Lecture Notes: Cost functions and the Bias-Variance Tradeoff

Ted Meeds[1,2]

[1] Informatics Institute, University of Amsterdam
[2] The Centre for Integrative Bioinformatics, Vrije University
tmeeds@gmail.com

Abstract In this note we motivate the choice of squared-loss and Gaussian likelihoods for linear regression. In particular, the choice of loss determines the optimal predictive function, for instance using squared-loss the optimal predictor is the conditional expectation, but for the sum of absolute errors, it is the conditional median. Then the irreducible loss is discussed, which leads to the ideas of bias and variance of our model. Irreducible loss if the loss we incur even if we knew the exact model, due to the noise in our data. Bias of our model refers to the difference between our expected prediction and the true function (across datasets) while variance refers to the variance of our predictions across data sets.

1 Loss functions for regression

Recall: using a Gaussian likelihood for t_n and maximizing the log-likelihood (or log-posterior) was equivalent to minimizing the sum-of-squares loss.

What is the motivation for the squared-error / loss?

More generally, imagine that we have a joint model $p(t, \mathbf{x})$ (i.e. a complete probability model of the data). For any problem, we need to make a specific *choice* for y; in other words, for the form of our **decision** function. Our choice will depend on the **expected loss**. This is easier to motivate with classification because we can assign understandable losses to misclassifying inputs (e.g. misdiagnose cancer).

2 Minimizing the expected loss

Our goal is to minimize the expected loss, given a general $p(t, \mathbf{x})$:

$$\mathbb{E}\left[\mathrm{L}\right] = \int\int \mathrm{L}\left(t, y(\mathbf{x})\right) p(t, \mathbf{x}) d\mathbf{x} dt \tag{1}$$

As an example, the squared loss: $\mathrm{L}\left(t, y(\mathbf{x})\right) = \left(t - y(\mathbf{x})\right)^2$. We assume that y is flexible and can take on many forms. In general we have:

$$\mathrm{L}\left(t, y(\mathbf{x})\right) = |t - y(\mathbf{x})|^q \tag{2}$$

where $q > 0$. We want:

$$y(\mathbf{x}) = \underset{y(\mathbf{x})}{\arg\min}\, \mathbb{E}\left[\mathrm{L}\right] \tag{3}$$

Some loss functions are shown in Figure 1. Note the quadratic penalty for the squared loss and the linear penalty for the absolute loss $q = 1$.

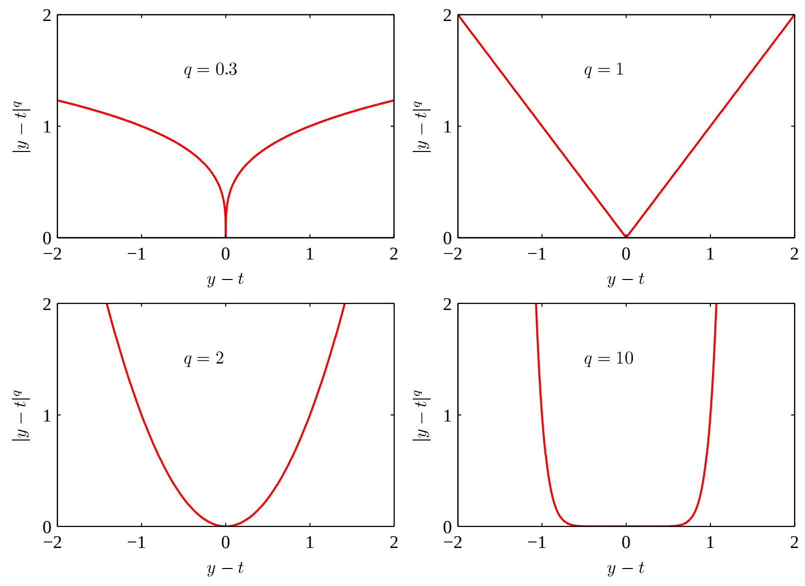

Figure 1. Regression loss functions (Bishop Figure 1.29).

