CMSE 890-002: Mathematics of Deep Learning, MSU, Spring 2020

Lecture 09: Linear Models and Avoiding the Curse of Dim.

January 31, 2020

Lecturer: Matthew Hirn

While k-nearest neighbors suffers from the curse of dimensionality absent low dimensional structure in the probability density function $p_X(x)$, linear models can circumvent the curse of dimensionality in certain circumstances.

Another advantage one may have is prior knowledge on the labeling function $y = F(x)+\varepsilon$. Indeed, if $F(x) = \langle x, \theta_0 \rangle$ is linear, and we know this fact but we do not know the parameters θ_0, we can still leverage this knowledge to restrict our model class to the class of linear models. Suppose this is the case, and that the labels are corrupted versions of $F(x)$, i.e.,

$$y = F(x) + \varepsilon = \langle x, \theta_0 \rangle + \varepsilon \,, \quad x \in \mathbb{R}^d \,, \;\; \varepsilon \sim \mathcal{N}(0, \sigma^2) \,.$$

Suppose we take our model class to be all possible linear models

$$\mathcal{F} = \{ f(x; \theta) = \langle x, \theta \rangle : \theta \in \mathbb{R}^d \} \,.$$

Given a training set $T = \{(x_i, y_i)\}_{i=1}^N$ we obtain a model from $\mathcal{F}$ by minimizing the squared loss:

$$\widehat{\theta}_T = \arg\min_{\theta \in \mathbb{R}^d} \frac{1}{N} \sum_{i=1}^N (y_i - \langle x_i, \theta \rangle)^2 \,.$$

From equation (5) we know that

$$\widehat{\theta}_T = (\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}\mathbf{y} \,,$$

where $\mathbf{X}$ is the $d \times N$ matrix containing the training point x_i on the i^{th} column, and $\mathbf{y}$ is the $N \times 1$ vector containing y_i in the i^{th} entry (note we have transposed $\mathbf{X}$ from its presentation in (5)).

Also let $\boldsymbol{\varepsilon}$ be the $N \times 1$ vector with ε_i as its i^{th} entry. Now let $x \in \mathbb{R}^d$ be a test point, which we consider as a $d \times 1$ vector to be consistent with $\mathbf{X}$. Our prediction for its label is $f(x; \widehat{\theta}_T)$, which we can write as:

$$\begin{aligned} f(x; \widehat{\theta}_T) &= \langle x, \widehat{\theta}_T \rangle \\ &= \langle x, (\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}\mathbf{y} \rangle \\ &= \langle x, (\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}(\mathbf{X}^T\theta_0 + \boldsymbol{\varepsilon}) \rangle \\ &= \langle x, \theta_0 + (\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}\boldsymbol{\varepsilon} \rangle \\ &= \langle x, \theta_0 \rangle + \langle x, (\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}\boldsymbol{\varepsilon} \rangle \\ &= F(x) + \langle \mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}x, \boldsymbol{\varepsilon} \rangle \\ &= F(x) + \sum_{i=1}^N (\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}x)(i)\varepsilon_i \,. \end{aligned}$$

From this calculation and the fact that $\mathbf{X}$ is independent of ε_i, we conclude that

$$\mathbb{E}_T[f(x;\widehat{\theta}_T)] = F(x) + \sum_{i=1}^{N} \mathbb{E}_T[(\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}x)(i)]\mathbb{E}_T[\varepsilon_i] = F(x)\,,$$

showing that the least squares fit is unbiased estimator for linear models since from Theorem 4.2 we have:

$$\text{bias } = F(x) - \mathbb{E}_T[f(x;\widehat{\theta}_T)]\,.$$

Therefore, if we apply Theorem 4.2 (bias-variance trade-off), we will have only the irreducible error σ^2 and the variance error,

