Lecture 12

7.4 Bias Variance Tradeoff

Adapted from [8, Chapter 7]

The parameter λ balances the complexity of the model and its generalization power to new points $x \in \mathcal{X}$. We examine this idea from the perspective of statistical learning theory.

As we have discussed, our goal is find a model f that minimizes the risk:

$$R[f] = \mathbb{E}[c(x, y, f(x)] = \int_{\mathcal{X}\times\mathcal{Y}} c(x, y, f(x))dP(x, y).$$

Since this is impossible, we use the only information we have, which is the training data, and consider the empirical risk:

$$R_{\text{emp}}[f] = \frac{1}{n}\sum_{i=1}^{n} c(x_i, y_i, f(x_i)).$$

As we saw in the previous sections, minimizing the empirical risk means that our model depends on the training data $\mathcal{T} = \{(x_i, y_i)\}_{i\leq n}$, i.e., $f = f_{\mathcal{T}}$. In other words, if you change the training data, you will change the model f. Recall that in the statistical framework, the training data is drawn i.i.d. from the joint distribution P. Thus a related quantity is the *conditional test error* given a particular training set:

$$\text{Err}_{\mathcal{T}} = \mathbb{E}[c(x, y, f(x))|\mathcal{T}].$$

The *expected test error* is defined as:

$$\text{Err} = \mathbb{E}_{\mathcal{T}}[\text{Err}_{\mathcal{T}}].$$

Notice that the expected test error takes the expectation not only over the random draws of the test data, but also the random draws of the training data. It thus measures the quality of the learning algorithm itself, not just the model f outputted by the algorithm for a specific training set.

Figure 14 shows that minimizing the empirical risk is not a good way to minimize the conditional test error or the expected test error. Indeed,

the empirical risk consistently decreases with the model complexity, typically dropping to zero if we increase the model complexity enough. However, such a model generalizes poorly, as we see the conditional and expected test errors increase at a certain point. In the case of kernel ridge regression, the parameter λ tunes the complexity of the model. In a learning task, one must also learn the value of λ that best minimizes the testing error.

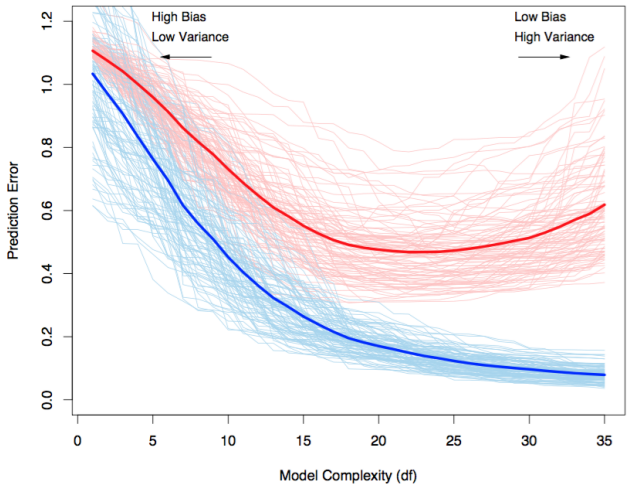

Figure 14: Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the empirical risk $R_{\text{emp}}[f]$, while the light red curves show the conditional test error $\text{Err}_{\mathcal{T}}$, for 100 random draws of the training set each of which consists of 50 data points, as the model complexity is increased. The bold curves show the expected test error Err and the expected risk $\mathbb{E}_{\mathcal{T}}[R[f]]$.

If we consider the squared loss $c(x, y, f(x)) = (y - f(x))^2$, then we can further analyze the expected test error. Let x_0 be a test point with label y_0, and let $\text{Err}(x_0)$ be the expected error at $x = x_0$ which is defined as:

$$\text{Err}(x_0) = \mathbb{E}_{\mathcal{T},x,y}[c(x, y, f(x))|x = x_0]. \tag{44}$$

For the quadratic loss, we can rewrite (44) as:

$$\begin{aligned}
\text{Err}(x_0) &= \mathbb{E}_{\mathcal{T}}[(y_0 - f(x_0))^2], \\
&= \left(\mathbb{E}_{\mathcal{T}}[f(x_0)] - y_0\right)^2 + \mathbb{E}_{\mathcal{T}}\left[f(x_0) - \mathbb{E}_{\mathcal{T}}[f(x_0)]\right]^2, \\
&= \text{Bias}\left(f(x_0)\right)^2 + \text{Var}\left(f(x_0)\right),
\end{aligned}$$

where

$$\begin{aligned}\text{Bias}\left(f(x_0)\right) &= \mathbb{E}_{\mathcal{T}}[f(x_0)] - y_0,\\ \text{Var}\left(f(x_0)\right) &= \mathbb{E}_{\mathcal{T}}\left[f(x_0) - \mathbb{E}_{\mathcal{T}}[f(x_0)]\right]^2.\end{aligned}$$