Let's assume $q = 2$; to minimize, we take the derivative wrt y:

$$\begin{aligned}
\frac{\partial \mathbb{E}[\mathrm{L}]}{\partial y(\mathbf{x})} &= \int 2\left(t - y(\mathbf{x})\right)(-1)p(t,\mathbf{x})dt && (4)\\
&= \int 2y(\mathbf{x})p(t,\mathbf{x})dt - \int 2tp(t,\mathbf{x})dt && (5)\\
&= 2y(\mathbf{x})\int p(t,\mathbf{x})dt - 2\int tp(t,\mathbf{x})dt && (6)\\
&= 2y(\mathbf{x})p(\mathbf{x}) - 2\int tp(t,\mathbf{x})dt = 0 && (7)\\
2y(\mathbf{x})p(\mathbf{x}) &= 2\int tp(t,\mathbf{x})dt && (8)\\
y(\mathbf{x}) &= \frac{\int tp(t,\mathbf{x})dt}{p(\mathbf{x})} && (9)\\
&= \int t\frac{p(t,\mathbf{x})}{p(\mathbf{x})}dt && (10)\\
&= \int tp(t|\mathbf{x})dt && (11)\\
&= \mathbb{E}[t|\mathbf{x}] && (12)
\end{aligned}$$

i.e. $y(\mathbf{x}) = \mathbb{E}[t|\mathbf{x}]$ for the quadratic loss. This matches the result we found using a Gaussian likelihood and using y as the mean or **expectation** of $t|\mathbf{x}$.

3 Generalized loss functions

Other interesting prediction functions and their losses: for $q = 1$, the **conditional median** minimizes the expected loss; for $q = 0$ the **conditional mode** minimizes the expected loss (i.e. the maximum of the conditional distribution). We will see it is more common to change the order on the penalty term (the prior over weights) than the loss function itself. This corresponds to keeping a Gaussian likelihood, but putting a Laplace prior over weights (absolute values versus squared penalty with a Gaussian prior).

The general form is:

$$\mathbb{E}[\mathrm{L}] = \int\int |t - y(\mathbf{x})|^q p(t,\mathbf{x})dtd\mathbf{x} \tag{13}$$

For $q = 1$

$$\begin{aligned}
\mathbb{E}[\mathrm{L}] &= \int\int |t - y(\mathbf{x})|p(t,\mathbf{x})dtd\mathbf{x} && (14)\\
&= \int p(\mathbf{x})\left(\underbrace{\int |t - y(\mathbf{x})|p(t|\mathbf{x})dt}_{\text{mean absolute deviation}}\right)d\mathbf{x} && (15)
\end{aligned}$$

It can be shown (it is an exercise and can be googled) that the **mean absolute deviation** (or MAD) is minimized at the **median**, i.e. $y(\mathbf{x}) = \text{median}\,(p(t|\mathbf{x}))$. Although not very common, using the median prediction may be a robust alternative to $q = 2$ and may perform better for none Gaussian error distributions. This is also called **least absolute deviations regression** (the objective is the sum of absolute deviations) and **Laplace regression** (the error terms have a Laplace distribution and the MLE us the median of the Laplace distribution).

4 Approaches for minimizing loss

Based on decision theory for regression, the optimal prediction for squared loss is $\mathbb{E}\left[t|\mathbf{x}\right]$. Assuming this is our loss of interest, there are 3 approaches to produce $\mathbb{E}\left[t|\mathbf{x}\right]$; each has pros and cons and different levels of modeling requirements (Bishop p46-48):

1. Build full model $p(t, \mathbf{x})$; normalize to compute $p(t|\mathbf{x})$, then compute conditional mean $\mathbb{E}\left[t|\mathbf{x}\right]$. Pros: more flexible, can detect outliers using $p(\mathbf{x})$; cons: more demanding, must build $p(\mathbf{x})$.
2. Build conditional model only $p(t|\mathbf{x})$, then compute conditional mean $\mathbb{E}\left[t|\mathbf{x}\right]$. Pros: can use a Bayesian approach for prediction, could use full conditional (not just expectation); cons: point estimates are similar to c), so no real cons wrt c.
3. Build $y(\mathbf{x})$ directly from data. Pros: equivalent to b) for many situations; cons: cannot do the pros of b).

