Leave-one-out approximations

9.520 Class 19, 23 April 2002

Sayan Mukherjee

Plan

- Cross-validation

- Why the leave-one-out estimate is almost unbiased ?

- Generalized approximate cross-validation

- Perceptron learning algorithm

- Leave-one-out bound for kernel machines (no b)

Plan

- Leave-one-out bound for kernel machines (with b)
- Span bound
- Leave-one-out bound for SVMs with b
- Worst case analysis of leave-one-out error

About this class

We introduce the idea of cross-validation, leave-one-out in its extreme form. We show that the leave-one-out estimate is almost unbiased. We then show a series of approximations and bounds on the leave-one-out error that are used for computational efficiency. First this is shown for least-squares loss then for the SVM loss function. We close by reporting in a worst case analysis the leave-one-out error is not a significantly better estimate of expected error than is the training error.

Cross-validation

Given $S^\ell = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_\ell, y_\ell)\}$. An algorithm is a mapping from $S \rightarrow f_S$. We would like to measure the generalization error.

Cross-validation is one approach to do this. Use $\ell - p$ samples to find the function $f_{S^{\ell-p}}$. Measure the error rate on the remaining p samples

$$e_1 = \frac{1}{p} \sum_{i \in S^p} V(f_{S^{\ell-p}}(\mathbf{x}_i), y_i).$$

Repeat this procedure N times and compute

$$\hat{e} = \frac{1}{N} \sum_{i=1}^{N} e_i.$$

Hopefully $\hat{e}$ is a good measure of generalization error of f_S.

The leave-one-out error is almost unbiased

For a function f_S^ℓ

$$
\begin{aligned}
I[f_{S^\ell}] &= \int_{\mathbf{x},y} V(f_{S^\ell}(\mathbf{x}), y) dP(\mathbf{x}, y) \\
\mathcal{L}(S^\ell) &= \sum_{i=1}^{\ell} V(f_{S^i}(\mathbf{x}_i), y_i).
\end{aligned}
$$

Theorem Luntz-Brailovsky
The leave-one-out estimator is almost unbiased

$$\frac{1}{\ell+1}\mathbb{E}\mathcal{L}(S^{\ell+1}) = I[f_{S^\ell}].$$

The leave-one-out error is almost unbiased (proof)

$$\begin{aligned}
\frac{1}{\ell+1}\mathbb{E}\mathcal{L}(S^{\ell+1}) &= \frac{1}{\ell+1}\int\sum_{i=1}^{\ell+1} V(f_{S^i}(\mathbf{x}_i),y_i)dP(\mathbf{x}_1,y_1)...dP(\mathbf{x}_{\ell+1},y_{\ell+1}) \\
&= \frac{1}{\ell+1}\int\sum_{i=1}^{\ell+1} (V(f_{S^i}(\mathbf{x}_i),y_i)dP(\mathbf{x}_i,y_i)) \\
& dP(\mathbf{x}_1,y_1)...dP(\mathbf{x}_{i-1},y_{i-1})dP(\mathbf{x}_{i+1},y_{i+1})...dP(\mathbf{x}_{\ell+1},y_{\ell+1}) \\
&= \frac{1}{\ell+1}\mathbb{E}\sum_{i=1}^{\ell+1} V(f_{S^i}(\mathbf{x}_i),y_i) = I[f_{S^\ell}].
\end{aligned}$$

Computing the leave-one-error is in general expensive

In general to compute the leave-one-out error one needs to train on ℓ training sets of size $\ell - 1$. This can take alot of time. The following slides show how one can either upper-bound or approximate the leave-one-out error using a function trained on all ℓ samples.

Leave-one-out cross-validation

Given the variational problem

$$\min_{f \in \mathcal{H}} \frac{1}{\ell} \sum_{i=1}^{\ell} (f(\mathbf{x}_i) - y_i)^2 + \lambda ||f||_K^2.$$

We known the solution has the form

$$f(x) = \sum_{i=1}^{\ell} c_i K(\mathbf{x}, \mathbf{x}_i),$$

where

$$\mathbf{c} = (\mathbf{K} + \lambda \ell \mathbf{I})^{-1} \mathbf{y}.$$

If we call $\mathbf{Q} = (\mathbf{K} + \lambda \ell \mathbf{I})^{-1}$ then the leave-out-out error is

$$I_S[f_{S^i}] = \frac{1}{\ell} \sum_{i=1}^{\ell} \left(\frac{y_i - f_S(\mathbf{x}_i)}{1 - \mathbf{Q}_{ii}} \right)^2.$$

Leave-one-out cross-validation (proof)

We define the vector $\mathbf{y}^*$ where $y_j^* = y_j$ if $j \neq i$ and $y_i^* = f_{S^i}(\mathbf{x}_i)$.

