CS5339 Lecture Notes #4:
Linear Regression

Jonathan Scarlett

April 3, 2021

Useful references:

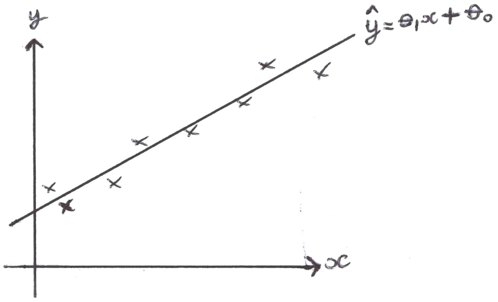
- Blog post by Jeremy Kun[1]
- Slide set `lecture_bo0.pdf` from a one-day course I gave[2]
- MIT lecture notes,[3] lecture 5
- Chapter 3 of Bishop's "Pattern Recognition and Machine Learning" book
- Section 9.2 of "Understanding Machine Learning" book

1 Linear Prediction

- In previous lectures, we looked at predicting binary labels $y_t \in \{-1, 1\}$. This is relevant in trying to learn "yes/no" questions (e.g., is this a spam email?) Here, we switch to the scenario where $y_t \in \mathbb{R}$. This is relevant when trying to predict (continuous) real-valued quantities (e.g., a stock price).
- We initially focus on *linear predictors* of the form

$$\hat{y}(\mathbf{x}) = \boldsymbol{\theta}^T \mathbf{x} + \theta_0 \tag{1}$$

for some $\boldsymbol{\theta} \in \mathbb{R}^d$ and $\theta_0 \in \mathbb{R}$. Non-linear predictors will be handled in later lectures.



[1] `http://jeremykun.com/2013/08/18/linear-regression/`

[2] `https://www.comp.nus.edu.sg/~scarlett/gp_slides`

[3] `http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-867-machine-learning-fall-2006/lecture-notes/`

- As with the binary setting, we can derive predictors via several approaches:

  1. Consider $\mathcal{D} = \{(\mathbf{x}_t, y_t)\}_{t=1}^n$ as simply being given, and try to find a predictor that fits the data.
  2. Model each $(\mathbf{x}_t, y_t)$ as being independently drawn from a distribution $P(\mathbf{x})P(y|\mathbf{x})$ parametrized by $(\boldsymbol{\theta}, \theta)$, and estimate these parameters using maximum likelihood.
  3. (Bayesian view) Model both the data and the parameters as random, so we have distributions $P(\mathbf{x})P(y|\mathbf{x})$ and $P(\boldsymbol{\theta}, \theta_0)$.

  We start with the second case, but we will quickly see that the resulting estimates $(\hat{\boldsymbol{\theta}}, \hat{\theta})$ have a natural interpretation in the first case. The Bayesian view is discussed at the end of the lecture.

- Motivating example:

  – Suppose we have a list of 1000 days' stock prices, and we want to train a regression algorithm that takes 10 consecutive days as input ($\mathbf{x}$), and outputs the prediction for the next day (y).

  – We can construct a data set $\mathcal{D} = \{(\mathbf{x}_t, y_t)\}_{t=1}^n$ as follows: (i) Let $\mathbf{x}_1 \in \mathbb{R}^{10}$ contain the first 10 prices, and y_1 be the 11th; (ii) Let $\mathbf{x}_2 \in \mathbb{R}^{10}$ contain the prices 2–11, and y_2 be the 12th; (iii) etc.

  – A linear model is reasonable, because it captures rules like "predict the next price to be the current price + the average increase of the 9 days before that".

- **Reminder:** *All models are wrong, but some models are useful*

2 Gaussian Model

Model and noise distribution.

- Consider a probabilistic model in which y_t is generated from $\mathbf{x}_t$ according to

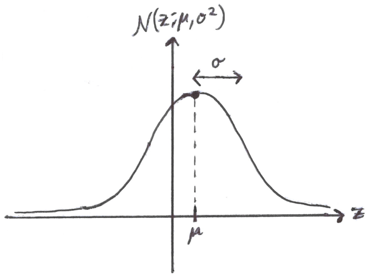
$$y_t = (\boldsymbol{\theta}^*)^T \mathbf{x}_t + \theta_0^* + z_t, \tag{2}$$

  where $(\boldsymbol{\theta}^*, \theta_0^*)$ are fixed and unknown, and z_t is random noise. (Included on the basis that we can rarely measure anything in the real world perfectly)

