Linear Regression

Muchang Bahng

Spring 2024

Contents

1 Low-Dimensional Ordinary Least Squares **5**
1.1 Bias-Variance Decomposition . . . 5
1.2 Least Squares Solution . . . 8
1.3 Likelihood Estimation . . . 10
1.4 Simple Linear Regression . . . 12
1.5 Concentration Bounds . . . 13

2 Significance Tests and Confidence Sets **16**
2.1 T Test . . . 16
2.2 F Test . . . 18

3 Ridge Regression **19**
3.1 Least Squares Solution . . . 20
3.2 Bias Variance Tradeoff . . . 21
3.3 Concentration Bounds . . . 21
3.4 Tuning the Regularization Coefficient . . . 21

4 Stepwise Regression **22**
4.1 Best Subset Regression . . . 22
4.2 Forward Stepwise Regression . . . 23
4.3 Bias Variance Tradeoff . . . 24
4.4 Stagewise Regression . . . 24

5 Lasso Regression **25**
5.1 Bias Variance Tradeoff . . . 26
5.2 Concentration Bounds . . . 26
5.3 Optimization . . . 26

6 Robust Regression **27**

7 Bayesian Linear Regression **28**
7.1 Regularization with Priors . . . 28

8 Generalized Linear Models **30**
8.1 Exponential Family . . . 32
8.1.1 Canonical Exponential Family . . . 34
8.2 Cumulant Generating Function . . . 37
8.3 Link Functions . . . 38
8.3.1 Canonical Link Functions . . . 39
8.4 Likelihood Optimization . . . 40

Bibliography **41**

In introductory courses, we start with linear predictors since it is easy to understand. Low dimensional linear regression is what statisticians worked in back in the early days, where data was generally low dimensional.[1] Generally, we had $d < n$, but these days, we are in the regime where $d > n$. For example, in genetic data, you could have a sample of $n = 100$ people but each of them have genetic sequences at $d = 10^6$. When the dimensions become high, the original methods of linear regression tend to break down, which is why I separate low and high dimensional linear regression. The line tends to be fuzzy between these two regimes, but we will not worry about strictly defining that now. Let's introduce the most general variant of linear regression.

Definition 0.1 (Linear Regression Model)

A **linear regression model** is a probabilistic model that predicts the conditional distribution of y given x as

$$y = b + w^T x + \epsilon \tag{1}$$

Another common and compact way of writing it is to encode x as a $(d+1)$-dimensional vector where $x_0 = 1$, and write

$$y = \beta^T x + \epsilon, \beta = (b, w) \in \mathbb{R}^{d+1} \tag{2}$$

Get used to both methods. It has the following assumptions.

1. *Linearity in Parameters.* Note that this does not mean linearity in the *covariates.*[a]
2. *Weak exogeneity.* The covariates are observed without error.
3. ϵ is 0-mean.
4. *Homoscedasticity*: ϵ has constant variance.
5. The ϵ's are uncorrelated with each other.
6. *No multicolinearity*: There exists no covariates that are perfectly correlated.

[a] Therefore you could build a regression using non-linear transformations of the covariates, for instance, $y = w_1 x_1 + w_2 x_2^2 + w_3 \log(x_1)$.

Let's go through these assumptions in detail.

1. *Linearity*: This is pretty straightforward, but many beginners assume that we must only fit *lines* with respect to the covariates x_i. This is not true (consider polynomial regression), and all we are assuming is linearity with respect to the *parameters*. If you need to further relax the assumption, you are better off using non-linear modeling.

2. *Weak exogeneity*: The sensitivity of the model can be tested to the assumption of weak exogeneity by doing bootstrap sampling for the covariates and seeing how the sampling affects the parameter estimates. Covariates measured with error used to be a difficult problem to solve, as they required errors-in-variables models, which have very complicated likelihoods. In addition, there is no universal fitting library to deal with these. But nowadays, with the availability of Markov Chain Monte Carlo (MCMC) estimation through probabilistic programming languages, it is a lot easier to deal with these using Bayesian hierarchical models (or multilevel models, or Bayesian graphical models—these have many names).

3. *Constant variance*: the simplest fix is to do a variance-stabilising transformation on the data. Assuming a constant coefficient of variation rather than a constant mean could also work. Some estimation libraries (such as the `glm` package in R) allow specifying the variance as a function of the mean.

4. *Independence of errors*: this is dangerous because in fields like finance, things are usually highly correlated in times of crisis. The most important thing is to understand how risky this assumption is for your setting. If necessary, add a correlation structure to your model, or do a multivariate regression. Both of these require significant resources to estimate parameters, not only in terms of computational power but also in the amount of data required. Another field that looks for correlation between samples is time series analysis, which we will get to in another set of notes.

[1] Quoting Larry Wasserman, even 5 dimensions was considered high and 10 was considered massive.

5. *No Multicollinearity.* Assume that two variables are perfectly correlated. Then, there would be pairs of parameters that are indistinguishable if moved in a certain linear combination. This means that the variance of $\hat{\boldsymbol{\beta}}$ will be very ill conditioned, and you would get a huge standard error in some direction of the β_i's. We can fix this by making sure that the data is not redundant and manually removing them, standardizing the variables, or making a change of basis to remove the correlation. If we just leave the model as is, numerical problems might occur depending on how the fitting algorithms invert the matrices involved. The t-tests that the regression produces can no longer be trusted.[2]

   In order to check multicollinearity, we compute the correlation matrix, and see if there are any off-diagonal entries that are very close to 1, then there is multicollinearity.

Some more terminology: **multiple linear regression** assumes that we have several covariates and one response. If we extend this to multiple responses (i.e. a response vector), this is called **multivariate linear regression**. The simple case for one response is called **simple linear regression**, and we will mention some nice formulas and intuition that come out from working with this.