We can show

$$f_{S^i}(\mathbf{x}_i) = \sum_{j=1}^{\ell} \mathbf{Q}_{ij} y_j^*.$$

Now

$$\begin{aligned}
f_{S^i}(\mathbf{x}_i) - y_i &= \sum_{j=1}^{\ell} \mathbf{Q}_{ij} y_j^* - y_i \\
&= \sum_{j \neq i} \mathbf{Q}_{ij} y_j + \mathbf{Q}_{ii} f_{S^i}(\mathbf{x}_i) - y_i \\
&= \sum_{j=1}^{\ell} \mathbf{Q}_{ij} y_j - y_i + \mathbf{Q}_{ii}(f_{S^i}(\mathbf{x}_i) - y_i)
\end{aligned}$$

Leave-one-out cross-validation (proof)

$$= \quad f_S(\mathbf{x}_i) - y_i + \mathbf{Q}_{ii}(f_{S^i}(\mathbf{x}_i) - y_i).$$

So

$$y_i - f_{S^i}(\mathbf{x}_i) = \frac{y_i - f_S(\mathbf{x}_i)}{1 - \mathbf{Q}_{ii}}.$$

Generalized approximate cross-validation

To compute the cross-validation error we need to invert the matrix $\mathbf{K} + \ell\lambda\mathbf{I}$ which can be expensive to compute.

An approximation to the cross vaidation error is

$$I_S[f_{S^i}] \approx \frac{1}{\ell}\frac{\sum_{i=1}^{\ell}(y_i - f_S(\mathbf{x}_i))^2}{(1 - \ell^{-1}\text{tr}\mathbf{Q})^2}.$$

We can compute the trace of $\mathbf{Q}$ from the eigenvalues, μ_i, of $\mathbf{K} + \ell\lambda\mathbf{I}$

$$\text{tr}\mathbf{Q} = \sum_{i=1}^{\ell} \mu_i^{-1}.$$

Perceptron mistake bound

Assume we are given a data set

$$\{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_\ell, y_\ell)\},$$

with $\mathbf{x}_i \in \mathbb{R}^n$ and $y_i = \{-1, 1\}$, which is *linearly separable*. This means that there exist $\mathbf{w} \in \mathbb{R}^n$ such that

$$(\mathbf{w}^\top \mathbf{x}_i) y_i > 0, \quad i = 1, ..., \ell.$$

Theorem: A perceptron can separate a linearly separable data set in a finite number of steps τ. Moreover, if R is the bound on the norm of the training vectors and ρ the distance of the closest point from a separating hyperplane, we have

$$\tau \leq \frac{R^2}{\rho^2}$$

Proof

Let $\hat{\mathbf{w}}$ be the unit normal vector of a hyperplane separating the ℓ data with no errors and such that the distance of the closest point is equal to ρ. For simplicity we assume that this hyperplane goes through the origin. For the constraint on the minimal distance we have

$$y_i \hat{\mathbf{w}}^\top \mathbf{x}_i \geq \rho > 0, \quad i = 1, ..., \ell.$$

Starting with $\mathbf{w}^{(0)} = 0$, we introduce the following learning rule:

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$$

if the point $\mathbf{x}_i$ is misclassified by $\mathbf{w}^{(t)}$, or $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)}$ otherwise.

Proof (cont.)

After τ updates we can write

$$\mathbf{w}^{(\tau)} = \sum_i d_i y_i \mathbf{x}_i$$

where d_i denotes the number of times in which $\mathbf{x}_i$ was misclassified over training. If the points are drawn randomly some of the d_i could be zero but we surely have

$$\sum d_i = \tau.$$

Now, since $\|\widehat{\mathbf{w}}\| = 1$, taking the dot product between $\widehat{\mathbf{w}}$ and $\mathbf{w}^{(\tau)}$ we find the following bound

$$\|\mathbf{w}^{(\tau)}\| \geq |\mathbf{w}^{(\tau)\top} \widehat{\mathbf{w}}| = |\sum_i d_i y_i \mathbf{x}_i^\top \widehat{\mathbf{w}}| \geq \tau\rho.$$

Therefore, $\|\mathbf{w}^{(\tau)}\|$ is bounded from below by a function growing linearly with τ.

Proof (cont.)