- The most widely-adopted noise distribution is Gaussian: $z_t \sim \mathcal{N}(0, \sigma^2)$. Recall that the PDF of a Gaussian is

$$\mathcal{N}(z; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(- \frac{(z-\mu)^2}{2\sigma^2} \right).$$

  An illustration:



- Under such a noise model, we see from (2) that

$$P(y|\mathbf{x}) = \mathcal{N}(y; \boldsymbol{\theta}^T\mathbf{x} + \theta_0, \sigma^2).$$

Again, we sometimes make the dependence on $(\boldsymbol{\theta}, \theta_0)$ and σ^2 explicit by writing $P(y|\mathbf{x})$ as $P(y|\mathbf{x}; \boldsymbol{\theta}, \theta_0, \sigma^2)$.

Maximum likelihood estimation.

- Suppose the data set $\mathcal{D} = \{(\mathbf{x}_t, y_t)\}_{t=1}^n$ is known to consist of independent samples generated via (2).
- Since we don't know σ^2, we can treat it as an additional parameter to be estimated along with $(\boldsymbol{\theta}, \theta_0)$. The likelihood function is then

$$L(\boldsymbol{\theta}, \theta_0, \sigma^2; \mathcal{D}) = \prod_{t=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp\bigg(-\frac{(y_t - \boldsymbol{\theta}^T\mathbf{x}_t - \theta_0)^2}{2\sigma^2}\bigg),$$

where the product $\prod_{t=1}^n$ is due to the assumption of independent data samples.

- Maximizing L is equivalent to maximizing its log, but the latter is more convenient to work with:

$$\log L(\boldsymbol{\theta}, \theta_0, \sigma^2; \mathcal{D}) = \text{const.} - \frac{n}{2}\log\sigma^2 - \frac{1}{2\sigma^2}\sum_{t=1}^n (y_t - \boldsymbol{\theta}^T\mathbf{x}_t - \theta_0)^2, \qquad (3)$$

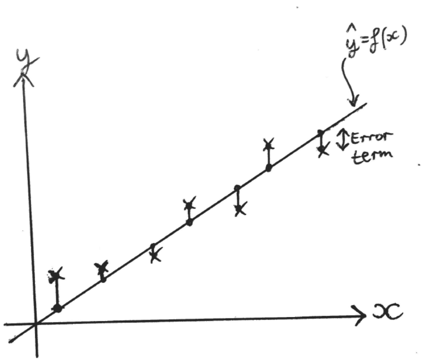
where const. represents a term that does not depend on $(\boldsymbol{\theta}, \theta_0, \sigma^2)$.

- We now notice that in this case there is no need to explicitly estimate σ^2; no matter what its value is, the maximum likelihood (ML) estimate of $(\boldsymbol{\theta}, \theta_0)$ is

$$(\hat{\boldsymbol{\theta}}, \hat{\theta}_0) = \arg\max_{\boldsymbol{\theta},\theta_0} \log L(\boldsymbol{\theta}, \theta_0, \sigma^2|\mathcal{D}) = \arg\min_{\boldsymbol{\theta},\theta_0} \sum_{t=1}^n (y_t - \boldsymbol{\theta}^T\mathbf{x}_t - \theta_0)^2. \qquad (4)$$

This is known as the *least squares estimate.*

  - Naturally, once these estimates are computed, the prediction rule for a new point $\mathbf{x}'$ is given by $\hat{y}(\mathbf{x}') = \hat{\boldsymbol{\theta}}^T\mathbf{x}' + \hat{\theta}_0$. The least squares rule is trying to minimize the sum of squared error terms (squares of the differences between the predictions and the actual labels):



- If we had assumed non-Gaussian noise, the ML estimate would have been different (and possibly more complicated).

3 Finding the Least Squares Estimate

- Like with logistic regression, we could try to solve (4) using stochastic gradient descent (in fact, this is often the best way to go for huge data sets!)
- But in this particular case, we can actually find a closed-form solution.
- First, let's switch to matrix notation:

$$\sum_{t=1}^{n}(y_t - \boldsymbol{\theta}^T \mathbf{x}_t - \theta_0)^2 = \|\mathbf{y} - \mathbf{X}\boldsymbol{\Theta}\|^2,$$

where $\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \in \mathbb{R}^n$, $\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T & 1 \\ \vdots & \vdots \\ \mathbf{x}_n^T & 1 \end{bmatrix} \in \mathbb{R}^{n\times(d+1)}$, and $\boldsymbol{\Theta} = \begin{bmatrix} \boldsymbol{\theta} \\ \theta_0 \end{bmatrix} \in \mathbb{R}^{d+1}$.