[2] I suggest reading this Wikipedia article on multicollinearity, as it contains useful information: `https://en.wikipedia.org/wiki/Multicollinearity`. Multicollinearity is a favorite topic of discussion for quant interviewers, and they usually have strong opinions about how it should be handled. The model's intended use will determine how sensitive it is to ignoring the error distribution. In many cases, fitting a line using least-squares estimation is equivalent to assuming errors have a normal distribution. If the real distribution has heavier tails, like the t-distribution, how risky will it make decisions based on your outputs? One way to address this is to use a technique like robust-regression. Another way is to think about the dynamics behind the problem and which distribution would be best suited to model them—as opposed to just fitting a curve through a set of points.

1 Low-Dimensional Ordinary Least Squares

When you learn linear regression for the first time, you are really learning a very specific part of linear regression called ordinary least squares, which is the linear model that comes with a very specific loss function.

Definition 1.1 (Mean Squared Error Loss)

The **MSE loss** is defined

$$L(y, x) = (y - f(x))^2 \tag{3}$$

Definition 1.2 (Ordinary Least Squares Regression)

The **OLS linear regression** model is a linear regression model that tells us to minimize the MSE loss.

Theorem 1.1 (Prediction Risk)

The **prediction risk**[a] of f is

$$R(f) = \mathbb{E}_{x,y}[(y - f(x))^2] = \int (y - f(x))^2 \, dP(x, y) \tag{4}$$

and the empricial risk is

$$\hat{R}(f) = \frac{1}{n} \sum_{i=1}^n (y^{(i)} - f(x^{(i)}))^2 \tag{5}$$

[a] This is a bit different from how Wasserman defines it in his lectures, but I think this is better.

This is a bit weird, since we are just *given* a loss function rather than having derived one from our model. There are two paths that we can take to derive this loss function.

1. The first is to have the extra convenient assumption that the errors ϵ are Gaussian. This is not too unrealistic since combinations of many random noise gives us a Gaussian by the CLT. As often done in machine learning, by computing the likelihood, we can take its negative logarithm to get our loss.

2. The second does *not* assume a distribution on ϵ and rather uses the Gauss-Markov theorem to directly say that the MSE loss minimizes variance among unbiased estimators. This is in a sense more fundamental.

Sometimes, the Gaussian error is given as an assumption, and sometimes it is not. We will go through all these points by first talking about the nice bias-variance decomposition of the MSE loss. Then, we will use the Gauss-Markov theorem justify the MSE loss and introduce the least-squares solution. Finally, we will look at the likelihood approach using Gaussian residuals.

1.1 Bias-Variance Decomposition

It is a well known fact that the true regressor—which may not be linear at all—that minimizes this loss is

$$f^*(x) = \mathbb{E}[Y \mid X = x] \tag{6}$$

which is the conditional expectation of Y given X. This is the true regressor function, which is the best approximation of Y over the σ-algebra generated by X. Therefore, if we consider a function class of linear predictors, we can decompose our risk, which is the distance between our estimated linear regressor and Y, as the sum of the distance between our estimator and the best regressor plus the distance between the best regressor and Y.

Theorem 1.2 (Pythagorean's Theorem)

The expected square loss over the joint measure $\mathbb{P}_{x,y}$ can be decomposed as

$$\mathbb{E}_{x,y}[(y-f(x))^2] = \mathbb{E}_{x,y}\big[\big(y-\mathbb{E}[y \mid x]\big)^2\big] + \mathbb{E}_x\big[\big(\mathbb{E}[y \mid x]-h(x)\big)^2\big] \tag{7}$$

That is, the squared loss decomposes into the squared loss of $\mathbb{E}[y \mid x]$ and $g(x)$, which is the intrinsic misspecification of the model, plus the squared difference of y with its best approximation $\mathbb{E}[y \mid x]$, which is the intrinsic noise inherent in y beyond the σ-algebra of x.

Proof.

We can write

$$\begin{aligned}
\mathbb{E}_{x,y}\big[\big(y-f(x)\big)^2\big] &= \mathbb{E}_{x,y}\big[\big((y-\mathbb{E}[y \mid x]) + (\mathbb{E}[y \mid x]-f(x))\big)^2\big] && (8)\\
&= \mathbb{E}_{x,y}\big[\big(y-\mathbb{E}[y \mid x]\big)^2\big] + \mathbb{E}_{x,y}[\{y-\mathbb{E}[y \mid x]\}\{\mathbb{E}[y \mid x]-f(x)\}] && (9)\\
&\qquad + \mathbb{E}_x\big[\big(\mathbb{E}[y \mid x]-f(x)\big)^2\big] && (10)\\
&= \mathbb{E}_{x,y}\big[\big(y-\mathbb{E}[y \mid x]\big)^2\big] + \mathbb{E}_x\big[\big(\mathbb{E}[y \mid x]-f(x)\big)^2\big] && (11)
\end{aligned}$$

where the middle term cancels out due to the tower property.

Note that since $\mathbb{E}\big[\big(\mathbb{E}[Y \mid X]-g(X)\big)^2\big]$ is the misspecification of the model, we cannot change this (positive) constant, so $\mathbb{E}\big[\big(Y-g(X)\big)^2\big] \geq \mathbb{E}[(Y-\mathbb{E}[Y \mid X])^2]$, with equality achieved when we perfectly fit g as $\mathbb{E}[Y \mid X]$ (i.e. the model is well-specified). Therefore, denoting $\mathcal{F}$ as the set of all $\sigma(X)$-measurable functions, then the minimum of the loss is attained when

$$\operatorname*{argmin}_{g\in\mathcal{F}} \mathbb{E}[L] = \operatorname*{argmin}_{g\in\mathcal{F}} \mathbb{E}\big[\big(Y-g(X)\big)^2\big] = \mathbb{E}[Y \mid X] \tag{12}$$

Essentially, we have decomposed our risk to a part that we can optimize and a part that we cannot, i.e. the intrinsic noise.

