Section 6

1 Kernelized Linear Regression

Consider regularized linear regression (without a bias, for simplicity). Our objective to find the optimal parameters $\hat{w} = \arg\min_w L(w)$ for a dataset $\{(x_i, y_i)\}_{i=1}^n$ that minimize the following loss function:

$$L(w) = \sum_{i=1}^{n} (w^T x_i - y_i)^2 + \lambda ||w||_2^2$$

We claim that the optimal $\hat{w}$ lies in the span of the of the datapoints. Concretely, there exists $\alpha_1, ..., \alpha_n \in \mathbb{R}$ such that:

$$\hat{w} = \sum_{i}^{n} \alpha_i x_i$$

In this case, we can re-write our prediction function, for an input $x \in \mathbb{R}^d$ as the following:

$$\begin{aligned}
\hat{f}(x) &= \hat{w}^T x \\
&= (\sum_{i}^{n} \alpha_i x_i)^T x \\
&= \sum_{i}^{n} \alpha_i x_i^T x \\
&= \sum_{i}^{n} \alpha_i < x_i, x >
\end{aligned}$$

Note that since we can write the prediction function in terms of inner products ($< x_i, x >$), we can replace this with your favorite kernel function $K(x_i, x)$ to get the following:

$$\hat{f}(x) = \sum_{i}^{n} \alpha_i K(x_i, x)$$

In a proof by algebra and substitution, one can transform the objective to the following:

$$\hat{\alpha} = \arg\min_{\alpha} ||\mathbf{K}\alpha - y||_2^2 + \lambda \alpha^T \mathbf{K} \alpha$$

where $\mathbf{K} \in \mathbb{R}^{n \text{ x } n}$, such that $\mathbf{K}_{ij} = K(x_i, x_j)$. Here, we are now optimizing over α instead of w. A proof by calculus and substitution will reveal that the closed form solution is the following:

$$\hat{\alpha} = (\mathbf{K} + \lambda \mathbf{I})^{-1} y$$

2 Proof of $\hat{w} \in Span(x_1, ..., x_n)$

We will prove this through contradiction. Assume $\hat{w} \notin Span(x_1, ..., x_n)$ solves $\arg\min_w L(w)$. Then, there exists a component of $\hat{w}$ that is perpendicular to the span, which we will call $w^\perp$. Concretely,

$$\hat{w} = \bar{w} + w^\perp$$

Where $\bar{w} = \sum_i^n \alpha_i x_i$ is the component of $\hat{w}$ in the span of the datapoints. Note that $\hat{w} \cdot x_i = \bar{w} \cdot x_i$, for every x_i since

$$\begin{aligned}
\hat{w} \cdot x_i &= (\bar{w} + w^\perp) \cdot x_i \\
&= \bar{w} \cdot x_i + w^\perp \cdot x_i \\
&= \bar{w} \cdot x_i + 0 && w^\perp \text{ is perpendicular to each } x_i \\
&= \bar{w} \cdot x_i
\end{aligned}$$

Additionally, note that $||\hat{w}||_2^2 \geq ||\bar{w}||_2^2$, because of the following:

$$\begin{aligned}
||\hat{w}||_2^2 &= ||\bar{w} + w^\perp||_2^2 \\
&= ||\bar{w}||_2^2 + ||w^\perp||_2^2 && \text{Pythagorean theorem} \\
&\geq ||\bar{w}||_2^2
\end{aligned}$$

Note that in the loss function we're trying to minimize the magnitude of w (with the regularization term $\lambda||w||_2^2$). Note that if $\forall_i \hat{w}^T x_i = \bar{w}^T x_i$, and $||\hat{w}|| \geq ||\bar{w}||$, then our optimization will always choose $w^\perp = 0$, meaning that $\hat{w} = \bar{w}$ and $\hat{w} \in Span(x_1, ..., x_n)$, which completes the contradiction.