- Using basic vector calculus (which you don't need to know), the derivative of $\|\mathbf{y} - \mathbf{X}\boldsymbol{\Theta}\|^2$ with respect to $\boldsymbol{\Theta}$ is $2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\boldsymbol{\Theta})$. Setting this to zero and solving for $\boldsymbol{\Theta}$ gives

$$\hat{\boldsymbol{\Theta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}. \tag{5}$$

The matrix $(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ is known as the *pseudo-inverse* of $\mathbf{X}$ (it is easy to check that it equals $\mathbf{X}^{-1}$ whenever $\mathbf{X}$ is square and invertible).

- Remarks.
  - The estimate $\hat{\boldsymbol{\Theta}} = \begin{bmatrix} \hat{\boldsymbol{\theta}} \\ \hat{\theta}_0 \end{bmatrix}$ is a linear function of $\mathbf{y}$ (but the dependence on $\mathbf{X}$ is non-linear)
  - We have implicitly assumed that $\mathbf{X}^T\mathbf{X}$ is invertible, which is usually OK when $n > d$ (more equations than unknowns) but not when $d < n$ (the "high-dimensional" setting).
- Once we have $(\hat{\boldsymbol{\theta}}, \hat{\theta}_0)$, we can substitute back into (3) and compute the ML estimate of σ^2:

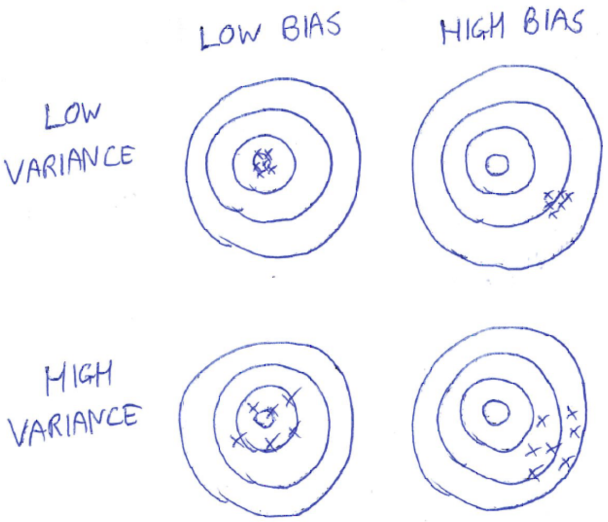
$$\hat{\sigma}^2 = \frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{\boldsymbol{\theta}}^T \mathbf{x}_t - \hat{\theta}_0)^2.$$

(Proving this requires a little calculus, taking the derivative with respect to σ^2). This does not enter into the predictor $\hat{y}$ (see (1)), but it gives a useful measure of the average prediction error for the samples in $\mathcal{D}$.

4 Bias and Variance

Illustrative picture.

- Roughly speaking, “low bias” means “correct on average”, and ”low variance” means “tending to behave similarly” (e.g., across several realizations of the random noise). An analogy in archery:



  - Note: This picture is purely for intuition and shouldn’t be viewed as a regression problem!

Motivating linear regression example.

- Suppose we are trying to predict a day’s stock price based on the 20 prior days’ prices ($d = 20$) – but we only have 20 data points ($n = 20$).
- Assuming $\mathbf{X}^T\mathbf{X}$ is invertible, we can find $\boldsymbol{\theta}$ such that $\|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|^2 = 0$ – but if the noise level is significant, this might end up being a very strange $\boldsymbol{\theta}$ amounting to “learning the noise” (e.g., $\boldsymbol{\theta} = (1, -3, 2.6, -17, 0.5, \dots)$)
  - If the noise values had been different, a very different $\boldsymbol{\theta}$ may have been chosen (*high variance*)
- Intuitively, if we could find a “simpler” $\boldsymbol{\theta}$ (e.g., $\boldsymbol{\theta} = (\frac{1}{2}, \frac{1}{3}, \frac{1}{5}, \frac{1}{8}, \dots)$) that gives $\|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|^2$ fairly small but not quite zero, we might still expect it to give better predictions for unseen $\mathbf{x}$.
- However, if we use least squares with *lots of data* (compared to the number of parameters) and/or the data is *less noisy*, then we are typically less likely to encounter this kind of problem.
  - More data $\implies$ Less risk of spurious solutions
  - More noise $\implies$ More risk of spurious solutions
- The notions of *bias* and *variance* help us understand this intuition.