Corollary 1.1 (Sufficient to Estimate Conditional Expectation)

Minimizing the prediction risk is equivalent to minimizing the risk of our estimator to the conditional distribution.

$$\operatorname*{argmin}_{f\in\mathcal{F}} R(f) = \operatorname*{argmin}_{f\in\mathcal{F}} \mathbb{E}_{x,y}\left[(\mathbb{E}[y|x]-f(x))^2\right] \tag{13}$$

Even though this example is specific for the mean squared loss, this same decomposition, along with the bias variance decomposition, exists for other losses. It just happens so that the derivations are simple for the MSE, which is why this is introduced first. However, the derivations for other losses are much more messy, and sometimes may not hold rigorously. However, the general intuition that more complex models tend to overfit (higher variance) still hold true.

Let's try to decompose this even more. In frequentist inference, we take a dataset $\mathcal{D}$ and optimize $\hat{f}$ that minimizes this empirical risk. Therefore, for a given $\mathcal{D}$, $\hat{f} = \hat{f}(\mathcal{D})$ is determined, and if $\mathcal{D} = (x^{(i)}, y^{(i)})^n$ is a random variable, then $\hat{f}$ is also a random variable, which we will denote as $\hat{f}_{\mathcal{D}}$ for clarity. It is useful to think of $\mathcal{D}$ as a random variable because by seeing how $\hat{f}_{\mathcal{D}}$ varies as the dataset changes, we can measure the uncertainty in our estimate of $\hat{f}_{\mathcal{D}}$ through $\mathcal{D}$.[3]

[3] If this didn't make sense to you, consider the following thought experiment. Suppose we had a large number of datasets each of size N and each drawn independently from the joint distribution $X \times Y$. For any given dataset $\mathcal{D}$, we can run our learning algorithm and obtain our best fit function $\hat{f}_{\mathcal{D}}$. Different datasets from the ensemble will give different functions and consequently different values of the squared loss.

Lemma 1.1 (Conditional Prediction Risk)

Our conditional prediction risk is

$$r(\mathcal{D}) = \mathbb{E}_{x,y}\left[(\mathbb{E}[y \mid x] - \hat{f}(x))^2 \mid \mathcal{D}\right] \tag{14}$$

If $\mathcal{D}$ is fixed, then this is a real number. If $\mathcal{D}$ is a random variable, then this is a real-valued random variable.

Ideally, we would like two things.

1. *Low Bias.* The average prediction we get over all $\hat{f}_{\mathcal{D}}$ trained on all possible samples of dataset $\mathcal{D}$ should be similar to our best regressor. That is,

$$\mathbb{E}_{\mathcal{D}}\left[\mathbb{E}_{x,y}\left[(\mathbb{E}[y \mid x] - \hat{f}_{\mathcal{D}}(x))^2\right]\right] \tag{15}$$

   should be as low as possible.

2. *Low Variance.* The variance of our conditional prediction risk

$$\mathrm{Var}_{\mathcal{D}}\left[\mathbb{E}_{x,y}\left[(\mathbb{E}[y \mid x] - \hat{f}_{\mathcal{D}}(x))^2\right]\right] \tag{16}$$

   should be as low as possible. That is, we may get very low bias for one dataset $\mathcal{D}$, but if we sampled a different dataset, we should not expect the bias to explode.

Unfortunately, having both low bias *and* low variance is not possible, and we wish to show that now.

Theorem 1.3 (Bias Variance Decomposition Under MSE Loss)

The expected optimal MSE loss decomposes to

$$\mathbb{E}_{\mathcal{D}}\left[(\mathbb{E}[y \mid x] - \hat{f}_{\mathcal{D}}(x))^2\right] = \underbrace{\big(\mathbb{E}[y \mid x] - \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)]\big)^2}_{(\text{bias of } \hat{f}_{\mathcal{D}})^2} + \underbrace{\mathbb{E}_{\mathcal{D}}\big[\big(\mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] - \hat{f}_{\mathcal{D}}(x)\big)^2\big]}_{\text{variance of } \hat{f}_{\mathcal{D}}} \tag{17}$$

Proof.

Consider the term $\big(\mathbb{E}[y \mid x] - \hat{f}_{\mathcal{D}}(x)\big)^2$ above, which models the discrepancy in our optimized hypothesis and the best approximation. We take $\mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)]$[a] So we can split the term into

$$\begin{aligned}
\big(\mathbb{E}[y \mid x] - \hat{f}_{\mathcal{D}}(x)\big)^2 &= \big[\big(\mathbb{E}[y \mid x] - \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)]\big) + \big(\mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] - \hat{f}_{\mathcal{D}}(x)\big)\big]^2 && (18)\\
&= \big(\mathbb{E}[y \mid x] - \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)]\big)^2 + \big(\mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] - \hat{f}_{\mathcal{D}}(x)\big)^2 && (19)\\
&\quad + 2\big(\mathbb{E}[y \mid x] - \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)]\big)\big(\mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] - \hat{f}_{\mathcal{D}}(x)\big) && (20)
\end{aligned}$$

Now take the expectation over $\mathcal{D}$, and for the third term, note that $\big(\mathbb{E}[y \mid x] - \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)]\big)$ is constant with respect to $\mathbb{D}$ anyways, so we can take it out of the expectation. Therefore,

$$\begin{aligned}
\mathbb{E}_{\mathcal{D}}\big[\big(\mathbb{E}[y \mid x] - \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)]\big)\big(\mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] - \hat{f}_{\mathcal{D}}(x)\big)\big] && (21)\\
= \big(\mathbb{E}[y \mid x] - \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)]\big)\,\mathbb{E}_{\mathcal{D}}\big[\mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] - \hat{f}_{\mathcal{D}}(x)\big] && (22)\\
= \big(\mathbb{E}[y \mid x] - \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)]\big) \cdot 0 = 0 && (23)
\end{aligned}$$

[a] Over all datasets $\mathcal{D}$, there will be a function $h_{\boldsymbol{\theta};\mathcal{D}}$, and averaged over all datasets $\mathcal{D}$ is $\mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}]$.

Let's parse these terms a bit more.