Expanding the square of $\|\mathbf{w}^{(\tau+1)}\|$ we find

$$\|\mathbf{w}^{(\tau+1)}\|^2 = \|\mathbf{w}^{(\tau)}\|^2 + \|\mathbf{x}_i\|^2 + 2y_i\mathbf{x}_i^\top\mathbf{w}^{(\tau)}.$$

Now, for all $i = 1, ..., \ell$ $\|\mathbf{x}_i\| \leq R$ and the cross product is not positive (because the i-th point has been misclassified). Therefore, at each step in which a correction takes place, the square of the norm of $\mathbf{w}^{(\tau)}$ does not increase by more than R^2.

Proof (cont.)

Therefore, after τ steps $\|\mathbf{w}^{(\tau)}\|^2$ is bounded from above by a function growing linearly with τ, or

$$\|\mathbf{w}^{(\tau)}\|^2 \leq \tau R^2.$$

Combining the two bounds we find

$$\tau^2 \rho^2 \leq \|\mathbf{w}^{(\tau)}\|^2 \leq \tau R^2,$$

which is a contradiction unless

$$\tau \leq \frac{R^2}{\rho^2}$$

Bounding the leave-one-out error

Note that the number of errors in the leave-one-out procedure has to be smaller than the the number of corrections τ the perceptron makes so

$$I_S[f_{S^i}] = \frac{1}{\ell} \sum_{i=1}^{\ell} \theta(-y_i f_{S^i}(\mathbf{x}_i)) \leq \frac{1}{\ell} \frac{R^2}{\rho^2}.$$

One can apply this bound to a SVM that is separable and has no b term.

Bound based upon number of support vectors

The leave-one-out error of a SVM can be bound by the number of support vectors N

$$I_S[f_{S^i}] \leq \frac{N}{\ell}.$$

Since the SVM solution has the form

$$f(x) = \sum_{i=1}^{N} c_i K(\mathbf{x}, \mathbf{x}_i),$$

when we remove a nonsupport vector nothing changes so leaving out that point would have no effect on accuracy. If we remove a support vector we simply assume that an error is made.

Bound for SVMs without a b term

For a SVM without a b term trained on ℓ points the solution has the form

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} c_i K(\mathbf{x}, \mathbf{x}_i).$$

For such an algorithm

$$\frac{1}{\ell}\sum_{i=1}^{\ell} \theta(-y_i f_{S^i}(\mathbf{x}_i)) \leq \frac{1}{\ell}\sum_{i=1}^{\ell} \theta(-y_i(f_S(\mathbf{x}_i) - c_i K(\mathbf{x}_i, \mathbf{x}_i))),$$

or

$$\begin{aligned} f_S(\mathbf{x}_i) - c_i K(\mathbf{x}_i, \mathbf{x}_i) &= \sum_{j \neq i} c_j K(\mathbf{x}_i, \mathbf{x}_j) \\ f_{S^i}(\mathbf{x}_i) &\geq \sum_{j \neq i} c_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \theta(-y_i f_{S^i}(\mathbf{x}_i)) &\leq \theta(-y_i \sum_{j \neq i} c_j K(\mathbf{x}_i, \mathbf{x}_j)). \end{aligned}$$

Bound for SVMs without a b term (proof)

The dual maximization problem for the leave-one-out SVM is

$$\max_{\Lambda_{\ell-i}} J_{\ell-i}(\Lambda_{\ell-i}) = \sum_{j \neq i} \alpha_i - \frac{1}{2} \sum_{j,k \neq i} y_j y_k \alpha_j \alpha_k K(\mathbf{x}_i, \mathbf{x}_j).$$

If we knew the optimal α_i^* for the ℓ point problem we could solve the following maximization problem to compute the remaining $\Lambda_{\ell-i}^*$

$$\max_{\Lambda_{\ell-i}} J_{\ell}(\Lambda_{\ell-i}) = J_{\ell-i}(\Lambda_{\ell-i}) - \alpha_i^* y_i \sum_{j \neq i} \alpha_j y_j K(\mathbf{x}_i, \mathbf{x}_j).$$

Bound for SVMs without a b term (proof)

We know the following two facts

$$\begin{aligned} J_\ell(\Lambda^*_{\ell-i}) &\geq J_\ell(\Lambda_{\ell-i}) \\ J_{\ell-1}(\Lambda^*_{\ell-i}) &\leq J_{\ell-1}(\Lambda_{\ell-i}) \end{aligned}$$

where $\Lambda^*_{\ell-i}$ are the optimal $\ell - i$ paramaters looking at all ℓ points and $\Lambda_{\ell-i}$ are the optimal $\ell - 1$ parameters looking at the $\ell - i$ points.