Calculations for least squares.

- Continuing with matrix notation, let's write the model (2) as

$$\mathbf{y} = \mathbf{X}\mathbf{\Theta}^* + \mathbf{z}, \tag{6}$$

  where we use a superscript $(\cdot)^*$ to highlight that these are the "true" parameters. The noise vector $\mathbf{z} \in \mathbb{R}^n$ is distributed as $\mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$.

- Substituting (6) into (5) gives

$$\hat{\mathbf{\Theta}} = \mathbf{\Theta}^* + (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{z}, \tag{7}$$

  and we can interpret the right-hand side as "true value + error term".

- Since $\mathbb{E}[\mathbf{z}] = \mathbf{0}$, we immediately obtain (for fixed $\mathbf{X}$) that

$$\mathbb{E}[\hat{\mathbf{\Theta}}] = \mathbf{\Theta}^*.$$

  This means that we are "correct on average" – in statistics terminology, the estimator is *unbiased.*

- It is also easy to compute the covariance (for fixed $\mathbf{X}$):

$$\begin{aligned}
\mathrm{Cov}[\hat{\mathbf{\Theta}}] &= \mathbb{E}\big[(\hat{\mathbf{\Theta}} - \mathbf{\Theta}^*)(\hat{\mathbf{\Theta}} - \mathbf{\Theta}^*)^T\big] \\
&= \mathbb{E}\big[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{z}\mathbf{z}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\big] \\
&= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbb{E}\big[\mathbf{z}\mathbf{z}^T\big]\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1} \\
&= \sigma^2(\mathbf{X}^T\mathbf{X})^{-1},
\end{aligned}$$

  where the second-last line uses linearity of expectation (note that $\mathbf{X}$ is not random here), and the last line applies $\mathbb{E}\big[\mathbf{z}\mathbf{z}^T\big] = \sigma^2\mathbf{I}$ (since $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$) and then cancels $(\mathbf{X}^T\mathbf{X})$ with its inverse.

  This is potentially not such good news – if the matrix $(\mathbf{X}^T\mathbf{X})^{-1}$ has any large entries, the corresponding entries of $\hat{\mathbf{\Theta}}$ will have high variance.

  - If we are lucky enough to be able to choose the inputs $\mathbf{x}_1, \dots, \mathbf{x}_n$ and then observe their labels $y_1, \dots, y_n$, we could try to choose them in a way that avoids this scenario. Learning problems with this flexibility are known as *active learning.*

- **General bias vs. variance property:**

  - Consider goal of minimizing the *mean square error* (MSE) $\mathbb{E}\big[\|\hat{\mathbf{\Theta}} - \mathbf{\Theta}^*\|^2\big]$, which measures how well we estimate $\mathbf{\Theta}^*$ on average (and can be viewed as an indication of how well we will perform prediction on *unseen* data samples).
  - The MSE vector estimate $\hat{\mathbf{\Theta}}$ with true value $\mathbf{\Theta}^*$ satisfies the following:

$$\mathbb{E}\big[\|\hat{\mathbf{\Theta}} - \mathbf{\Theta}^*\|^2\big] = \underbrace{\|\mathbb{E}[\hat{\mathbf{\Theta}}] - \mathbf{\Theta}^*\|^2}_{\text{bias (squared)}} + \underbrace{\mathbb{E}\big[\|\hat{\mathbf{\Theta}} - \mathbb{E}[\hat{\mathbf{\Theta}}]\|^2\big]}_{\text{variance}} \tag{8}$$

  - No reason to believe bias = 0 is optimal! (in general, it is not – see below)
  - It is a simple exercise to prove that variance = $\mathrm{Tr}[\mathrm{Cov}[\hat{\mathbf{\Theta}}]]$ *(Hint: First get to the expression* $\mathbb{E}\big[\mathrm{Tr}[(\hat{\mathbf{\Theta}} - \mathbb{E}[\hat{\mathbf{\Theta}}])^T(\hat{\mathbf{\Theta}} - \mathbb{E}[\hat{\mathbf{\Theta}}])]\big]$*, then apply* $\mathrm{Tr}[AB] = \mathrm{Tr}[BA]$ *)*