1. The bias $\mathbb{E}[y \mid x] - \mathbb{E}_\mathcal{D}[\hat{f}_\mathcal{D}(x)]$ is a random variable of x that measures the difference between the average of our learned predictor $\mathbb{E}_\mathcal{D}[\hat{f}_\mathcal{D}(x)]$ and the true regressor $\mathbb{E}[y \mid x]$.
2. The variance $\mathbb{E}_\mathcal{D}\big[\big(\mathbb{E}_\mathcal{D}[\hat{f}_\mathcal{D}(x)] - \hat{f}_\mathcal{D}(x)\big)^2\big]$ is a random variable of x that measures the variability of our learned functions $\hat{f}_\mathcal{D}$ around our mean $\mathbb{E}_\mathcal{D}[\hat{f}_\mathcal{D}(x)]$.

Therefore, we can substitute this back into our Pythagoras decomposition, where we must now take the expected bias and the expected variance over x to get a form like

$$\text{Expected Loss} = (\text{Expected Bias})^2 + \text{Expected Variance} + \text{Noise} \tag{24}$$

Corollary 1.2 (Bias Variance Decomposition of Expected MSE Loss)

The expected optimal MSE loss decomposes to

$$\mathbb{E}_\mathcal{D}\mathbb{E}_{x,y}\big[(y - \hat{f}_\mathcal{D}(x))^2\big] = \mathbb{E}_x\Big[\underbrace{\big(\mathbb{E}[y \mid x] - \mathbb{E}_\mathcal{D}[\hat{f}_\mathcal{D}(x)]\big)^2}_{(\text{expected bias})^2}\Big] + \underbrace{\mathbb{E}_\mathcal{D}\Big[\mathbb{E}_x\Big[\big(\mathbb{E}_\mathcal{D}[\hat{f}_\mathcal{D}(x)] - \hat{f}_\mathcal{D}(x)\big)^2\Big]\Big]}_{\text{expected variance}} \tag{25}$$

$$+ \underbrace{\mathbb{E}_{x,y}[(y - \mathbb{E}[y \mid x])^2]}_{\text{noise}} \tag{26}$$

Proof.

By taking the expectation over x and swapping the expectations (since x and $\mathcal{D}$ are independent), and finally substituting back to Pythagoras decomposition, we get the following.

1.2 Least Squares Solution

Note that we have assumed that $\mathbf{X}^T\mathbf{X}$ was invertible in order for such a solution to be unique, i.e. $\mathbf{X}$ must be full rank. This process breaks down when it isn't invertible, e.g. if there are repetitions in the features (one feature is a linear combination of the others and hence not full column rank). We will talk more about this soon.

Definition 1.3 (Hat Matrix)

For convenience of notation, let's call

$$\mathbf{H} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T \tag{27}$$

the $n \times n$ **hat matrix**, which is essentially a projection of the observed y_i's to the predictions.

$$\hat{\mathbf{y}} = \mathbf{H}\mathbf{y} \tag{28}$$

Lemma 1.2 (Properties)

The hat matrix is an orthogonal projection matrix that projects to the column space of $\mathbf{X}$.

Note that this parallels the orthogonal projection of conditional expectation to the true function onto the subspace of X measurable functions. Except that we are not doing this in function space, but rather the sample space $\mathbb{R}^n$.

We can also see that the residuals $\hat{\epsilon}_i = y_i - \hat{y}_i$ has the property that

$$\hat{\boldsymbol{\epsilon}} = \mathbf{y} - \hat{\mathbf{y}} = (\mathbf{I}_n - \mathbf{H})\mathbf{y} \tag{29}$$

Now if we look back to the derivative of the loss S, we really want to set

$$\mathbf{X}^T(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}) = \mathbf{X}^T\hat{\boldsymbol{\epsilon}} = \mathbf{0} \tag{30}$$

Theorem 1.4 (Least Squares Solution For Linear Regression)

Given the design matrix $\mathbf{X}$, we can present the linear model in vectorized form:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon},\ \boldsymbol{\epsilon} \sim N(\mathbf{0}, \sigma^2\mathbf{I}) \tag{31}$$

The solution that minimizes the squared loss is

$$\begin{aligned} \boldsymbol{\beta} &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} \in \mathbb{R}^d \\ \mathrm{Var}(\hat{\boldsymbol{\beta}}) &= \hat{\sigma}^2(\mathbf{X}^T\mathbf{X})^{-1} \in \mathbb{R}^{d\times d} \end{aligned}$$

Proof.

The errors can be written as $\boldsymbol{\epsilon} = \mathbf{Y} - \mathbf{X}\boldsymbol{\beta}$, and you have the following total sum of squared errors:

$$S(\boldsymbol{\beta}) = \boldsymbol{\epsilon}^T\boldsymbol{\epsilon} = (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})$$

We want to find the value of $\boldsymbol{\beta}$ that minimizes the sum of squared errors. In order to do this, remember the following matrix derivative rules when differentiating with respect to vector $\mathbf{x}$.

1. $\mathbf{x}^T\mathbf{A} \mapsto \mathbf{A}$
2. $\mathbf{x}^T\mathbf{A}\mathbf{x} \mapsto 2\mathbf{A}\mathbf{x}$

Now this should be easy.

$$\begin{aligned} S(\boldsymbol{\beta}) &= \mathbf{Y}^T\mathbf{Y} - \boldsymbol{\beta}^T\mathbf{X}^T\mathbf{Y} - \mathbf{Y}^T\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\beta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} \\ &= \mathbf{Y}^T\mathbf{Y} - 2\mathbf{Y}^T\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\beta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} \\ \frac{\partial}{\partial\boldsymbol{\beta}}S(\boldsymbol{\beta}) &= -2\mathbf{X}^T\mathbf{Y} + 2\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} \end{aligned}$$

and setting it to $\mathbf{0}$ gives

$$2\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} - 2\mathbf{X}^T\mathbf{Y} = 0 \implies \mathbf{X}^T\mathbf{X}\boldsymbol{\beta} = \mathbf{X}^T\mathbf{Y}$$

and the variance of $\boldsymbol{\beta}$, by using the fact that $\mathrm{Var}[\mathbf{A}\mathbf{X}] = \mathbf{A}\,\mathrm{Var}[X]\mathbf{A}^T$, is

$$\mathrm{Var}(\hat{\boldsymbol{\beta}}) = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\ \sigma^2\mathbf{I}\ \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1} = \sigma^2(\mathbf{X}'\mathbf{X})^{-1}(\mathbf{X}'\mathbf{X})(\mathbf{X}'\mathbf{X})^{-1} = \sigma^2(\mathbf{X}'\mathbf{X})^{-1}$$

