

CS5470: Assignment-1

Submission Deadline: 16-Nov-2017 (Thursday) 10:30am;
Presentations in lecture: 11am-12pm on the same day.

Note: You may use *any* programming language of your choice and any publicly available software/code from Internet. However, I recommend using `cvx` available at <http://cvxr.com/cvx/>. The only restriction is: do NOT discuss this assignment with your course-mates or share code with them. In case you have a doubt, then approach me directly. It is easy to identify defaulters during the demo.

Let $\mathcal{X} \subset \mathbb{R}^n$, loss be square-loss and

$$\mathcal{F}_W = \{f \mid \exists w \in \mathcal{H}_k, \|w\|_{\mathcal{H}_k} \leq W \ni f(x) = \langle w, \phi_k(x) \rangle_{\mathcal{H}_k}\},$$

where $k(x, z) = e^{-\frac{1}{2}\|x-z\|^2}$ is the (normalized) Gaussian kernel, and $\phi_k, \mathcal{H}_k$ are the kernel induced map and Hilbert space. Assume that the dataset is normalized¹. From our lectures we have the following different algorithms for simultaneously selecting the right $\hat{W}_m$ from the set $\{10^{-3}, 5 * 10^{-3}, 10^{-2}, 5 * 10^{-2}, 10^{-1}, 5 * 10^{-1}, 1, 5, 10, 50, 100, 500, 1000\}$ and the right $\hat{w}_m \ni \|w\| \leq \hat{W}_m$:

1. Guaranteed/Structural Risk Minimization (SRM) + ERM:
 - (a) For each W solve the corresponding ERM and obtain the minimum empirical risk. Also, compute the empirical maximum discrepancy² of the loss-class for each W . Then find the $\hat{W}_m$ that minimizes the

¹Normalize data in each input feature to zero mean and unit variance. After this only, split into training, validation, test etc. (if required).

²Note that, for every W , ERM is a convex program and hence can be solved using `cvx`. However, computing the empirical maximum discrepancy is NOT convex. But, fortunately, it can be solved efficiently. Please read section 3.5.2 in http://www2.isye.gatech.edu/~nemirovs/Lect_ModConvOpt.pdf. The idea is to write down the SDP relaxation, which is exact in this special case, and solve it using `cvx`. Please use the definition of empirical maximum discrepancy as $\hat{D}_m(\mathcal{H}) \equiv \max_{h \in \mathcal{H}} \frac{2}{m} \sum_{i=1}^{\frac{m}{2}} h(x_i, y_i) - \frac{2}{m} \sum_{i=\frac{m}{2}+1}^m h(x_i, y_i)$ as the maximum discrepancy of $\mathcal{H}$ (at m), where $\mathcal{D} = \{(x_1, y_1), \dots, (x_m, y_m)\}$ is the given training set

sum of these two terms. Report $\hat{w}_m$ as the empirical risk minimizer corresponding to $\hat{W}_m$ ³.

- (b) For each W solve the corresponding ERM and obtain the minimum empirical risk. Also, compute the empirical Rademacher average⁴ for each W . Then find the $\hat{W}_m$ that minimizes the sum of the first term and twice the second. Report $\hat{w}_m$ as the empirical risk minimizer corresponding to $\hat{W}_m$.
- (c) For each W solve the corresponding ERM and obtain the minimum empirical risk. Then find the $\hat{W}_m$ that minimizes the sum of this and $2\frac{WR}{\sqrt{m}}$. Report $\hat{w}_m$ as the empirical risk minimizer corresponding to $\hat{W}_m$.
- (d) Solve⁵:

$$(\hat{W}_m, \hat{w}_m) \equiv \operatorname{argmin}_{W \geq 0, \|w\| \leq W} \hat{R}_m[f] + 2\frac{WR}{\sqrt{m}}.$$

2. ERM + ERM:

- (a) Split given training dataset into two equal parts: training and validation. For each W , find empirical risk minimizer using the new training data alone. With each such minimizer, compute error on validation set, pick the best and report the corresponding W as $\hat{W}_m$ and corresponding ERMizer as $\hat{w}_m$.
- (b) Repeat above 100 times with various random splits. Pick $\hat{W}_m, \hat{w}_m$ by looking at sum of all validation errors.
- (c) Pick $\hat{W}_m$ that minimizes 5-fold cross validation error. For definition of k-fold cross validation error, please refer to the Internet.
- (d) Repeat above 100 times with various random splits. Pick $\hat{W}_m, \hat{w}_m$ by looking at sum of all 5-fold cross-validation errors.

Your goal is to compare all these 8 methods in terms of computational effort and accuracy (as measured on an independent test set) for various values of m . More specifically, do the following in case of your dataset:

Split (randomly) your dataset into two equal parts: training and test. Now, consider increasingly bigger subsets of the training set $m = 5, 6, 7, 8, 9, 10, 11, 12, 13, \dots$ as the actual training set for each of the 8 methods. With each m and each of the 8 methods above, note the cpu-time for entire process of choosing $\hat{W}_m, \hat{w}_m$ and also note the accuracy on test set. Now plot a cpu-time vs. test set accuracy plot for each of the method. You may skip plotting an m value and method combination that takes “too long”. You should automate your code so that I can run it on any dataset including the one assigned to you.

³Bound in Theorem 5 (a) in <http://www.jmlr.org/papers/volume3/bartlett02a/bartlett02a.pdf> is the motivation for this algorithm.

⁴For this you will need to solve a maximization problem 2^m times, one for each pattern of σ s. Each of this max. can be again solved using the SDP relaxation, which in this case, is solving a generalized eigen value problem.

⁵This is a convex program and can be solved by cvx.