Problem Set - CS729

February 12, 2012

1. Show that i) hinge-loss is convex and Lipschitz continuous but non-differentiable ii) squared hinge-loss is convex and differentiable but not Lipschitz continuous ii) the truncated hinge-loss is neither convex nor concave; however is Lipschitz continous and not differentiable.

2. Consider the case of binary classification. Suppose that the (usually) unknown joint distribution P(x,y) relating the input x and output y is known. In this case find the Bayes optimal classifier as well as the risk with it. Justify your results using probability theory.

3. Consider a finite set of binary classifiers: $\mathcal{F} = \{f_1, \ldots, f_n\}$, where $f_i : \mathcal{X} \to \{-1, 1\} \; \forall \; i$. Show that the conditional Rademacher average of this function class is upper bounded: $\hat{\mathcal{R}}_m(\mathcal{F}) \leq 2\sqrt{\frac{\log(|\mathcal{F}|)}{m}}$. Now compare the bound (2.4) and (2.7) in lecture notes. Which is better?

4. Starting from (2.5) in lecture notes, obatin a VC-type bound similar to (2.7) that involves the maximum discrepency[1] $\mathcal{M}(\mathcal{F})$ measure of function class complexity rather than the Rademacher average. Needless to say, the VC-type bound should involve quantities which can be computed from the training set alone.

5. Obtain an upper bound on the true risk with the SRM candidate involving the true risk with the ERM candidate, which holds with high probability. Does your upper bound conclusively say that asymptotically SRM is better than ERM? Or perhaps the other way?

6. The following involves simulations using the Parkinsons Dataset available at `archive.ics.uci.edu/ml/datasets/Parkinsons`. The objective is to try out various model selection procedures and access their performance on this dataset. Lets assume the ERM problem is solved using an SVM formalism (The Ivanov regularized version[2]). Also lets (randomly[3]) split

[1] You may use `http://jmlr.csail.mit.edu/papers/volume3/bartlett02a/bartlett02a.pdf` for the definition of Maximum discrepency. However you should not use any results in that paper. In particular you must start from (2.4) and follow similar steps as in lectures and arrive at an analogous expression for (2.7).

[2] Use `cvx` available at `cvxr.com/cvx/` for solving the corresponding SOCP problem.

[3] While splitting make sure the ratio of +ve to -ve examples is maintained.

the dataset into two parts one with 100 examples that are to be used for building the model/model-parameters (i.e. training); while the rest are to be used to evaluate the performance of the model built (i.e. test set). The model selection schemes you need to try are: i) SRM using (2.7) and further bounding by (2.8) ii) leave-one-out cross-validation. Restrict your model selection problem i.e. search for optimal W in the range 10^{-2} to 10^{2} taking 5 values uniformly in log-scale i.e., try $10^{-2}, 10^{-1}, 1, 10, 100$ as values of W. By evaluating the test set accuracy achieved with SRM and leave-one-out decide which one wins. Now take increasing number of samples in the same range of W: 5 (already done), 10, 15, 20. What happens to the test set error? Do you see that SRM does not overfit whereas leave-one-out overfits? Better confidence on your observations may be got by considering various random train-test splits.

7. Given an input space $\mathcal{X}$ with $\|x\| \leq r \; \forall x \in \mathcal{X}$ and the non-homogeneous polynomial kernel $k(x,y) = (1+x^\top y)^d$, show that $\|\phi_k(x)\|_{\mathcal{H}_k} \leq \sqrt{(1+r^2)^d}$. Using this with the radius-margin bound provides a way of arguing that the class of functions induced by non-homogeneous polynomial kernel grows with d. Now can you use a similar argument to show that the Gaussian kernel (any suitable variant) is "bigger" than polynomial kernel?

8. Given a +ve kernel k show that that kernel $k^{'}$ given by: $k^{'}(x,y) = \frac{k(x.y)}{\sqrt{k(x,x)k(y,y)}}$ is also +ve.

9. For any kernel k show that $k(x,y) \leq (k(x,x) + k(y,y))/2$.

10. Solve following problems from Smola's book: 2.10, 2.15, 2.20, 13.2, 13.9.