But we don't know the true σ^2, so we estimate it with $\hat{\sigma}^2$ by taking the variance of the residuals. Therefore, we have

$$\begin{aligned} \boldsymbol{\beta} &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} \in \mathbb{R}^d \\ \mathrm{Var}(\hat{\boldsymbol{\beta}}) &= \hat{\sigma}^2(\mathbf{X}^T\mathbf{X})^{-1} \in \mathbb{R}^{d\times d} \end{aligned}$$

Example 1.1 (Copying Data)

What happens if you copy your data in OLS? In this case, our MLE estimate becomes

$$\begin{aligned} &\left(\begin{pmatrix} X \\ X \end{pmatrix}^T \begin{pmatrix} X \\ X \end{pmatrix}\right)^{-1} \begin{pmatrix} X \\ X \end{pmatrix}^T \begin{pmatrix} Y \\ Y \end{pmatrix} = \\ &\qquad = (X^TX + X^TX)^{-1}(X^TY + X^TY) = (2X^TX)^{-1}2X^TY = \hat{\beta} \end{aligned}$$

and our estimate is unaffected. However, the variance shrinks by a factor of 2 to

$$\frac{\sigma^2}{2}(\mathbf{X}^T\mathbf{X})^{-1} \tag{32}$$

A consequence of that is that confidence intervals will shrink with a factor of $1/\sqrt{2}$. The reason is that we have calculated as if we still had iid data, which is untrue. The pair of doubled values are obviously dependent and have a correlation of 1.

Theorem 1.5 (Gauss-Markov Theorem)

Given a dataset with

1. mean zero residuals $\mathbb{E}[\epsilon_i] = 0$, i.e. $\mathbb{E}[\mathbf{Y} \mid \mathbf{X}] = \mathbf{X}\boldsymbol{\beta}$.
2. homoscedacity $\text{Var}[\epsilon_i] = \sigma^2 < \infty$ for all i,
3. uncorrelated residuals $\text{Cov}(\epsilon_i, \epsilon_j) = 0$ for all $i \neq j$. This and the previous assumption can be combined into $\text{Cov}[\mathbf{Y} \mid \mathbf{X}] = \sigma^2 \mathbf{I}_n$.

We were concerned with estimating the parameters $\beta_1, \ldots, \beta_d$. Now let's generalize this and consider the problem of estimating, for some known constants $c_1, \ldots, c_{d+1}$, the point estimator

$$\theta = c_1\beta_1 + c_2\beta_2 + \ldots + c_d\beta_d + c_{d+1} \tag{33}$$

Then the estimator

$$\hat{\theta} = c_1\hat{\beta}_1 + c_2\hat{\beta}_2 + \ldots + c_d\hat{\beta}_d + c_{d+1} \tag{34}$$

where $\hat{\beta}_i$ is clearly an unbiased estimator of θ and it is a linear estimator of θ, i.e.

$$\hat{\theta} = \sum_{i=1}^n b_i y_i \tag{35}$$

for some known (given $\mathbf{X}$) constants b_i. Then, the Gauss-Markov theorem states that the estimator $\hat{\theta}$ has the smallest (best) variance among *all* linear unbiased estimators of θ, i.e. $\hat{\theta}$ is BLUE.

1.3 Likelihood Estimation

Now given these assumptions, what is the likelihood of the data?

Lemma 1.3 (Likelihood)

Given a dataset $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^N$, our likelihood is

$$L(\theta; \mathcal{D}) = \prod_{i=1}^N p(y^{(i)} \mid x^{(i)}; \theta) = \prod_{i=1}^N \frac{1}{\sqrt{2\pi\sigma^2}} \exp\bigg(- \frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2} \bigg)$$

By taking the negative log, we can get our loss.

Definition 1.4 (Mean Squared Error Loss)

We can take its negative log, remove additive constants, and scale accordingly to get

$$\begin{aligned} \ell(\theta) &= -\frac{N}{2}\ln\sigma^2 - \frac{N}{2}\ln(2\pi) + \frac{1}{2\sigma^2}\sum_{i=1}^N \big(y^{(i)} - \boldsymbol{\theta}^T\mathbf{x}^{(i)}\big)^2 \qquad (36) \\ &= \frac{1}{2}\sum_{i=1}^N \big(y^{(i)} - \boldsymbol{\theta}^T\mathbf{x}^{(i)}\big)^2 \qquad (37) \end{aligned}$$

which then corresponds to minimizing the sum of squares error function.

Theorem 1.6 (Gradient Descent for Linear Regression)

Taking the gradient of this log likelihood w.r.t. θ gives

$$\nabla_\theta \ell(\theta) = \sum_{i=1}^N (y^{(i)} - \theta^T x^{(i)}) x^{(i)}$$

and running gradient descent over a minibatch $M \subset \mathcal{D}$ gives

$$\begin{aligned} \theta &= \theta - \eta \nabla_\theta \ell(\theta) \\ &= \theta - \eta \sum_{(x,y)\in M} (y - \theta^T x)x \end{aligned}$$

This is guaranteed to converge since $\ell(\theta)$, as the sum of convex functions, is also convex.
Note that since we can solve this in closed form, by setting the gradient to 0, we have

$$0 = \sum_{n=1}^N y^{(n)} \boldsymbol{\phi}(\mathbf{x}^{(n)})^T - \mathbf{w}^T \left(\sum_{n=1}^N \boldsymbol{\phi}(\mathbf{x}^{(n)}) \boldsymbol{\phi}(\mathbf{x}^{(n)})^T \right)$$

which is equivalent to solving the least squares equation

$$\mathbf{w}_{ML} = (\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathbf{Y}$$