The bias term is the amount by which the average of our estimate differs from the true value. The variance term, on the other hand, is the expected squared deviation of $f(x_0)$ around its mean. Typically the more complex we make the model f, the lower the (squared) bias but the higher the variance. Thus selecting the complexity of the model f can be interpreted as a tradeoff between the model bias and the model variance. Figure 15 illustrates this point by plotting the bias, variance, and expected test error as a function of the model complexity.

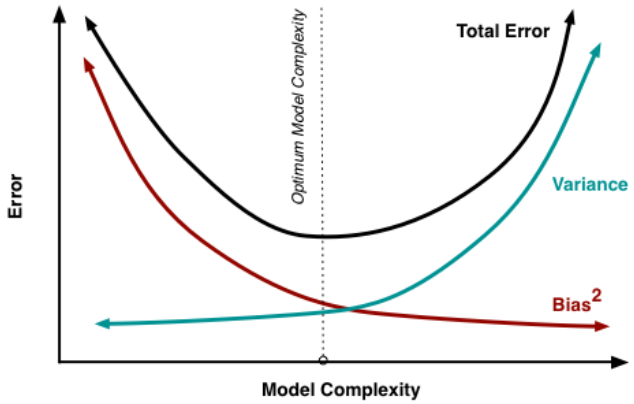

Figure 15: Plot of the model bias, variance, and expected test error as a function of the model complexity. Image taken from `http://scott.fortmann-roe.com/docs/BiasVariance.html`

Exercises

Exercise 22. Write a kernel ridge regression code that can use any kernel k and value of $\lambda > 0$. Make sure the code can compute the weights $\alpha = (\alpha_i)_{i=1}^n$ from the training data and evaluate the model f at a new point, i.e., compute $f(x) = \sum_{i=1}^n \alpha_i k(x_i, x)$.

Exercise 23. Download the dataset `house_data.mat`, which is from the paper [9]. In that paper they authors study the energy performance of buildings from a machine learning perspective. Their data set $\mathcal{X}$ consists of

eight measurements of each building: (1) Relative compactness; (2) Surface area; (3) Wall area; (4) Roof area; (5) Overall height; (6) Orientation; (7) Glazing area; (8) Glazing area distribution. Using tools from statistical machine learning, the authors regress two quantities indicative of the energy performance of the building: $\mathcal{Y}_1 =$ Heating load and $\mathcal{Y}_2 =$ Cooling load.

Using your kernel ridge regression code, regress each of the regression labels $\mathcal{Y}_1$ and $\mathcal{Y}_2$ (separately) using 50% of the data for training and 50% for testing. Test *many* different values of λ to find the optimal bias-variance tradeoff. Test as well several different kernels to see how the performance varies. Turn in your code and report on your results. Hint 1: You may want to z-score the data first along each of the eight dimensions. Hint 2: For testing λ, you may want to sample it on a logarithmic scale, that is sample s linearly from $(-C, C)$ for some $C > 0$, and then use $\lambda = 2^s$.

8 Spectral Graph Theory and Clustering

8.1 Introduction

Adapted from [10, Lecture 1]

8.1.1 Graphs

We begin with an overview of some of the topics and ideas we will cover. But first we define a graph. A *graph* $G = (V, E)$ is a set of *vertices* V, and a set of *edges* E. In an *undirected graph*, the edge set is a set of *unordered* pairs of vertices. Unless otherwise specified, by graph we will mean an undirected graph with a finite number of vertices. An example is:

- $V = \{1, 2, 3, 4, 5, 6\}$
- $E = \Big\{\{1,2\}, \{1,5\}, \{2,3\}, \{2,5\}, \{3,4\}, \{4,5\}, \{4,6\}\Big\}$

Graphs have natural visual representations. Vertices can be represented as dots, and two vertices are connected by a line. The above graph is visualized in Figure 16.