We can now state the following

$$\begin{aligned} J_{\ell-i}(\Lambda^*_{\ell-i}) - \alpha_i^* y_i \sum_{j \neq i} \alpha_j^* y_j K(\mathbf{x}_i, \mathbf{x}_j) &\geq J_{\ell-i}(\Lambda_{\ell-i}) - \alpha_i^* y_i \sum_{j \neq i} \alpha_j^i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \alpha_i^* y_i \sum_{j \neq i} \alpha_j^i y_j K(\mathbf{x}_i, \mathbf{x}_j) &\geq \alpha_i^* y_i \sum_{j \neq i} \alpha_j^* y_j K(\mathbf{x}_i, \mathbf{x}_j) + J_{\ell-i}(\Lambda^i_{\ell-i}) \\ &\quad - J_{\ell-i}(\Lambda^*_{\ell-i}) \\ &\geq \alpha_i^* y_i \sum_{j \neq i} \alpha_j^* y_j K(\mathbf{x}_i, \mathbf{x}_j). \end{aligned}$$

So

$$\begin{aligned} \alpha_i^* y_i \sum_{j \neq i} \alpha_j^i y_j K(\mathbf{x}_i, \mathbf{x}_j) &\geq \alpha_i^* y_i \sum_{j \neq i} \alpha_j^* y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ f_{S^i}(\mathbf{x}_i) &\geq \sum_{j \neq i} c_j K(\mathbf{x}_i, \mathbf{x}_j). \end{aligned}$$

Bound for SVMs with a b term

For a SVM with a b term trained on ℓ points the solution has the form

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} c_i K(\mathbf{x}, \mathbf{x}_i) + b.$$

For such an algorithm

$$\frac{1}{\ell} \sum_{i=1}^{\ell} \theta(-y_i f_{S^i}(\mathbf{x}_i)) \leq |\{i : 2\alpha_i R^2 + \xi_i \geq 1\}|,$$

where $R \geq K(\mathbf{x}, \mathbf{x}) - K(\mathbf{x}, \mathbf{z})$ for all $\mathbf{x}, \mathbf{z}$.

Here the dual maximization problem is

$$\max_{\Lambda_{\ell-i}} J_{\ell-i}(\Lambda_{\ell-i}) = \sum_{j \neq i} \alpha_i - \frac{1}{2} \sum_{j,k \neq i} y_j y_k \alpha_j \alpha_k K(\mathbf{x}_i, \mathbf{x}_j),$$

subject to $\sum_{j \neq i} y_j \alpha_j = 0$ and $0 \leq \alpha \leq C$.

Span bound

If the set of support vectors remain unchanged under the leave-one-out procedure then

$$y_i(f_S(\mathbf{x}_i) - f_{S^i}(\mathbf{x}_i)) = \alpha_i S_i^2,$$

where S_i is the distance between the point $\Phi(\mathbf{x}_i)$ and the set Ω_i

$$\Omega_i = \left\{ \sum_{j \neq i, \alpha_i > 0} \lambda_j \Phi(\mathbf{x}_j), \ \sum_{j \neq i} \lambda_j = 1 \right\}.$$

From this it can be shown

$$\frac{1}{\ell} \sum_{i=1}^{\ell} \theta(-y_i f_{S^i}(\mathbf{x}_i)) = \frac{1}{\ell} \sum_{i=1}^{\ell} \theta(\alpha_i S_i^2 - 1).$$

Worst case analysis for leave-one-out estimator

For certain types of algorithms, k-Nearest Neighbors for example, it was shown that the deviation between the leave-one-out estimator and the expected error is $O\left(\sqrt{\frac{1}{n}}\right)$ but one cannot bound the deviation between to empirical error and expeceted error.

This prompted the following question about VC classes. *Is the leave-one-out estimator a significantly better estimate of the expected error than the empirical error ?*

A negative result

For VC classes the leave-one-out estimate is not significantly better than the training error as an estimate of the expected error.

For a function class with VC dimension d

$$\mathbb{E}_S[I[f_S] - I_S[f_S]] \le \Theta\left(\sqrt{\frac{d(\ln \frac{2n}{d} + 1) + \ln \frac{9}{\delta}}{n}}\right) + M\delta.$$

For a function class with VC dimension d an implication of stability results is that

$$\mathbb{E}_S\left[\frac{1}{n}\sum_{i=1}^{n} V(f_{S^i}, z_i) - I_S[f_S]\right] \le \Theta\left(\sqrt{\frac{d(\ln \frac{2n}{d} + 1) + \ln \frac{9}{\delta}}{n}}\right) + M\delta,$$

$$\mathbb{E}_S\left[\frac{1}{n}\sum_{i=1}^{n} V(f_{S^i}, z_i) - I[f_S]\right] \le \Theta\left(\sqrt{\frac{d(\ln \frac{2n}{d} + 1) + \ln \frac{9}{\delta}}{n}}\right) + M\delta.$$