# COS513: FOUNDATIONS OF PROBABILISTIC MODELS
# LECTURE 10

MELISSA CARROLL, LINJIE LUO

## 1. BIAS-VARIANCE TRADE-OFF (CONTINUED FROM LAST LECTURE)

If $\mathcal{V} = \{(X_n, Y_n)\}$ are observed data, the linear regression problem can be modeled as:

$$Y_n | X_n, \beta \sim N(\beta X_n, \sigma^2) \tag{1}$$

$\beta X_n$ is therefore true response mean, around which we expect the observed response $Y_n$ to vary according to the Gaussian noise term. Consider a new input $X$ for which we estimate $\beta$ with the estimator $\hat{\beta}$, which we now view as a random variable, dependent on $\mathcal{V}$. The MSE (Mean Squared Error) of the estimator $\hat{\beta}$ over the distribution $\mathcal{D}$ of all inputs $X$ for which $\beta$ is the true estimated parameter is:

$$\begin{aligned} \text{MSE}(\hat{\beta}) &= \mathbb{E}_{\mathcal{D}}[(\hat{\beta}X - \beta X)^2] \\ &= \left[\mathbb{E}[(\hat{\beta}X)^2] - (\mathbb{E}[\hat{\beta}X])^2\right] + \left[(\mathbb{E}[\hat{\beta}X] - \beta X)^2\right] \end{aligned} \tag{2}$$

Eq. 2 is the sum of two terms that represent:

(1) $\mathbb{E}[(\hat{\beta}X)^2] - (\mathbb{E}[\hat{\beta}X])^2$: the variance of the estimator $\hat{\beta}$, i.e. how sensitive the estimator is to randomness in the data.
(2) $(\mathbb{E}[\hat{\beta}X] - \beta X)^2$: the squared bias of the estimator, i.e. how closely $\hat{\beta}$ approximates the true value of the parameter $\beta$.

An unbiased estimator is one for which the squared bias term is 0. In Figure 1, we view the distribution of $\hat{\beta}X$ over $\mathcal{D}$ as a Normal distribution parameterized by the two MSE terms. For an unbiased estimator, the distribution is centered at the true value $\beta X$.

The Maximum Likelihood Estimate (MLE), or Least Squares estimate, is an unbiased estimate. The **Gauss-Markov Theorem** states that among all unbiased estimates, MLE has the smallest variance. Therefore, if we wish to have an unbiased estimator, the best estimate we can choose is MLE. Our intuition should be that we should always choose an unbiased estimator. Indeed, classical statistics dealt only with unbiased estimators. However, note that the fact that an estimator is unbiased says nothing about the variance of the estimator. Thus, for a given dataset $\mathcal{V}$, the error $\hat{\beta}X - \beta X$ may in fact

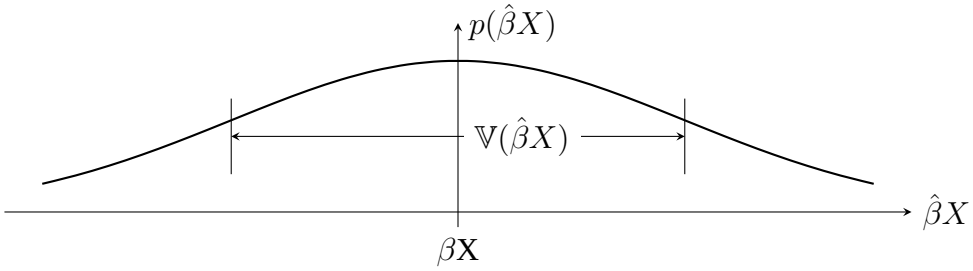


FIGURE 1. Distribution of unbiased estimator

be very large. Therefore, an estimator with slight bias but small variance will be preferable to an unbiased estimator with a very large variance. The remainder of this lecture will discuss how modern statistics allows for this tradeoff between bias and variance.