5 The irreducible loss

Even if we produce the best possible regression model, i.e. with very good generalization performance, the inherent noise (it is random, we cannot identify it) in the data make it impossible to achieve zero loss on held-out data. To see this, we can make use of out previous result, that the optimal decision is $\mathbb{E}\left[t|\mathbf{x}\right]$, and rewrite the squared loss L:

$$\begin{aligned}(y(\mathbf{x}) - t)^2 &= (y(\mathbf{x}) - \mathbb{E}\left[t|\mathbf{x}\right] + \mathbb{E}\left[t|\mathbf{x}\right] - t)^2 \quad (16)\\ &= (y(\mathbf{x}) - \mathbb{E}\left[t|\mathbf{x}\right])^2 + (\mathbb{E}\left[t|\mathbf{x}\right] - t)^2 + 2\left(y(\mathbf{x}) - \mathbb{E}\left[t|\mathbf{x}\right]\right)\left(\mathbb{E}\left[t|\mathbf{x}\right] - t\right) \quad (17)\end{aligned}$$

Now take the expectation for each of the three terms:

$$\begin{aligned}\mathbb{E}\left[\text{L}\right] = &\int\int (y(\mathbf{x}) - \mathbb{E}\left[t|\mathbf{x}\right])^2 p(t, \mathbf{x}) d\mathbf{x} dt \quad (18)\\ &+ \int\int (\mathbb{E}\left[t|\mathbf{x}\right] - t)^2 p(t, \mathbf{x}) d\mathbf{x} dt \quad (19)\\ &+ \int\int 2\left(y(\mathbf{x}) - \mathbb{E}\left[t|\mathbf{x}\right]\right)\left(\mathbb{E}\left[t|\mathbf{x}\right] - t\right) p(t, \mathbf{x}) d\mathbf{x} dt \quad (20)\end{aligned}$$

First term of $\mathbb{E}[\mathbf{L}]$

$$\int\int \underbrace{\left(y(\mathbf{x}) - \mathbb{E}[t|\mathbf{x}]\right)^2}_{\text{only depends on } \mathbf{x}} d\mathbf{x}dt = \int \left(y(\mathbf{x}) - \mathbb{E}[t|\mathbf{x}]\right)^2 p(\mathbf{x})d\mathbf{x} \tag{21}$$

Second term of $\mathbb{E}[\mathbf{L}]$

$$\begin{aligned}
\int\int \left(\mathbb{E}[t|\mathbf{x}] - t\right)^2 p(t,\mathbf{x})d\mathbf{x}dt &= \int\int \left(\mathbb{E}[t|\mathbf{x}] - t\right)^2 \underbrace{p(t|\mathbf{x})p(\mathbf{x})}_{=p(t,\mathbf{x})} d\mathbf{x}dt &(22)\\
&= \int \left[\left(\mathbb{E}[t|\mathbf{x}] - t\right)^2 p(t|\mathbf{x})dt\right] p(\mathbf{x})d\mathbf{x} &(23)\\
&= \int \underbrace{\left[\left(t - \mathbb{E}[t|\mathbf{x}]\right)^2 p(t|\mathbf{x})dt\right]}_{=\mathbb{V}(t|\mathbf{x})} p(\mathbf{x})d\mathbf{x} &(24)\\
&= \int \mathbb{V}(t|\mathbf{x})p(\mathbf{x})d\mathbf{x} &(25)
\end{aligned}$$

Third term of $\mathbb{E}[\mathbf{L}]$

$$\int\int 2\left(y(\mathbf{x}) - \mathbb{E}[t|\mathbf{x}]\right)\left(\mathbb{E}[t|\mathbf{x}] - t\right)p(t,\mathbf{x})d\mathbf{x}dt \tag{26}$$

$$= 2\int\int \left[y(\mathbf{x})\,\mathbb{E}[t|\mathbf{x}] - y(\mathbf{x})t + \mathbb{E}[t|\mathbf{x}]\,t - \mathbb{E}[t|\mathbf{x}]^2\right] p(t,\mathbf{x})d\mathbf{x}dt \tag{27}$$