3 Random Forest and Bagging

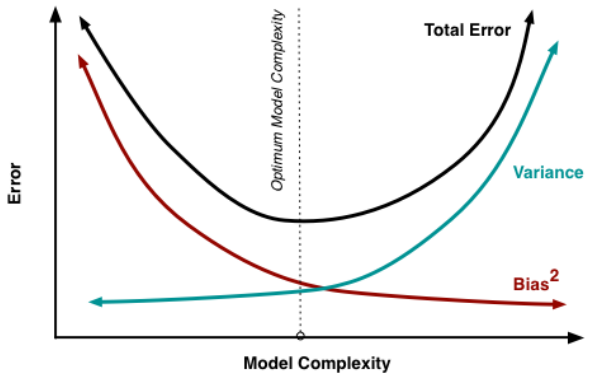
Remember from bias-variance tradeoff, we derived the following equation:

$$\begin{aligned}
&\text{Test Error} = \text{Variance} + \text{Bias}^2 = \\
&\mathbb{E}_{D \sim P^N, (x,y) \sim P}[(h_D(x) - \bar{h}(x))^2] + \mathbb{E}_{(x,y) \sim P}[(\bar{h}(x) - \bar{y}(x))^2]
\end{aligned}$$

where $\bar{h}(x) = \mathbb{E}_D[h_D(x)]$, which can be thought as the weighted average over all possible h_D, each trained on a different possible dataset. $\bar{y}(x)$ is the underlying ground truth function. So variance measures how much does an individual hypothesis deviates from an average over all possible hypotheses. Bias measures how much the *expected* hypothesis (i.e. the average over all possible hypotheses above) deviates from the true underlying function.

Exercise A decision tree without depth limit has _____ variance and _____ bias.

Recall, we also had the following graph from bias-variance tradeoff, where the optimal model has relatively low bias and low variance.



Random Forest keeps the bias to be relatively low, and reduces variance effectively. In particular, random forest is an ensemble of random trees (thus the name), $\{T_b\}_1^B$, each trained on a corresponding *bootstrapped* dataset Z_b^*. The algorithm for constructing random forests is shown on the next page.

Bootstrapping: Given a dataset $Z = \{(x_i, y_i)\}_1^N$, the bootstrapped dataset $Z_b^* = \{(x_i', y_i')\}_1^N$, where (x_i', y_i') are elements uniformly sampled from Z with replacement.

Algorithm 15.1 *Random Forest for Regression or Classification.*

1. For $b = 1$ to B:
   (a) Draw a bootstrap sample $\mathbf{Z}^*$ of size N from the training data.
   (b) Grow a random-forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.
      i. Select m variables at random from the p variables.
      ii. Pick the best variable/split-point among the m.
      iii. Split the node into two daughter nodes.
2. Output the ensemble of trees $\{T_b\}_1^B$.

To make a prediction at a new point x:

Regression: $\hat{f}_{\text{rf}}^B(x) = \frac{1}{B}\sum_{b=1}^B T_b(x)$. m~p/3

Classification: Let $\hat{C}_b(x)$ be the class prediction of the bth random-forest tree. Then $\hat{C}_{\text{rf}}^B(x) = \textit{majority vote}\ \{\hat{C}_b(x)\}_1^B$. m~sqrt(p)

For a tree T_b trained on Z_b^*, let h_{T_b} denote the corresponding hypothesis function. Note, on expectation, the variance for each individual tree has not changed:

$$\mathbb{E}_{D\sim P^N,(x,y)\sim P}[(h_T(x) - \bar{h}(x))^2] = \mathbb{E}_{D\sim P^N,(x,y)\sim P}[(h_{T_b}(x) - \bar{h}(x))^2] = \sigma^2$$

However, now we have B trees, though still dependent, they are much less correlated than having B identical trees. In math, this means:

$$\mathbb{E}_{D\sim P^N,(x,y)\sim P}[(h_{T_i}(x) - \bar{h}(x))(h_{T_j}(x) - \bar{h}(x))] = \delta\sigma^2$$

Now, if use the ensemble of the hypotheses, we get a much smaller variance:

$$\begin{aligned}
&\mathbb{E}[(\frac{1}{B}\sum_{b=1}^B h_{T_b}(x) - \bar{h}(x))^2] \\
&= \frac{1}{B^2}\sum_{b=1}^B \mathbb{E}[(h_{T_b}(x) - \bar{h}(x))^2] + \frac{1}{B^2}\sum_{i\neq j}\mathbb{E}[(h_{T_i}(x) - \bar{h}(x))(h_{T_j}(x) - \bar{h}(x))] \\
&= \frac{1}{B^2}(B\sigma^2 + (B-1)B\delta\sigma^2) = \frac{1}{B}\sigma^2 + \frac{B-1}{B}\delta\sigma^2
\end{aligned}$$

Lastly, notice in the above derivations, we kind of cheated by assuming

$$\bar{h}(x) = \mathbb{E}[h_T(x)] = \mathbb{E}[h_{T_b}(x)]$$

However, the last equality is only true asymptotically when the number of data points $N \to \infty$. Nevertheless, we can assume them to be approximately equal when N is large.