1.1. **Regularization.** In regression, this trade-off is made through *regularization*, which:

- Involves placing a constraint on $\hat{\beta}$.
- Encourages "smaller" and "simpler" models because the space of values of $\hat{\beta}$ considered is smaller.
- Intuitively, prevents *overfitting* to the training data, leading to better generalization.
- Aids model interpretation by producing "simpler" models (although attempting to interpret $\beta$ weights should often be avoided).

## 2. RIDGE REGRESSION

The most popular form of regularized regression is **Ridge Regression**, which places a constraint on the sum of squares of the $\beta$ weights. Formally, Ridge optimizes the Residual Sum of Squares (RSS) subject to a constraint on $\sum_{i=1}^{p} \beta_i^2$:

$$\min \sum_{n=1}^{N} (y_n - \beta^T x_n)^2 \quad \text{s.t.} \quad \sum_{i=1}^{p} \beta_i^2 < s \tag{3}$$

We visualize the Ridge optimization in Figure 2. Assume $x_n$ is a two-dimensional vector, i.e. $p = 2$. First, consider the RSS term in Eq. 3. The MLE estimate $\hat{\beta}$ will lie at a point in the two-dimensional coefficient space. The RSS at this point is $\sum_{n=1}^{N} (y_n - \hat{\beta}^T x_n)^2$ . Likewise, all other points $\tilde{\beta}$ in the coefficient space have an RSS of $\sum_{n=1}^{N} (y_n - \tilde{\beta}^T x_n)^2$ . In fact, for all $\tilde{\beta}$ other than $\hat{\beta}$, there will be an infinite set of points with the same RSS as $\tilde{\beta}$, and these points will lie on an ellipse. We can thus plot the contours

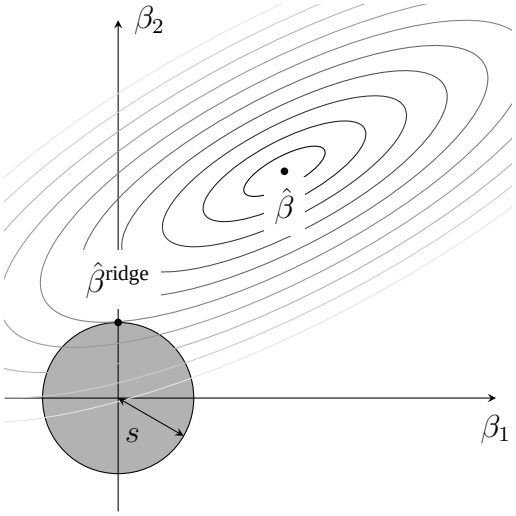


FIGURE 2. Illustration of the Ridge regression optimization when $p = 2$. The concentric ellipses are contours of $\beta$ with equal RSS. Sphere $s$ constrains the search space for $\beta$. Ridge solves for the value of $\beta$ with minimal RSS among all $\beta$ values lying within $s$.

of RSS values emanating from $\hat{\beta}$ as nested ellipses. Since these contours correspond to increasing RSS values, they also correspond to increasing *biases* of their associated estimates.

Now consider the constraint term in Eq. 3. $\sum_{i=1}^{p} \beta_i^2$ is a measure of the Euclidean distance from the origin to $\beta$, and the constraint $< s$ dictates the radius of the circle in which $\beta$ is constrained to lie. Thus, when we solve Eq. 3, we are seeking the value $\tilde{\beta}$ within the sphere of radius $s$ with minimal RSS, i.e. the point in sphere $s$ that lies on the contour ellipse closest to $\hat{\beta}$. This point will be a unique point at which the edge of $s$ touches a contour ellipse, i.e. the point on sphere $s$ closest to $\hat{\beta}$. Because we have limited the range of $\beta$ values being considered, the *variance* of our estimate will necessarily be smaller than when considering the full range of $\beta$ values. If $\hat{\beta}$ lies within $s$, the Ridge estimate is equivalent to the MLE. In all other cases, however, the resulting estimate will have higher bias than the MLE but smaller variance, which is precisely the effect we are seeking. As $s$ is increased, the estimate bias will decrease but the variance will increase, and vice versa.

