CSE 546: Machine Learning **Lecture 6**

Feature Selection: Part 2

Instructor: Sham Kakade

1 Greedy Algorithms (continued from the last lecture)

There are variety of greedy algorithms and numerous naming conventions for these algorithms. These algorithms must rely on some stopping condition (or some condition to limit the sparsity level of the solution).

1.1 Stagewise Regression / Matching Pursuit / Boosting

Here, we typically do no regularize our objective function and, instead, directly deal with the empirical loss $\hat{L}(w_1, w_2, \ldots w_n)$. This class of algorithms for minimizing an objective function $\hat{L}(w_1, w_2, \ldots w_n)$ is as follows:

1. Initialize: $w = 0$
2. choose the coordinate i_* which can result in the greatest decrease in error, i.e.
$$i_* \in \arg\min_i \min_{z \in R} F(w_1, \ldots, w_{i-1}, z, w_{i+1}$$
3. update w_{i_*} as follows:
$$w_i \leftarrow \arg\min_{z \in R} F(w_1, \ldots, w_{i_*-1}, z, w_{i_*+1}, \ldots w_d)$$
where the optimization is over the i-th coordinate (holding the other coordinates fixed).
4. While some termination condition is not met, return to step 2. This termination condition can be looking at the error on some holdout set or simply just running the algorithm for some predetermined number of steps.

Variants: Clearly, many variants are possible. Sometimes (for loss functions other than the square loss) it is costly to do the minimization exactly so we sometimes choose i_* based on another method (e.g. the magnitude of the gradient of a coordinate). We could also re-optimize all the weights of all those features which were are currently added. Also, sometimes we do *backward steps* where we try to prune away some of the features which are added.

Relation to boosting: In boosting, we sometimes do not explicitly enumerate the set of all features. Instead, we have a "weak learner" which provides us with a new feature. The importance of this viewpoint is that sometimes it is difficult to enumerate the set of all features (e.g. our features could be decision trees, so our feature vector x could be of dimension the number of possible tress). Instead, we just assume some oracle which in step 2 which provides us with a feature.

There are numerous variants.

1.2 Stepwise Regression / Orthogonal Matching Pursuit

Note that the previous algorithm finds i_* by only checking the improvement in performance keeping all the other variables fixed. At any given iteration, we have some subset $\mathcal{S}$ of features whose weights are not 0. Instead, when determining which coordinate i to add, we could look improvement based on reoptimizing the weights on the full set $\mathcal{S} \cup \{i\}$. This is a more costly procedure computationally, though there are some ways to reduce the computational cost.

2 Feature Selection in the Orthogonal Case

Let us suppose there are s relevant features out of the d possible features. Throughout this analysis, let us assume that:

$$\mathbf{Y} = \mathbf{X}w_* + \eta,$$

where $\mathbf{Y} \in R^n$ and $\mathbf{X} \in \mathbb{R}^{n \times d}$ ad $\eta \in R^n$ is the Gaussian noise vector (with each coordinate sampled in$N(0, \sigma^2)$). We assume that the support of w_* (the number of non-zero entries) is s.

Let us suppose that our design matrix is orthogonal. In particular, suppose that:

$$\Sigma = \frac{1}{n}\mathbf{X}^\top \mathbf{X} = \text{diagonal}$$

Now let us consider the least squares estimate (ignoring feature selection issues). Under the diagonal assumption, without loss of generality, we can assume that:

$$\mathbf{\Sigma} = \mathrm{I}$$

(by rescaling each coordinate). Here we have that j-th coordinate of the (global) least squares estimate $[\hat{w}]_j$ is just correlation between j-th dimension and $\mathbf{Y}$.

$$[\hat{w}_{\text{least squares}}]_j = \frac{1}{n}[\mathbf{X}^\top \mathbf{Y}]_j = \frac{1}{n}\sum_i \mathbf{X}_{i,j} Y_i$$

2.1 A high probability regret bound

Suppose we knew the support size s. Let us consider the the estimator which minimizes the empirical loss and has support only on s coordinates. In particular, consider the estimator:

$$\hat{w}_{\text{subset selection}} = \arg\min_{\text{support}(w) \leq s} \hat{L}(w)$$

where the $\inf$ is over vectors with support size s.

In the orthogonal case, computing this estimator is actually rather easy. Provided we have scaled the coordinates so that $\mathbf{\Sigma}$ is the identity, our estimate simply choose these the s largest coordinates of $\hat{w}_{\text{least squares}}$. In other words, a simple forward greedy algorithm suffices to compute $\hat{w}_{\text{subset selection}}$.

Now let us explicitly provided the following high probability bound on the parameter error:

Theorem 2.1. *(high probability bound) We have that with probability greater than $1 - \delta$:*

$$\|\hat{w}_{\textit{subset selection}} - w_*\|^2 \leq \frac{10 s \log(2d/\delta)}{n}\sigma^2$$

Proof. For any particular coordinate j, the Gaussian tail bound implies that, with probability greater than $1 - \delta$, that:

$$[\hat{w}_{\text{least squares}}]_j \leq [w_*]_j + \sqrt{\frac{2\sigma^2 \log(1/\delta)}{n}} \tag{1}$$

and also that:

$$[\hat{w}_{\text{least squares}}]_j \geq [w_*]_j - \sqrt{\frac{2\sigma^2 \log(1/\delta)}{n}}$$

(using the Gaussian tail bound which is proved in the optional reading). We would like these inequalities to *simultaneously* hold for all coordinates j.