Other examples of (abstract) graphs are:

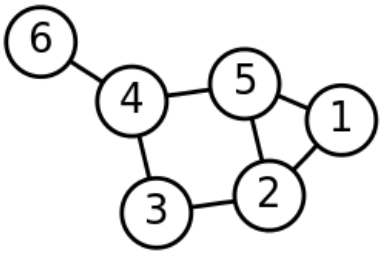

Figure 16: Example of a graph.

- The path on n vertices: The vertices are $V = \{1, \ldots, n\}$ and the edges are $E = \Big\{\{i, i+1\} : i = 1, \ldots, n-1\Big\}$.
- The ring on n vertices: The vertices are $V = \{1, \ldots, n\}$, and the edges are $E = \Big\{\{i, i+1\} : i = 1, \ldots, n-1\Big\} \cup \{1, n\}$.
- The hypercube on 2^k vertices. The vertices are elements of $\{0,1\}^k$, e.g., $(1,0,0,1) \in \{0,1\}^4$, and edges exist between vertices that differ in only one coordinate.

Abstract graphs are nice, but in data science graphs are used to model connections or relations between things, where the "things" are the vertices and the connections/relations are the edges. Some examples are:

- Friendship graphs (like Facebook): People are vertices, edges exist between pairs of people who are friends (see Figure 17).
- Airplane route graphs: Cities are vertices, and edges exist between pairs of cities for which there is a direct flight (see Figure 18).
- Network graphs: Devices, routers and computers are vertices, edges exist between pairs that are connected.