We can solve for the estimate $\hat{\beta}^{\text{ridge}}$ directly with the following constrained optimization, where $\lambda$ represents a complexity parameter, sometimes called the Ridge, or $L_2$ Penalty:

$$\hat{\beta}^{\text{ridge}} = \arg\min_{\beta} \sum_{n=1}^{N} \frac{1}{2}(y_n - \beta^T x_n)^2 + \lambda \sum_{i=1}^{p} \beta_i^2 \tag{4}$$

The $1/2$ is added for mathematical convenience when minimizing Eq. 4.

There are two nice things about Eq. 4:

- For a fixed value of $\lambda$, this equation is convex and therefore easy to minimize, which is the major reason why Ridge is the most popular regularization technique for regression.
- There is essentially a one-to-one mapping between $\lambda$ and $s$, such that when $\lambda$ increases, $s$ effectively decreases. (Note: technically this mapping depends on the the number of data points $N$, such that as $N$ increases, $s$ effectively increases to accomodate the larger dataset. This subtlety is important when considering Bayesian linear regression.)

## 3. Choosing $\lambda$ via Cross-Validation

Despite these niceties, by introducing $\lambda$, we have added an additional parameter to be optimized. So how do we solve for the optimal $\lambda$? A first inclination might be to simply use MLE, just like we do with $\beta$. Unfortunately, since MLE seeks to minimize the RSS, the optimal value for $\lambda$ will always be $0$, rendering useless our attempts at regularization. Intuitively, by regularizing, we are hoping to improve the *generalization* of our model to other datasets $\mathcal{V}$. Thus, a natural way to optimize $\lambda$ is to train models with different values of $\lambda$ and evaluate the error of these models on a different dataset we believe to be drawn from the same distribution as the dataset used for training our model (*generalization error*). This procedure is commonly used in model-fitting and is called *cross-validation*. The cross-validation procedure is as follows:

(1) Choose candidate values for $\lambda$.
(2) Divide the data $(X_n, Y_n)$ into $K$ folds.
(3) For each fold $k$ and candidate $\lambda$:
    - Estimate $\hat{\beta}^{\text{ridge}}_{k,\lambda}$ on out-of-fold samples, i.e. $x_n \in j = \{1...K\}, j \neq k$.
    - Compute generalization error on in-fold samples: $\epsilon_{n,\lambda} = (y_n - \hat{\beta}^{\text{ridge}}_{k,\lambda} x_n)^2$ for $n$ in fold $k$.

    At this point, we have evaluated the error $\epsilon$ for every data point

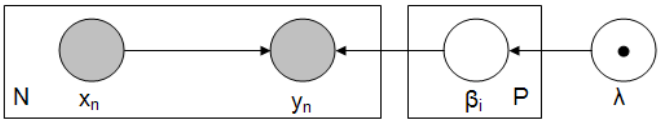


FIGURE 3. Bayesian view of regression, with $\lambda$ as a prior on $\beta$. MAP estimation under this model is equivalent to Ridge Regression.

$n$ in the data, where our estimate was computed based on data not including $n$.

- Finally, select $\lambda = \arg\min_\lambda \frac{1}{N} \sum_{n=1}^{N} \epsilon_{n,\lambda}$.

Note that cross-validation, while helping solve for one parameter, introduces another parameter in its place: $K$. In Elements of Statistical Learning, Hastie et al. conclude, after a discussion about the sensitivity of cross-validation to the choice of $K$, that one should simply choose the value $K = 5$ folds. A student also raises a cautionary note about cross-validation: avoid $x_n$ being too systematically similar to the data points in other folds, or the effects of overfitting to the training data may go unnoticed. Shuffling the examples to remove systematic biases is often warranted.

## 4. BAYESIAN LINEAR REGRESSION

Bayesian Linear Regression is closely related to Ridge, as illustrated in Figure 3.