$$= 2\int \left[y(\mathbf{x})\,\mathbb{E}[t|\mathbf{x}]\,p(\mathbf{x}) - y(\mathbf{x}) \underbrace{\int tp(t,\mathbf{x})dt}_{p(\mathbf{x})\underbrace{\int tp(t|\mathbf{x})dt}_{\mathbb{E}[t|\mathbf{x}]}} + \underbrace{\int \mathbb{E}[t|\mathbf{x}]\,tp(t,\mathbf{x})dt}_{\mathbb{E}[t|\mathbf{x}]p(\mathbf{x})\underbrace{\int tp(t|\mathbf{x})dt}_{\mathbb{E}[t|\mathbf{x}]}} - \mathbb{E}[t|\mathbf{x}]^2 p(\mathbf{x}) \right] d\mathbf{x} \tag{28}$$

$$= 2\int \left[y(\mathbf{x})\,\mathbb{E}[t|\mathbf{x}]\,p(\mathbf{x}) - y(\mathbf{x})p(\mathbf{x})\,\mathbb{E}[t|\mathbf{x}] + \mathbb{E}[t|\mathbf{x}]^2 p(\mathbf{x}) - \mathbb{E}[t|\mathbf{x}]^2 p(\mathbf{x})\right] d\mathbf{x} \tag{29}$$

$$= 0 \tag{30}$$

Back to $\mathbb{E}[\mathbf{L}]$ To understand this result, you must think of $\mathbb{E}[t|\mathbf{x}]$ as the **true** conditional expectation function (not one based on a single data set). We can therefore see that our expected loss has two parts. The first part depends on our function y, which can, in principle, learn the true function, and therefore can be driven to 0. The second part does not depend on y, and is the conditional variance of the problem itself. Since this is independent of our model, this is the **irreducible error** of the data.

$$\mathbb{E}\left[\mathrm{L}\right] = \underbrace{\int \left(y(\mathbf{x}) - \mathbb{E}\left[t|\mathbf{x}\right]\right)^2 p(\mathbf{x})d\mathbf{x}}_{\text{model error can go to 0}} + \underbrace{\int \mathbb{V}(t|\mathbf{x})p(\mathbf{x})d\mathbf{x}}_{\text{irreducible error}} \tag{31}$$

Therefore due to **intrinsic** noise in the data, our expected loss should be greater than 0. This is noise that we **cannot predict** with the information / features of the data we are given.

6 The Bias-Variance trade-off

We now present a **frequentist** view of model complexity. Overfitting is common in machine learning, but is harder to do with a Bayesian approach because we can integrate over parameters (and therefore their uncertainty). Here, taking a frequentist view, we assume the parameters are estimated from the data with no uncertainty, and it is the observed data that has uncertainty. What happens when we consider all possible data sets for a particular problem?

Earlier we derived the irreducible loss:

$$\mathbb{E}\left[\mathrm{L}\right] = \underbrace{\int \left(y(\mathbf{x}) - \mathbb{E}\left[t|\mathbf{x}\right]\right)^2 p(\mathbf{x})d\mathbf{x}}_{\text{model error can go to 0}} + \underbrace{\int \mathbb{V}(t|\mathbf{x})p(\mathbf{x})d\mathbf{x}}_{\text{irreducible error}} \tag{32}$$

The prediction model implicitly conditioned on a single data set $\mathcal{D}$; i.e. $y(\mathbf{x}, \mathcal{D})$. To be consistent with Bishop, we will use $h(\mathbf{x}) = \mathbb{E}\left[t|\mathbf{x}\right]$ to represent the **unknown**, but **true** underlying function of $\mathbf{x}$ that is **independent** of any data set.

$$\mathbb{E}\left[\mathrm{L}(\mathbf{x})|\mathcal{D}\right] = \underbrace{\int \left(y(\mathbf{x}, \mathcal{D}) - h(\mathbf{x})\right)^2 p(\mathbf{x})d\mathbf{x}}_{\mathrm{L}(\mathbf{x},\mathcal{D})} + \int \sigma^2(\mathbf{x})p(\mathbf{x})d\mathbf{x} \tag{33}$$

We will now decompose $\mathrm{L}(\mathbf{x}, \mathcal{D})$ and then take the expectation over all possible data sets generated by the underlying problem / data generating mechanism.