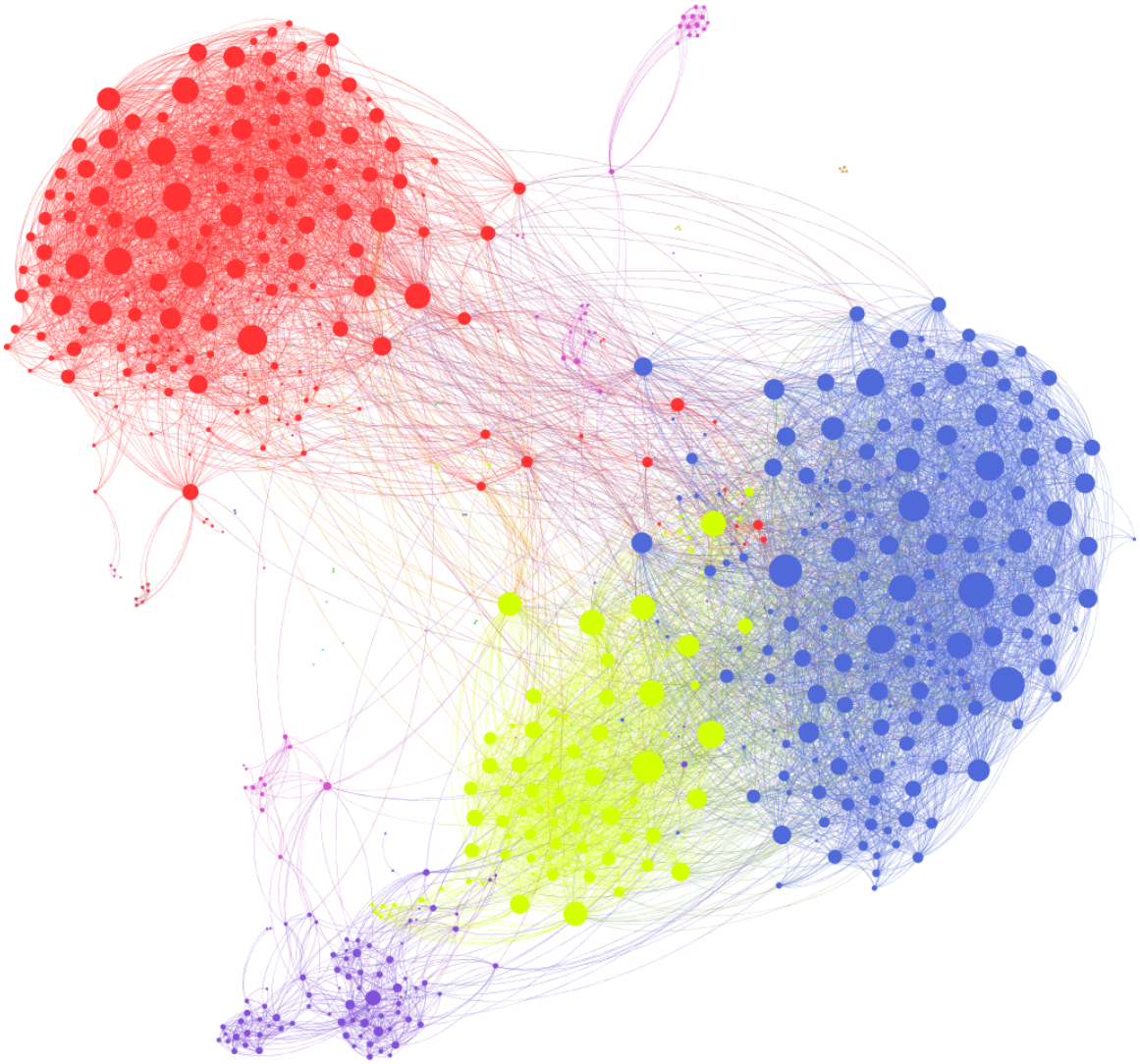

Figure 17: Facebook friendship graph of a particular person from `https://griffsgraphs.wordpress.com/tag/social-network/`. Per the description on the website, red are high school friends, blue are college friends, yellow are his girlfriend's friends, purple are academic colleagues, and pink are friends met from traveling.

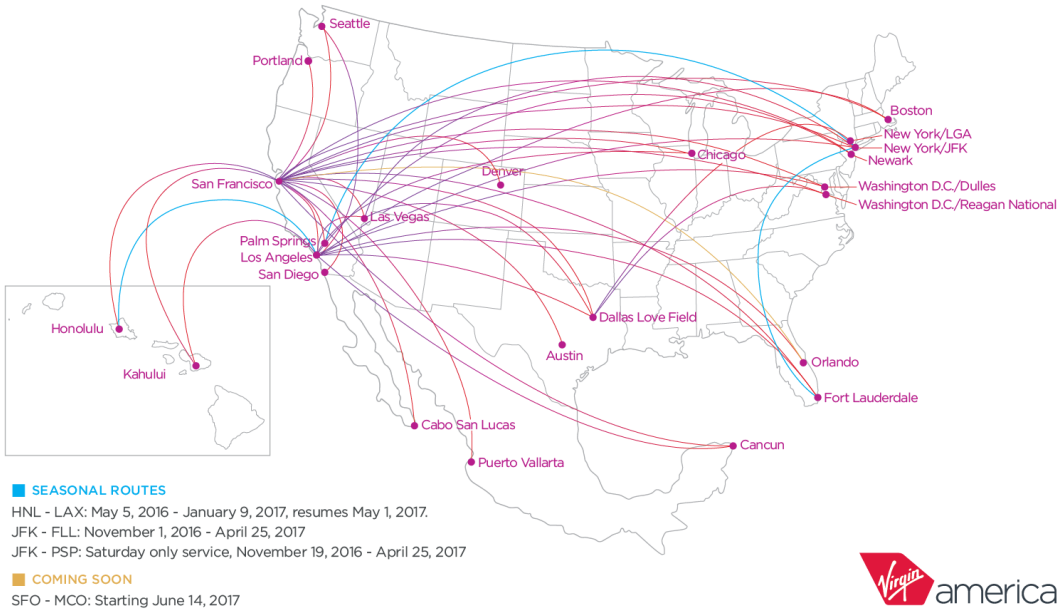

Figure 18: Virgin America direct flights, visualized as a graph.

References

[1] 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.

[2] Afonso S. Bandeira. Ten lectures and forty-two open problems in the mathematics of data science. MIT course *Topics in Mathematics of Data Science*, 2015.

[3] Jon Shlens. A tutorial on principal component analysis. arXiv:1404.1100, 2014.

[4] Karl Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine, Series 6*, 2(11):559–572, 1901.

[5] V. A. Marchenko and L. A. Pastur. Distribution of eigenvalues in certain sets of random matrices. *Mat. Sb. (N.S.)*, 72(114):507–536, 1967.

[6] J. Baik, G. Ben-Arous, and S. Péché. Phase transition of the largest eigenvalue for nonnull complex sample covariance matrices. *The Annals of Probability*, 33(5):1643–1697, 2005.

[7] Debashis Paul. Asymptotics of sample eigenstructure for a large dimensional spiked covariance model. *Statistica Sinica*, pages 1617–1642, 2007.

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

[9] Athanasios Tsanas and Angeliki Xifara. Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. *Energy and Buildings*, 49:560–567, 2012.

[10] Daniel A. Spielman. Spectral graph theory. *Yale Course Notes*, Fall, 2009.