As in our previous probabilistic view of linear regression:

$$y_n | x_n, \beta \sim N(\beta^T x_n, \sigma^2) \tag{5}$$

However, note that we have now placed a prior on the coefficients, the (fixed) parameter $\lambda$:

$$\beta_i \sim N(0, 1/2\lambda) \tag{6}$$

Consider the MAP (maximum a posteriori) estimation of $\beta$ under this model:

$$\hat{\beta}^{\text{MAP}} = \arg\max_\beta \left\{ \log P(\beta | x_{1:N}, y_{1:N}, \lambda) \right\} \tag{7}$$

Noting that because we are only considering the max, normalization constants don't matter, and we obtain, via the re-ordered chain rule:

$$\begin{aligned} \hat{\beta} &= \arg\max_\beta \left\{ \log(P(y_{1:N} | x_{1:N}, \beta) \prod_{i=1}^{p} P(\beta_i | \lambda)) \right\} \\ &= \arg\max_\beta \left\{ \log P(y_{1:N} | x_{1:N}, \beta) + \sum_{i=1}^{p} \log P(\beta_i | \lambda) \right\} \end{aligned} \tag{8}$$

Recall that, given Eq. 5:

$$
\begin{aligned}
P(y_{1:N}|x_{1:N},\beta) &= \frac{1}{(2\pi\sigma^2)^{N/2}}\exp\left\{-\frac{1}{2\sigma^2}\sum_{n=1}^{N}(y_n-\beta^T x_n)^2\right\} \\
&= \frac{1}{(2\pi\sigma^2)^{N/2}}\exp\left\{-\frac{1}{2\sigma^2}\text{RSS}(\beta)\right\}
\end{aligned} \tag{9}
$$

Again, noting that normalization constants do not matter, we have that:

$$
\arg\max_{\beta}\left\{\log P(y_{1:N}|x_{1:N},\beta)\right\} = \arg\max_{\beta}\left\{\text{-RSS}(\beta)\right\} \tag{10}
$$

Given Eq. 6:

$$
P(\beta_i|\lambda) = \frac{1}{\sqrt{2\pi/\lambda}}\exp\left\{-\frac{\beta_i^2\lambda}{2}\right\} \tag{11}
$$

Thus, given Equations 10 and 11 and again ignoring constants, we have:

$$
\hat{\beta}^{\text{MAP}} = \arg\max_{\beta}\left\{-\text{RSS}(\beta)-\lambda\sum_{i=1}^{p}\beta_i^2\right\} \tag{12}
$$

where the variance of $\beta$ is $\lambda/2$. Note that Eq. 12 takes the same form as Ridge regression. Therefore, MAP under the Bayesian model with a prior on $\beta$ of $\lambda$ is equivalent to performing Ridge regression with penalty parameter $\lambda$.

Note that as $\lambda$ increases, the more the MAP estimate diverges from the MLE, and vice versa. In effect, the $\text{RSS}(\beta)$ term corresponds to the influence of the data on the model, while the $\lambda\sum_{i=1}^{p}\beta_i^2$ term corresponds to the influence of the prior, i.e. making the variance of the estimate smaller is, in effect, indicating an increasing certainty that $\beta = 0$. As with all Bayesian models, as the influence of the prior increases, the influence of the data decreases. $\lambda$ controls this data versus prior tradeoff. As previously noted, unlike $\lambda$, $s$ grows with the size of the data. Given enough data, the influence of the data will eventually overwhelm the influence of the prior, and the sphere $s$ will grow so large as to encompass $\hat{\beta}$, making the MAP estimate equivalent to MLE. Note that we outlined a procedure for estimating $\lambda$ via cross-validation, but a true Bayesian would of course never fit $\lambda$ in such a way, because doing so involves using the data twice.