$$\begin{aligned}
\mathrm{L}(\mathbf{x}, \mathcal{D}) &= \left(y(\mathbf{x}, \mathcal{D}) - h(\mathbf{x})\right)^2 \\
&= \left(\underbrace{y(\mathbf{x}, \mathcal{D})}_{\text{model using } \mathcal{D}} \overbrace{- \underbrace{\mathbb{E}_{\mathcal{D}}\left[y\left(\mathbf{x}, \mathcal{D}\right)\right]}_{\text{expected model}} + \mathbb{E}_{\mathcal{D}}\left[y\left(\mathbf{x}, \mathcal{D}\right)\right]}^{\text{same trick for } \mathbb{E}[\mathrm{L}]} - \underbrace{h(\mathbf{x})}_{\text{true function}} \right)^2 \\
&= \left(y(\mathbf{x}, \mathcal{D}) - \mathbb{E}_{\mathcal{D}}\left[y\left(\mathbf{x}, \mathcal{D}\right)\right]\right)^2 + \left(\mathbb{E}_{\mathcal{D}}\left[y\left(\mathbf{x}, \mathcal{D}\right)\right] - h(\mathbf{x})\right)^2 \\
&\quad + 2\left(y(\mathbf{x}, \mathcal{D}) - \mathbb{E}_{\mathcal{D}}\left[y\left(\mathbf{x}, \mathcal{D}\right)\right]\right)\left(\mathbb{E}_{\mathcal{D}}\left[y\left(\mathbf{x}, \mathcal{D}\right)\right] - h(\mathbf{x})\right)
\end{aligned}$$

Next we take the expectation of $\mathrm{L}(\mathbf{x},\mathcal{D})$ over data sets. First, we show that the third term cancels:

$$\begin{aligned}
&\int 2\left(y(\mathbf{x},\mathcal{D}) - \mathbb{E}_{\mathcal{D}}\left[y\left(\mathbf{x},\mathcal{D}\right)\right]\right)\left(\mathbb{E}_{\mathcal{D}}\left[y\left(\mathbf{x},\mathcal{D}\right)\right] - h(\mathbf{x})\right) p(\mathcal{D}) d\mathcal{D} \\
&\quad = 2\,\mathbb{E}_{\mathcal{D}}\left[\left(y(\mathbf{x},\mathcal{D}) - \mathbb{E}_{\mathcal{D}}\left[y\left(\mathbf{x},\mathcal{D}\right)\right]\right)\left(\mathbb{E}_{\mathcal{D}}\left[y\left(\mathbf{x},\mathcal{D}\right)\right] - h(\mathbf{x})\right)\right] \\
&\quad = \mathbb{E}_{\mathcal{D}}\left[y(\mathbf{x},\mathcal{D})\,\mathbb{E}_{\mathcal{D}}\left[y\left(\mathbf{x},\mathcal{D}\right)\right] - y(\mathbf{x},\mathcal{D})h(\mathbf{x}) - \mathbb{E}_{\mathcal{D}}\left[y\left(\mathbf{x},\mathcal{D}\right)\right]^2 + \mathbb{E}_{\mathcal{D}}\left[y\left(\mathbf{x},\mathcal{D}\right)\right]h(\mathbf{x})\right] \\
&\quad = \mathbb{E}_{\mathcal{D}}\left[y\left(\mathbf{x},\mathcal{D}\right)\right]\mathbb{E}_{\mathcal{D}}\left[y\left(\mathbf{x},\mathcal{D}\right)\right] - \mathbb{E}_{\mathcal{D}}\left[y\left(\mathbf{x},\mathcal{D}\right)\right]h(\mathbf{x}) - \mathbb{E}_{\mathcal{D}}\left[y\left(\mathbf{x},\mathcal{D}\right)\right]^2 + \mathbb{E}_{\mathcal{D}}\left[y\left(\mathbf{x},\mathcal{D}\right)\right]h(\mathbf{x}) \\
&\quad = 0
\end{aligned}$$