$$\begin{aligned}
\text{Err}(\mathcal{F}, x) &= \sigma^2 + \text{Var}_T\left[\sum_{i=1}^{N}(\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}x)(i)\varepsilon_i\right] \\
&= \sigma^2 + \mathbb{E}_T\left[\left(\sum_{i=1}^{N}(\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}x)(i)\varepsilon_i\right)^2\right] \\
&= \sigma^2 + \mathbb{E}_T\left[\sum_{i,j=1}^{N}(\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}x)(i)\varepsilon_i(\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}x)(j)\varepsilon_j\right] \\
&= \sigma^2 + \sum_{i,j=1}^{N}\mathbb{E}_T[(\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}x)(i)(\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}x)(j)]\underbrace{\mathbb{E}_T[\varepsilon_i\varepsilon_j]}_{\sigma^2\delta(i-j)} \\
&= \sigma^2 + \sigma^2\sum_{i=1}^{N}\mathbb{E}_T[(\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}x)(i)^2] \\
&= \sigma^2 + \sigma^2\mathbb{E}_T\left[\|\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}x\|_2^2\right]\,.
\end{aligned}$$

Now, we also have:

$$\begin{aligned}
\|\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}x\|_2^2 &= (\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}x)^T\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}x \\
&= x^T(\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}x = x^T(\mathbf{X}\mathbf{X}^T)^{-1}x
\end{aligned}$$

Therefore

$$\mathbb{E}_T\left[\|\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}x\|_2^2\right] = \mathbb{E}_T[x^T(\mathbf{X}\mathbf{X}^T)^{-1}x] = x^T\mathbb{E}_T[(\mathbf{X}\mathbf{X}^T)^{-1}]x\,.$$

Now, this quantity is a real number. We can write any real number $a \in \mathbb{R}$ as Trace$[a]$, where we view a as a 1×1 matrix. Recall that Trace$[A] = \sum_{k=1}^{m} A(k,k)$ for an $m \times m$ matrix A. We also note that for matrices A, B, C we have

$$\text{Trace}[ABC] = \text{Trace}[BCA]$$

whenever the matrix multiplications make sense. Therefore we have:

$$\begin{aligned} x^T \mathbb{E}_T[(\mathbf{X}\mathbf{X}^T)^{-1}]x &= \text{Trace}\left[x^T \mathbb{E}_T[(\mathbf{X}\mathbf{X}^T)^{-1}]x\right] \\ &= \text{Trace}\left[\mathbb{E}_T[(\mathbf{X}\mathbf{X}^T)^{-1}]xx^T\right] . \end{aligned} \tag{16}$$

It thus follows that:

$$\text{Err}(\mathcal{F}, x) = \sigma^2 \left(1 + \text{Trace}\left[\mathbb{E}_T[(\mathbf{X}\mathbf{X}^T)^{-1}]xx^T\right]\right) [8] .$$

Now let us compute $\text{Err}(\mathcal{F})$, which we write as:

$$\text{Err}(\mathcal{F}) = \mathbb{E}_X\left[\text{Err}(\mathcal{F}, X)\right] .$$

Therefore we need to compute the expectation with respect to a test point $X = x$ of the quantity (16). Since the trace is just a summation and expectation is linear, we can interchange them. Let us also assume that $\mathbb{E}_X[X] = (0, \ldots, 0)$ so that $\text{cov}(X) = \mathbb{E}[XX^T]$. We then have

$$\begin{aligned} \mathbb{E}_X\left\{\text{Trace}\left[\mathbb{E}_T[(\mathbf{X}\mathbf{X}^T)^{-1}]XX^T\right]\right\} &= \text{Trace}\left\{\mathbb{E}_X\left[\mathbb{E}_T[(\mathbf{X}\mathbf{X}^T)^{-1}]XX^T\right]\right\} \\ &= \text{Trace}\left\{\mathbb{E}_T[(\mathbf{X}\mathbf{X}^T)^{-1}]\mathbb{E}_X[XX^T]\right\} \\ &= \text{Trace}\left\{\mathbb{E}_T[(\mathbf{X}\mathbf{X}^T)^{-1}]\text{cov}(X)\right\} . \end{aligned} \tag{17}$$

At this point we know that

$$\mathbb{E}_T[\mathbf{X}\mathbf{X}^T] = N\text{cov}(X) ,$$

where we have the factor N instead of $N-1$ because we are assuming that $\mathbb{E}_X[X] = (0, \ldots, 0)$ and that we know this fact. Now, we would like to assert that

$$\mathbb{E}_T[(\mathbf{X}\mathbf{X}^T)^{-1}] \propto N^{-1}\text{cov}(X)^{-1} .$$