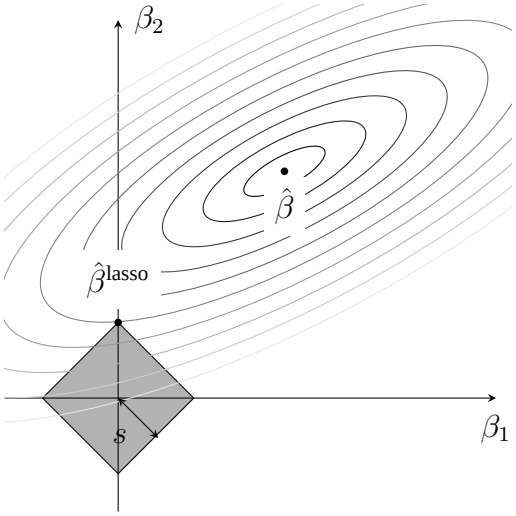


FIGURE 4. Illustration of the LASSO optimization when $p = 2$. The concentric ellipses are contours of $\beta$ with equal RSS. Polygon $s$ constrains the search space for $\beta$. LASSO solves for the value of $\beta$ with minimal RSS among all $\beta$ values lying within $s$, which will lie on a "corner" of $s$.

## 5. LASSO

Consider the following alternative regularization, which estimates $\beta$ by minimizing RSS subject to an $L_1$ norm constraint $\sum_{i=1}^{p} |\beta_i|$:

$$\min \sum_{n=1}^{N} (y_n - \beta^T x_n)^2 \quad \text{s.t.} \quad \sum_{i=1}^{p} |\beta_i| < s \tag{13}$$

This form of regularization is known as the **LASSO** (Least Absolute Shrinkage and Selection Operator):

$$\hat{\beta}^{\text{lasso}} = \arg\min_{\beta} \left\{ \sum_{n=1}^{N} \frac{1}{2} (y_n - \beta^T x_n)^2 + \lambda \sum_{i=1}^{p} |\beta_i| \right\} \tag{14}$$

In Figure 4, we visualize the LASSO optimization just as we did the Ridge optimization.

As with Ridge, the optimal $\beta$ will lie on the periphery of $s$ at the point closest to $\hat{\beta}$. Although difficult to visualize in 2 dimensions, in higher dimensionality, the RSS contours will touch a (multi-dimensional) "corner" of $s$ first (unlike with Ridge), implying that at least one coefficient is $0$. The only exception is when the contours touch at a 45 degree angle, implying colinearity in the features. Therefore, LASSO zeros out some coefficients

(particularly in higher dimensions) and finds a *sparse* solution. Note that Eq. 14 is still convex, since any penalty norm $>= 1$ is convex.

Why would we want a sparse solution?

- In many regression applications, it is known that only a subset of the features/variables will be relevant.
- A naïve approach to finding this relevant subset is to try training models with all possible subsets, which is of course intractable.
- By setting some coefficients to $0$, LASSO is in effect performing a form of feature selection, by choosing which inputs make a difference in solving the problem.
- Therefore LASSO is performing subset selection yet is convex and thus easy to optimize.
- In some cases, it can be shown that LASSO is "sparsistent," in that it will find the true relevant subset.
- Narrowing the subset of variables makes interpreting the coefficients easier.
- Sparse solutions are best if the number of variables is much greater than the number of data points, $P \gg N$.

5.1. **Bayesian Interpretation.** From our discussion of the correspondence between MAP estimation in Bayesian linear regression and Ridge, it was shown that the Ridge penalty is equivalent to assuming a Gaussian prior on $\beta$, i.e. $\beta_i \sim N$. LASSO has a similar Bayesian interpretation: the LASSO ($L_1$) penalty is equivalent to assuming a Laplace distribution of $\beta$ values, i.e. $p(\beta_i) \propto \exp\{\lambda|\beta_i|\}$ .

Note: Park and Casella discuss a Bayesian approach to solving the LASSO: Park and Casella (2008). The Bayesian LASSO. JASA 103(482).

## 6. LARS

**LARS** (Least Angle Regression) is an efficient algorithm for solving the LASSO. It computes the entire regularization path, or optimal solution for each possible number of features, in one pass, allowing the optimal size of the diamond to be easily determined using cross-validation, which is not very expensive. This one-pass regularization path discovery is what makes the LARS + LASSO combination very popular.

Efron, B., Johnstone, I., Hastie, T. and Tibshirani, R. (2003). Least angle regression. Annals of Statistics.