Note that if we write out the bias term out explicitly, we can see that it just accounts for the translation (difference) between the average of the outputs $\bar{y} = \frac{1}{N}\sum_{n=1}^N y_n$ and the average of the basis functions $\bar{\phi_j} = \frac{1}{N}\sum_{n=1}^N \phi_j(\mathbf{x}^{(n)})$.

$$w_0 = \bar{y} - \sum_{j=1}^{M-1} w_j \bar{\phi_j}$$

We can also maximize the log likelihood w.r.t. σ^2, which gives the MLE

$$\sigma^2_{ML} = \frac{1}{N}\sum_{n=1}^N \big(y^{(n)} - \mathbf{w}_{ML}^T \boldsymbol{\phi}(\mathbf{x}^{(n)})\big)^2$$

Code 1.1 (MWE for OLS Linear Regression in scikit-learn)

Here is a minimal working example of performing linear regression with scikit-learn. Note that the input data must be of shape (n, d).

```python
import numpy as np
from sklearn.linear_model import LinearRegression

X = np.array([[1, 1], [1, 2], [2, 2], [2, 3]])
y = np.dot(X, np.array([1, 2])) + 3

model = LinearRegression()
model.fit(X, y)
print(X)
print(y)
print(model.score(X, y))
print(model.intercept_)
print(model.coef_)
print(model.predict(np.array([[3, 5]])))
```

```
[[1 1]
 [1 2]
 [2 2]
 [2 3]]
[ 6  8  9 11]
1.0
3.0000000000000018
[1. 2.]
[16.]
.
.
.
.
.
```

1.4 Simple Linear Regression

The simple linear regression is the special case of the linear regression with only one covariate.[4]

$$y = \alpha + x\beta \tag{38}$$

which is just a straight line fit. Interviewers like this model for its aesthetically pleasing theoretical properties. A few of them are described here, beginning with parameter estimation. For n pairs of (x_i, y_i),

$$y_i = \alpha + \beta x_i + \epsilon_i \tag{39}$$

To minimize the sum of squared errors

$$\sum_i \epsilon_i^2 = \sum_i (y_i - \alpha - \beta x_i)^2 \tag{40}$$

Taking the partial derivatives w.r.t. α and β and setting them equal to 0 gives

$$\sum_i (y_i - \hat{\alpha} - \hat{\beta} x_i) = 0$$
$$\sum_i (y_i - \hat{\alpha} - \hat{\beta} x_i) x_i = 0$$

From just the first equation, we can write

$$n\bar{y} = n\hat{\alpha} + n\hat{\beta}\bar{x} \implies y = \hat{\alpha} + \hat{\beta}\bar{x} \implies \hat{\alpha} = \bar{y} - \hat{\beta}\bar{x} \tag{41}$$

The second equation gives

$$\sum_i x_i y_i = \hat{\alpha} n \bar{x} + \hat{\beta} \sum_i x_i^2 \tag{42}$$

and substituting what we derived gives

$$\begin{aligned} \sum_i x_i y_i &= (\bar{y} - \hat{\beta}\bar{x}) n \bar{x} + \hat{\beta} \sum_i x_i^2 \\ &= n\bar{x}\bar{y} + \hat{\beta} \bigg(\Big(\sum_i x_i^2 \Big) - n\bar{x}^2 \bigg) \end{aligned}$$

[4] I've included a separate section on this since this was especially important for quant interviews.

and so we have

$$\hat{\beta} = \frac{\big(\sum_i x_i y_i\big) - n\bar{x}\bar{y}}{\big(\sum x_i^2\big) - n\bar{x}^2} = \frac{\sum_i x_i y_i - \bar{x} y_i}{\sum x_i^2 - \bar{x} x_i} = \frac{\sum_i (x_i - \bar{x}) y_i}{\sum_i (x_i - \bar{x}) x_i} \tag{43}$$

Now we can use the identity

$$\sum_i (x_i - \bar{x})(y_i - \bar{y}) = \sum_i y_i (x_i - \bar{x}) = \sum_i x_i (y_i - \bar{y})$$

to substitute both the numerator and denominator of the equation to

$$\hat{\beta} = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2} = \frac{\text{cov}(x, y)}{\text{var}(x)} = \rho_{xy} \frac{s_y}{s_x}$$

where ρ_{xy} is the correlation between x and y, and the variance and covariance represent the sample variance and covariance (indicated in lower case letters). Therefore, the correlation coefficient ρ_{xy} is precisely equal to the slope of the best fit line when x and y have been standardized first, i.e. $s_x = s_y = 1$.

Example 1.2 (Switching Variables)

Say that we are fitting Y onto X in a simple regression setting with MLE β_1, and now we wish to fit X onto Y. How will the MLE slope change? We can see that

$$\beta_1 = \rho \frac{s_y}{s_x}, \quad \beta_2 = \rho \frac{s_x}{s_y}$$

and so

$$\beta_2 = \rho^2 \frac{1}{\rho} \frac{s_x}{s_y} = \rho^2 \frac{1}{\beta_1} = \beta_1 \frac{\text{var}(x)}{\text{var}(y)}$$

The reason for this is because regression lines don't necessarily correspond to one-to-one to a casual relationship. Rather, they relate more directly to a conditional probability or best prediction.

The **coefficient of determination** R^2 is a measure tells you how well your line fits the data. When you have your y_i's, their deviation around its mean is captured by the sample variance $s_y^2 = \sum_i (y_i - \bar{y})^2$. When we fit our line, we want the deviation of y_i around our predicted values $\hat{y}_i$, i.e. our sum of squared loss $\sum_i (y_i - \hat{y}_i)^2$, to be lower. Therefore, we can define

$$R^2 = 1 - \frac{\text{MSELoss}}{\text{var}(y)} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

In simple linear regression, we have

$$R^2 = \rho_{yx}^2$$

An R^2 of 0 means that the model does not improve prediction over the mean model and 1 indicates perfect prediction. However, a drawback of R^2 is that it can increase if we add predictors to the regression model, leading to a possible overfitting.