5 Regularization and Ridge Regression

- The *ridge regression estimator* reduces variance at the expense of increasing the bias:

$$(\hat{\boldsymbol{\theta}}, \hat{\theta}_0) = \arg\min_{\boldsymbol{\theta},\theta_0} \sum_{t=1}^{n} (y_t - \boldsymbol{\theta}^T \mathbf{x}_t - \theta_0)^2 + \lambda \sum_{j=1}^{d} \theta_j^2,$$

  for some $\lambda \geq 0$ (setting $\lambda = 0$ recovers (4)).

  - Note that just like with SVM, we do not penalize θ_0.

- For notational convenience, let's focus on the case that there is no offset: $\theta_0 = 0$, and we only minimize $\sum_{t=1}^{n}(y_t - \boldsymbol{\theta}^T\mathbf{x}_t)^2 + \lambda \sum_{j=1}^{d} \theta_j^2$. In matrix form, this gives

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|^2 + \lambda \|\boldsymbol{\theta}\|^2, \quad (9)$$

  where now $\mathbf{X} \in \mathbb{R}^{n \times d}$ only has d columns; we don't append the column of 1s.

  - Returning to the motivating example in the previous section, since we penalize large values of $\|\boldsymbol{\theta}\|^2$, we are now less likely to choose the spurious solution that has large values but gives $\|\mathbf{y}-\mathbf{X}\boldsymbol{\theta}\|^2 = 0$.

- Finding the optimal $\boldsymbol{\theta}$ is done similarly to the case $\lambda = 0$, and yields:

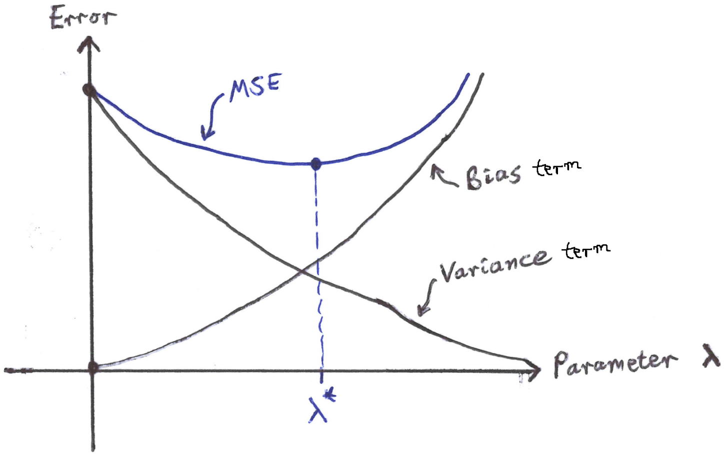
  - The closed-form solution

$$\hat{\boldsymbol{\theta}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}. \quad (10)$$

    Note that $\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}$ is *always* invertible when $\lambda > 0$.

  - Once again, the prediction rule for a new point $\mathbf{x}'$ is $\hat{y}(\mathbf{x}') = \hat{\boldsymbol{\theta}}^T \mathbf{x}'$.

- Bias-variance trade-off:



  - The bias (derived using (10), $\mathbb{E}[\mathbf{y}] = \mathbf{X}\boldsymbol{\theta}^*$, and writing $\mathbf{X}^T\mathbf{X} = \mathbf{X}^T\mathbf{X} + \lambda\mathbf{I} - \lambda\mathbf{I}$) is

$$\mathbb{E}[\hat{\boldsymbol{\theta}}] - \boldsymbol{\theta}^* = -\lambda(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\boldsymbol{\theta}^*.$$

  The eigenvalues of $\mathbf{I} - \lambda(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}$ are between 0 and 1, so on average the estimate "shrinks" the ground truth. This is to be expected given that higher values of $\|\boldsymbol{\theta}\|^2$ are penalized more.