The union bounds states that for events $\mathcal{E}_1$ to $\mathcal{E}_k$ that:

$$\Pr(\mathcal{E}_1 \text{ or } \mathcal{E}_2 \dots \text{ or } \mathcal{E}_k) \leq \sum_j \Pr(\mathcal{E}_j)$$

Now consider the following $2d$ events: one of the above 2 equations fail for some coordinate j. Note that if use $\delta/2d$ in the above the cumulative failure probability is less than:

$$\Pr(\text{ any failure }) \leq \sum_j \Pr(\text{ one-sided failure for j}) \leq 2d(\delta/2d) = \delta$$

Hence, we have that, with probability greater than $1-\delta$, that:

$$\max_j |[\hat{w}_{\text{least squares}}]_j - [w_*]\| \leq \sqrt{\frac{2\sigma^2 \log(2d/\delta)}{n}} := \Delta$$

Here, we have that δ is a bound on the cumulative failure probability. Also, we have defined Δ in the last equality.

Let $\mathcal{S}_*$ be the optimal support set (e.g. the support set of w_*). For any w we have:

$$\|w - w_*\|^2 = \sum_{j \notin \mathcal{S}_*} [w]_j^2 + \sum_{j \in \mathcal{S}_*} ([w]_j - [w_*]_j)^2$$

Now, for those features $j \notin \mathcal{S}_*$, we have $[\hat{w}_{\text{least squares}}]_j \leq \Delta$. Hence, for those features $j \in \mathcal{S}_*$, we have that:

$$|[\hat{w}_{\text{least squares}}]_j - [w_*]_j| \leq 2\Delta$$

To see this note that, if $[w_*]_j > 2\Delta$, then we will include this feature (and our estimated error is Δ). If $[w_*]_j \leq 2\Delta$, then we might mistakenly exclude this feature (in which case the above is also true).

Hence,

$$\sum_{j \in \mathcal{S}_*} ([\hat{w}_{\text{least squares}}]_j - [w_*]_j)^2 \leq 4s\Delta^2$$

Also, as we only include at most s features we have:

$$\sum_{j \notin \mathcal{S}^\star} ([\hat{w}_{\text{least squares}}]_j)^2 \leq s\Delta^2$$

Adding these together (and using the value of Δ) completes the proof. □

2.2 The Lasso in the orthogonal case

Let us consider the case where $\Sigma = \mathrm{I}$. Note if Σ is simply diagonal, then must rescale the coordinates for this algorithm to work [1].

We can now argue that using $\lambda = \sigma\sqrt{\frac{\log d}{n}}$ suffices to give the Lasso a high probability of success.

Theorem 2.2. *(Lasso in the orthogonal case) Suppose $\Sigma = \mathrm{I}$. Set $\lambda = 10\sigma\sqrt{\frac{\log(d/\delta)}{n}}$. Let*

$$\hat{w}_{lasso} \in \arg\min \frac{1}{n}\|\mathbf{X}w - \mathbf{Y}\|^2 + \lambda\|w\|_1$$

Then we have that with probability greater than $1-\delta$

$$\|\hat{w}_{lasso} - w_*\|^2 \leq c\frac{s\log(d/\delta)}{n}$$

(where c is a universal constant).

[1] Also, note if we had used the greedy algorithm, we do not need to explicitly do this this rescaling.

Proof. Note that:

$$\frac{1}{n}\|\mathbf{X}w - \mathbf{Y}\|^2 + \lambda\|w\|_1 = \|w - w_*\|^2 + \frac{1}{n}\eta^\top \mathbf{X}(w - w_*) + \lambda\|w\|$$

using that $\Sigma = \mathrm{I}$ and the definition of Y.

Hence, the Lasso is minimizing:

$$\|w - w_*\|^2 + \frac{1}{\sqrt{n}}\tilde{\eta}^\top(w - w_*) + \lambda\|w\|_1$$

where we have defined $\tilde{\eta} = \frac{1}{\sqrt{n}}\mathbf{X}^\top\eta$. By assumption on $\mathbf{X}$ we have that $\tilde{\eta}$ is a $N(0, \mathrm{I}_d)$.

Hence, the Lasso simplifies to solving separate. 1-dimensional problems. Also, with probability greater than $1 - \delta$, each coordinate of $\frac{1}{\sqrt{n}}\tilde{\eta}$ is bounded by $\sigma\sqrt{2\frac{\log d/\delta}{n}}$. Setting λ to this value ensures that all irrelevant features will be thresholded to 0. And all relevant features will have their weight shrunk by λ, which results in the regret that is same as the subset selection algorithm. □

3 When do the Lasso and the greedy algorithm also have low regret?

There has been much work showing that the Lasso and the greedy algorithms can obtain regret bounds comparable to the subset selection algorithm.

These assumptions can be viewed as relaxations to the orthogonal condition. One weakening is based on incoherence (which is essentially an assumption that the features matrix $\mathcal{X}$ has properties similar to that of a random matrix). Namely, this assumption is that for all coordinates $j \neq k$:

$$\frac{1}{n}\sum_i \mathbf{X}_{i,j}\mathbf{X}_{i,k} \approx \frac{1}{\sqrt{n}}$$

where we also assume that:

$$\frac{1}{n}\sum_i \mathbf{X}_{i,j}^2 = 1$$

The *Restricted Isometry Property (RIP)* is a weaker assumption than this. Under either of these assumptions, both the Lasso and the greedy algorithms can have risk bounds comparable to that of the subset selection algorithm.