Theorem 1.7 ()

The residual sum of squares (RSS) is equal to the a proportion of the variance of the y_i's.

$$\text{RSS} = \sum (y_i - \hat{y}_i)^2 = (1 - \rho^2) \sum (y_i - \bar{y})^2 \tag{44}$$

1.5 Concentration Bounds

Let's get a deeper understanding on linear regression by examining the convergence of the empirical risk minimizer to the true risk minimizer. We can develop a naive bound using basic concentration of measure.

Theorem 1.8 (Exponential Bound)

Let $\mathcal{P}$ be the set of all distributions for $X \times Y$ supported on a compact set. There exists constants c_1, c_2 s.t. that the following is true. For any $\epsilon > 0$,

$$\sup_{\mathbb{P} \in \mathcal{P}} \mathbb{P}^n \big(r(\hat{\beta}_n) > r(\beta_*(\mathbb{P}) + 2\epsilon) \big) \leq c_1 e^{-nc_2 \epsilon^2} \tag{45}$$

Hence

$$r(\hat{\beta}_n) - r(\beta_*) = O_{\mathbb{P}} \left(\sqrt{\frac{1}{n}} \right) \tag{46}$$

Proof.

Given any β, define $\tilde{\beta} = (-1, \beta)$ and $\Lambda = \mathbb{E}[ZZ^T]$ where $Z = (Y, X)$. Note that

$$r(\beta) = \mathbb{E}(Y - \beta^T X)^2 = \mathbb{E}[(Z^T \tilde{\beta})^2] = \tilde{\beta}^T \Lambda \tilde{\beta}. \tag{47}$$

Similarly,

$$\hat{r}_n(\beta) = \tilde{\beta}^T \hat{\Lambda}_n \tilde{\beta} \tag{48}$$

where

$$\hat{\Lambda}_n = \frac{1}{n} \sum_{i=1}^n Z_i Z_i^T. \tag{49}$$

So

$$|\hat{r}_n(\beta) - r(\beta)| = |\tilde{\beta}^T (\hat{\Lambda}_n - \Lambda) \tilde{\beta}| \leq \|\tilde{\beta}\|_1^2 \Delta_n \tag{50}$$

where

$$\Delta_n = \max_{j,k} |\hat{\Lambda}_n(j,k) - \Lambda(j,k)|. \tag{51}$$

By Hoeffding's inequality and the union bound (applied to each entry of the matrix $\hat{\Lambda}_n - \Lambda$),

$$\mathbb{P} \left(\sup_\beta |\hat{r}_n(\beta) - r(\beta)| > \epsilon \right) \leq c_1 e^{-nc_2 \epsilon^2}. \tag{52}$$

On the event $\sup_\beta |\hat{r}_n(\beta) - r(\beta)| < \epsilon$, we have

$$r(\beta_*) \leq r(\hat{\beta}_n) \leq \hat{r}_n(\hat{\beta}_n) + \epsilon \leq \hat{r}_n(\beta_*) + \epsilon \leq r(\beta_*) + 2\epsilon. \tag{53}$$

The second inequality uses the fact that $|\hat{r}_n(\hat{\beta}_n) - r(\hat{\beta}_n)| < \epsilon$, the third uses the definition of $\hat{\beta}_n$ as the minimizer of $\hat{r}_n$, and the fourth uses $|\hat{r}_n(\beta_*) - r(\beta_*)| < \epsilon$.
Therefore,

$$\mathbb{P}^n(r(\hat{\beta}_n) > r(\beta_*(\mathbb{P})) + 2\epsilon) \leq \mathbb{P} \left(\sup_\beta |\hat{r}_n(\beta) - r(\beta)| \geq \epsilon \right) \leq c_1 e^{-nc_2 \epsilon^2}. \tag{54}$$

Taking the supremum over $\mathbb{P} \in \mathcal{P}$ gives the first result.
For the second result, the exponential bound implies that for any $\delta > 0$,

$$\mathbb{P}(r(\hat{\beta}_n) - r(\beta_*) > t) \leq c_1 e^{-nc_2 t^2/4} \tag{55}$$

for $t > 0$. Setting this equal to δ and solving for t gives $t = O(\sqrt{\log(1/\delta)/n})$. Since this holds for all $\delta > 0$, we have $r(\hat{\beta}_n) - r(\beta_*) = O_{\mathbb{P}}(\sqrt{1/n})$.

However, this is not a very tight bound, and we can do better. The next theorem reveals to us that in linear regression, the bounds are of order $\frac{d}{n}$, and so scales linearly with dimension d.

Theorem 1.9 (Gyorfi, Kohler, Krzyzak, Walk, 2002 [GKKW02])

Let $\sigma^2 = \sup_x \mathrm{Var}[Y \mid X = x] < \infty$. Assume that all random variables are bounded by $L < \infty$. Then

$$\mathbb{E} \int |\hat{\beta}^T x - m(x)|^2 \, d\mathbb{P}(x) \leq 8 \inf_{\beta} \int |\beta^T x - m(x)|^2 \, d\mathbb{P}(x) + \frac{Cd(\log(n) + 1)}{n} \tag{56}$$

Proof.

Straightforward but long. Omitted.

You can see that the bound contains a term of the form

$$\frac{d \log(n)}{n} \tag{57}$$

and under the low dimensional case, d is small and bound is good. However, as d becomes large, then we don't have as good of theoretical guarantees.