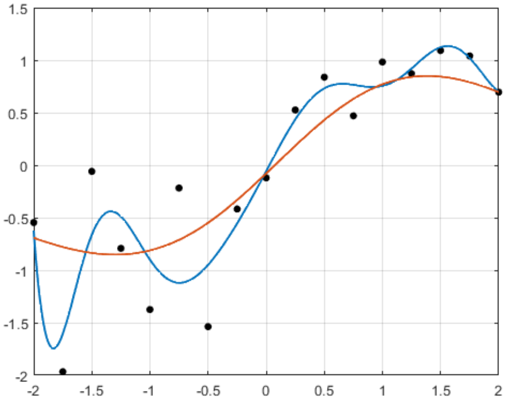
  - The covariance (requires more effort to derive) is

$$\mathrm{Cov}[\hat{\boldsymbol{\theta}}] = \sigma^2\Big((\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1} - \lambda(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-2}\Big).$$

    Taking the trace yields the variance corresponding to (8).

- We will see a simple example of the bias-variance trade-off for ridge regression in the tutorials. To gain intuition, it is easier to give an example in polynomial regression (fitting a polynomial instead of a straight line – see the next lecture on how we can still use "linear" regression techniques to do this):

  - Example curves fit to some data points:



  - The more erratic (blue) curve has no regularization, and is very sensitive in the sense that it tends to track noise in the data.
  - The less erratic (red) curve has regularization, and gives a simpler curve more aligned with the general trend of the data while being less sensitive to noise.
  - Generally speaking, regularization acts as a *stabilizer*, in the sense that it makes the output stay more similar when small changes are made to the data

- Recalling that least squares (i.e., $\lambda = 0$) is equivalent to maximum likelihood (ML) estimation under Gaussian noise, this is one of many examples showing that *ML is not always the right thing to do* (especially with limited data)

- The bias-variance trade-off is certainly not unique to linear regression and ℓ_2-regularization. Another example is the *k-nearest neighbors rule*, which (given an unseen $\mathbf{x}$) predicts y to be the average label value among the k closest points from the data set. Increasing k increases bias, but reduces variance.[4]

6 Bayesian Viewpoint

- There are (at least) two distinct viewpoints in statistics and machine learning:

[4] See `https://www.youtube.com/watch?v=n5Zxi22801Q` for a video lecture containing this example.

  - **Frequentist view.** The parameter $\boldsymbol{\theta}$ is just some fixed vector that we don't know
  - **Bayesian view.** We can encode our belief of the possible/likely values of $\boldsymbol{\theta}$ through a distribution $p(\boldsymbol{\theta})$ (e.g., $\boldsymbol{\theta} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$)

- **Bayes' rule:**

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{D})}$$

  which reads in Bayesian terminology as

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}$$

  Note that the likelihood $p(\mathcal{D}|\boldsymbol{\theta})$ should be interpreted as $p(y_1, \dots, y_n|\mathbf{x}_1, \dots, \mathbf{x}_n, \boldsymbol{\theta})$ (as opposed to a joint probability on $\mathbf{x}$'s and y's), since the input $\mathbf{x}$ is always given/known.

- Advantages and disadvantages of Bayesian methods:

  · (+) Natural way to incorporate prior knowledge

  · (+) Gives not only a prediction, but a full posterior distribution (e.g., to provide estimates of the level of (un)certainty)

  · (+) State-of-the-art performance in several applications

  · (−) Choosing a prior can be difficult

  · (−) With an incorrect prior, can have very undesirable behavior (e.g., claiming high confidence but actually being completely wrong)

  · (−) Exact posterior calculation usually impossible, need to approximate (e.g., with Monte Carlo or variational methods)

  · (−) Even with approximations, considerable computation time is often required

- Bayesian perspective on Ridge Regression:
  - A useful observation: Gaussian prior & Gaussian noise $\Longrightarrow$ Gaussian posterior
    - ∗ This is an example of so-called "conjugate priors", where the prior and posterior distributions are in the same family
  - More precise description:
    - ∗ Linear model $y_t = \boldsymbol{\theta}^T \mathbf{x}_t + z_t$ with <u>random</u> $\boldsymbol{\theta}$
    - ∗ Gaussian prior $\boldsymbol{\theta} \sim N(\mathbf{0}, \mathbf{I})$
    - ∗ Gaussian noise $z_t \sim N(0, \sigma^2)$ with independence between samples
  - Since the posterior of $\boldsymbol{\theta}$ is Gaussian, it is fully specified by its mean and covariance matrix. It can be shown (see the tutorial question) that the posterior mean is

$$\boldsymbol{\mu}_n = (\mathbf{X}^T\mathbf{X} + \sigma^2\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$$

    which is precisely ridge regression.
    - ∗ The covariance matrix also has a simple closed form (also explored in the tutorial question) – this can be used to give uncertainty estimates.