The expectation over datasets for the first two terms is:

$$\begin{aligned}
\mathbb{E}_{\mathcal{D}}\left[\mathrm{L}(\mathbf{x},\mathcal{D})\right] &= \mathbb{E}_{\mathcal{D}}\left[\left(y(\mathbf{x},\mathcal{D}) - \mathbb{E}_{\mathcal{D}}\left[y\left(\mathbf{x},\mathcal{D}\right)\right]\right)^2\right] + \mathbb{E}_{\mathcal{D}}\left[\underbrace{\left(\mathbb{E}_{\mathcal{D}}\left[y\left(\mathbf{x},\mathcal{D}\right)\right] - h(\mathbf{x})\right)^2}_{\text{constant wrt } \mathcal{D}}\right] \\
&= \underbrace{\mathbb{E}_{\mathcal{D}}\left[\left(y(\mathbf{x},\mathcal{D}) - \mathbb{E}_{\mathcal{D}}\left[y\left(\mathbf{x},\mathcal{D}\right)\right]\right)^2\right]}_{\text{variance of predictions @ } \mathbf{x}} + \underbrace{\left(\mathbb{E}_{\mathcal{D}}\left[y\left(\mathbf{x},\mathcal{D}\right)\right] - h(\mathbf{x})\right)^2}_{(\text{bias})^2 \text{ of predictions @ } \mathbf{x}}
\end{aligned}$$

Now we take the expectation over the entire loss (including intrinsic noise) and also average over $\mathbf{x}$ using $p(\mathbf{x})$:

$$\begin{aligned}
\mathbb{E}\left[\mathbb{E}_{\mathcal{D}}\left(\mathrm{L}(\mathbf{x})\right)\right] = &\int \mathbb{E}_{\mathcal{D}}\left[\left(y(\mathbf{x},\mathcal{D}) - \mathbb{E}_{\mathcal{D}}\left[y\left(\mathbf{x},\mathcal{D}\right)\right]\right)^2\right] p(\mathbf{x}) d\mathbf{x} + \\
&\int \left(\mathbb{E}_{\mathcal{D}}\left[y\left(\mathbf{x},\mathcal{D}\right)\right] - h(\mathbf{x})\right)^2 p(\mathbf{x}) d\mathbf{x} + \\
&\int \sigma^2(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} \\
&= \text{variance} + (\text{bias})^2 + \text{noise}
\end{aligned}$$

6.1 How does $\boldsymbol{\lambda}$ effect the bias-variance?

Recall: using Gaussian basis features with L2 penalty over weights using λ.

What is the effect of λ? Draw $L = 100$ datasets. For each dataset learn the model $y(\mathbf{x},\mathcal{D})$ (which is implicitly conditioned on λ). Now average the predictions across all the datasets producing $\mathbb{E}_{\mathcal{D}}\left[y(\mathbf{x},\mathcal{D})\right]$. The dataset predictions and the expectation can be used to compute the empirical variance, and the expectation and the true function can be used to compute the bias (for a given input vector). We can compute the total bias and variance by summing over the test set (rather than the integral over $\mathbf{x}$). The result, plotted for each value of $\log\lambda$, is shown in Figure 2 (Why is there a gap between the test error and the sum of the variance and bias term?).

Low bias models are able to capture part of the noise in the data, and are thus high variance because for different data sets the noise they capture will be

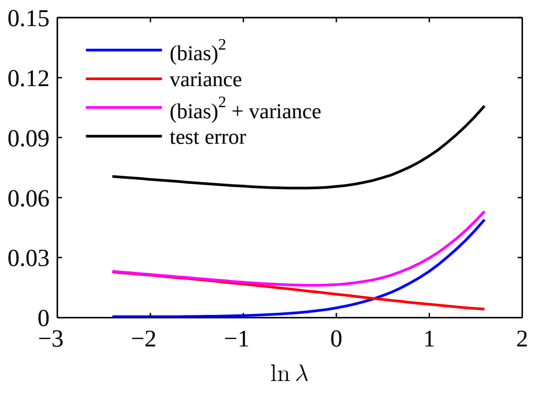

Figure 2. Empirically computed bias-variance trade-off (Bishop Figure 3.6). We can make several remarks about the bias-variance trade-off by considering three different values of $\ln \lambda$. 1) at $\ln \lambda = -2$: there is virtually no penalty on the weights; it can fit **all** data sets **perfectly** (and therefore fits the noise exactly); solutions vary wildly, but on average the prediction is good (i.e. has low bias); the error from the variance is only slightly worse than at the optimum $\ln \lambda^\star$. 2) at $\ln \lambda = 1.5$: there is a very strong penalty on the weights, and the model **cannot fit any** data set well (it underfits); solutions are all the same, but are poor; since the solutions are all the same, the variance is near 0, and all non-irreducible error is from the bias. 3) at $\ln \lambda^\star$: the sweet-spot and the optimal empirical trade-off; also where the test error is minimized (no surprise since we are averaging over many datasets).

