a thorough solution.

Writing your own problems is a very important way to really learn the material. The famous "Bloom's Taxonomy" that lists the levels of learning is: Remember, Understand, Apply, Analyze, Evaluate, and Create. Using what you know to create is the top-level. We rarely ask you any HW questions about the lowest level of straight-up remembering, expecting you to be able to do that yourself. (e.g. make yourself flashcards) But we don't want the same to be true about the highest level.

As a practical matter, having some practice at trying to create problems helps you study for exams much better than simply counting on solving existing practice problems. This is because thinking about how to create an interesting problem forces you to really look at the material from the perspective of those who are going to create the exams.

Besides, this is fun. If you want to make a boring problem, go ahead. That is your prerogative. But it is more fun to really engage with the material, discover something interesting, and then come up with a problem that walks others down a journey that lets them share your discovery. You don't have to achieve this every week. But unless you try every week, it probably won't happen ever.

Solution

Code Appendix