However, as was pointed out in class, this is not always true[9]. One might try to argue if N is very large, then $\mathbf{X}\mathbf{X}^T \approx N\text{cov}(X)$ with a small variance, but at this time I am not sure if this can be made rigorous. For now, one regime in which we can make things rigorous is the following. Suppose that each training data point $\{x_i\}_{i=1}^N$ is sampled independently from the normal distribution with zero mean and $d \times d$ covariance matrix $\Sigma = \text{cov}(X)$, i.e.,

$$x_i \sim \mathcal{N}(\mathbf{0}, \Sigma) ,$$

where $\mathbf{0} = (0, \ldots, 0)$. In this case we have [7],

$$\mathbb{E}_T[(\mathbf{X}\mathbf{X}^T)^{-1}] = (N - d - 1)^{-1}\Sigma^{-1} = (N - d - 1)^{-1}\text{cov}(X)^{-1} .$$

[8] Thanks for Yani Udiani for helping to make this part on the trace clearer.

[9] Thanks to Anna Little and Dylan Molho for pointing out an error in the original calculation, and thanks to Mohit Bansil, Anna Little, Gautam Sreekumar, and Ali Zare for valuable discussions thereafter.

Therefore, picking up from (17), we have:

$$\begin{aligned}\text{Trace}\left\{\mathbb{E}_T[(\mathbf{X}\mathbf{X}^T)^{-1}]\text{cov}(X)\right\} &= \frac{1}{N-d-1}\text{Trace}\left\{\text{cov}(X)^{-1}\text{cov}(X)\right\} \\ &= \frac{1}{N-d-1}\text{Trace}[\mathbf{I}] \\ &= \frac{d}{N-d-1}\,.\end{aligned}$$

To conclude, when we have a linear label model $y = \langle x, \theta_0\rangle + \varepsilon$, and when our data points $x \in \mathbb{R}^d$ are sampled from the normal distribution with mean $\mathbf{0}$[10], we have:

$$\text{Err}(\mathcal{F}) = \sigma^2\left(1 + \frac{d}{N-d-1}\right)\,,$$

where $\mathcal{F}$ is the model class of all possible linear models. We thus see that to have a small expected test error (modulo the irreducible error) the number of training points N must grow essentially linearly in the dimension d, thus circumventing the curse of dimensionality (recall the earlier examples of the cube Q with the uniform distribution and 1-nearest neighbor, in which the number of training points grew as a power of d).

On the other hand, we imposed a very heavy assumption on the data generation process, namely that the labels y be linear functions (plus noise) of the data points x. The k-nearest neighbors algorithm imposes no such restriction and in the limit of infinite training data can approximate nearly any model, but N must grow exponentially fast in the dimension d, which is unrealistic. The field of machine learning has developed a number of algorithms "in between" linear regression/classification and k-nearest neighbors, with the goal of increasing the capacity of the model class while not too drastically increasing the complexity of the search space and thus being restricted by the curse of dimensionality. In Section 5 we briefly discuss some of the non-deep learning approaches along these lines.

[10] If anyone can generalize this calculation further, I would be interested in seeing that!

References

[1] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations*, 2015.

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition*, 2009.

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105. 2012.

[4] Larry Greenemeier. AI versus AI: Self-taught AlphaGo Zero vanquishes its predecessor. *Scientific American*, October 18, 2017.

[5] Pankaj Mehta, Marin Bukov, Ching-Hao Wang, Alexandre G.R. Day, Clint Richardson, Charles K. Fisher, and David J. Schwab. A high-bias, low-variance introduction to Machine Learning for physicists. arXiv:1803.08823, 2018.

[6] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer-Verlag New York, 2nd edition, 2009.

[7] J. Hartlap, P. Simon, and P. Schneider. Why your model parameter confidences might be too optimistic - unbiased estimation of the inverse covariance matrix. *Astronomy and Astrophysics*, 464(1):399–404, 2007.

[8] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Adaptive Computation and Machine Learning. The MIT Press, 2002.