different. On the other hand, high-bias models usually extract some but not all of the structure from the data and this regularity will appear in the different data sets, so that their solutions will look similar and thus have low-variance. Examples of this using λ to control the complexity are shown in Figure 3. These different cases should tell us that we should not avoid complex models because of overfitting, but instead harness their capacity. How do we do this in practice? **Bagging**: use bootstrapped data sets, then average; **ensemble methods**: at end of course; **Bayesian**: use a predictive distribution.

7 Stochastic Gradient Descent

For very large datasets, the **batch** update solution for $\mathbf{w}_{\text{MLE}}$ and $\mathbf{w}_{\text{MAP}}$ may be too expensive. This may be more likely the case for more complex models, such as Neural Networks, and others where the cost function decomposes into a sum over data cases:

$$\begin{aligned}\underset{\mathcal{D}}{\mathbb{E}}(\mathbf{w}) &= \sum_{n=1}^{N} e_n \\ e_n &= \frac{1}{2}\left(t_n - \mathbf{w}^T \boldsymbol{\phi}_n\right)^2\end{aligned}$$

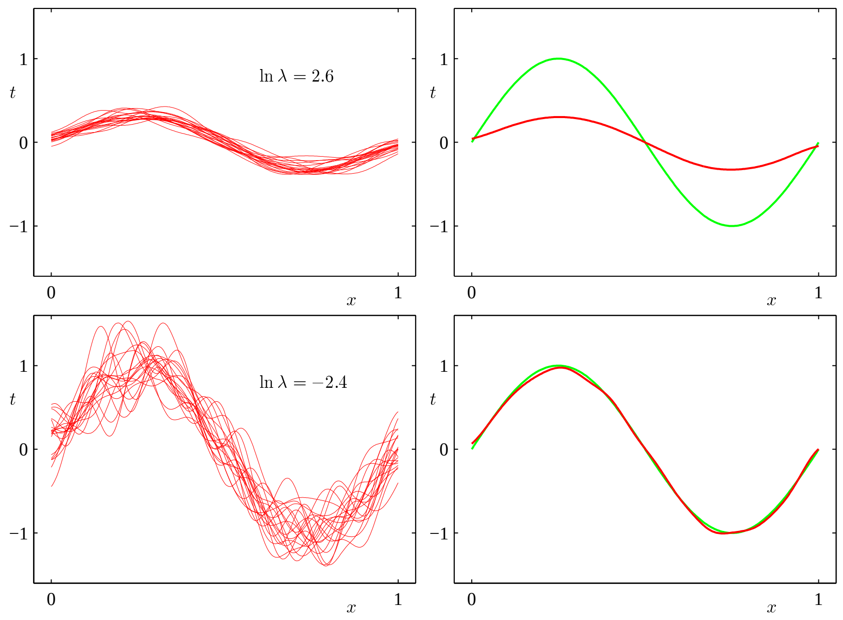

Figure 3. Examples of bias-variance using λ to control complexity (Bishop Figure 3.5).

where e_n is the individual error term and we have shown it for the sum-squares objective. The gradient also decomposes into a sum over individual gradients ∇e_n. We can then use the following **sequential** or **online** update algorithm, called **stochastic gradient descent**:

$$\begin{aligned}\mathbf{w}^{(t+1)} &= \mathbf{w}^{(t)} - \gamma^{(t)} \nabla e_i \\ &= \mathbf{w}^{(t)} + \gamma^{(t)} (t_i - \mathbf{w}^{(t)T} \boldsymbol{\phi}_i) \boldsymbol{\phi}_i\end{aligned}$$

where $i \sim \mathcal{U}(1, N)$, i.e. at each iteration, a single data vector is chosen at random from the data set and its gradient is used to update $\mathbf{w}$. SGD has variations where a batch of gradients are used (instead of just one). The algorithm is guaranteed to converge under mild conditions, especially the learning rate decays to 0 as $t \to \infty$.