Theorem 1.10 (Central Limit Theorem of OLS)

We have

$$\sqrt{n}(\hat{\beta} - \beta) \xrightarrow{d} N(0, \Gamma) \tag{58}$$

where

$$\Gamma = \Sigma^{-1} \mathbb{E}\big[(Y - X^T \beta)^2 X X^T\big] \Sigma^{-1} \tag{59}$$

The covariance matrix Γ can be consistently estimated by

$$\hat{\Gamma} = \hat{\Sigma}^{-1} \hat{M} \hat{\Sigma}^{-1} \tag{60}$$

where

$$\hat{M}(j, k) = \frac{1}{n} \sum_{i=1}^n X_i(j) X_i(k) \hat{\epsilon}_i^2 \tag{61}$$

and $\hat{\epsilon}_i = Y_i - \hat{\beta}^T X_i$.

2 Significance Tests and Confidence Sets

This is not as emphasized in the machine learning literature, but it is useful to know from a statistical point of view.[5]

2.1 T Test

Given some multilinear regression problem where we must estimate $\boldsymbol{\beta} \in \mathbb{R}^{D+1}$ (D coefficients and 1 bias), we must determine whether there is actually a linear relationship between the x and y variables in our dataset $\mathcal{D}$. Say that we have a sample of N points $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^N$. Then, for each ensemble of datasets $\mathcal{D}$ that we sample from the distribution $(X \times Y)^N$, we will have some estimator $\boldsymbol{\beta}$ for each of them. This will create a sampling distribution of $\boldsymbol{\beta}$'s where we can construct our significance test on.

So what should our sampling distribution of $\hat{\boldsymbol{\beta}}$ be? It is clearly normal since it is just a transformation of the normally distributed Y: $\hat{\boldsymbol{\beta}} \sim N(\boldsymbol{\beta}, \sigma^2 (X^T X)^{-1})$. Therefore, only considering one element β_i here,

$$\frac{\hat{\beta}_i - \beta_i}{\sigma \sqrt{(X^T X)^{-1}_{ii}}} \sim N(0,1)$$

But the problem is that we don't know the true σ^2, and we are estimating it with $\hat{\sigma}^2$. If we knew the true σ^2 then this would be a normal, but because of this estimate, our normalizing factor is also random. It turns out that the residual sum of squares (RSS) for a multiple linear regression

$$\sum_i (y_i - x_i^T \beta)^2$$

follows a χ^2_{n-d} distribution. Additionally from the χ^2 distribution of RSS we have

$$\frac{(n-d)\hat{\sigma}^2}{\sigma^2} \sim \chi^2_{n-d}$$

where we define $\hat{\sigma}^2 = \frac{\text{RSS}}{n-d}$ which is an unbiased estimator for σ^2. Now there is a theorem that says that if you divide a $N(0,1)$ distribution by a χ^2_k/k distribution (with k degrees of freedom), then it gives you a t-distribution with the same degrees of freedom. Therefore, we divide

$$\frac{\frac{\hat{\beta}_i - \beta_i}{\sqrt{(X^T X)^{-1}_{ii}}}}{\hat{\sigma}} = \frac{\sigma \sim N(0,1)}{\sigma \chi^2_{n-d}/(n-d)} = \frac{\sim N(0,1)}{\chi^2_{n-d}/(n-d)} = t_{n-d}$$

where the standard error of the distribution is

$$\text{SE}(\hat{\beta}_i) = \sigma_{\hat{\beta}_i} = \sigma \sqrt{(X^T X)^{-1}_{ii}}$$

In ordinary linear regression, we have the null hypothesis $h_0 : \beta_i = 0$ and the alternative $h_a : \beta_i \neq 0$ for a two sided test or $h_a : \beta_i > 0$ for a one sided test. Given a certain significance level, we compute the critical values of the t-distribution at that level and compare it with the test statistic

$$t = \frac{\hat{\beta} - 0}{\text{SE}(\hat{\beta})}$$

Now given our β, how do we find the standard error of it? Well this is just the variance of our estimator $\boldsymbol{\beta}$, which is $\hat{\sigma}^2 (\mathbf{X}^T \mathbf{X})^{-1}$, where $\hat{\sigma}^2$ is estimated by taking the variance of the residuals ϵ_i. When there is a single variable, the model reduces to

$$y = \beta_0 + \beta_1 x + \epsilon$$

[5] This is also asked in quant interviews.

and

$$\mathbf{X} = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix}, \qquad \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}$$

and so

$$(\mathbf{X}'\mathbf{X})^{-1} = \frac{1}{n\sum x_i^2 - (\sum x_i)^2} \begin{pmatrix} \sum x_i^2 & -\sum x_i \\ -\sum x_i & n \end{pmatrix}$$

and substituting this in gives

$$\sqrt{\widehat{\text{Var}}(\hat{\beta}_1)} = \sqrt{[\hat{\sigma}^2(\mathbf{X}'\mathbf{X})^{-1}]_{22}} = \sqrt{\frac{\hat{\sigma}^2}{\sum x_i^2 - (\sum x_i)^2}} = \sqrt{\frac{\hat{\sigma}^2}{\sum (x_i - \bar{x}_i)^2}}$$

Example 2.1 ()

Given a dataset

```
  Hours Studied for Exam 20 16 20 18 17 16 15 17 15 16 15 17 16 17 14
  Grade on Exam 89 72 93 84 81 75 70 82 69 83 80 83 81 84 76
```

The hypotheses are $h_0 : \beta = 0$ and $h_a : \beta \neq 0$, and the degrees of freedom for the t-test is $df = N - (D+1) = 13$, where $N = 15$ is the number of datapoints and $D = 1$ is the number of coefficients (plus the 1 bias term). The critical values is ± 2.160, which can be found by taking the inverse CDF of the t-distribution evaluated at 0.975.
Now we calculate the t score. We have our estimate $\beta_1 = 3.216, \beta_0 = 26.742$, and so we calculate

$$\hat{\sigma}^2 = \frac{1}{15} \sum_{i=1}^{15} \big(y_i - (3.216 x_i + 26.742)\big) = 13.426$$

$$\sum_i (x_i - \hat{x}_i)^2 = 41.6$$

and therefore, we can compute

$$t = \frac{\beta_1}{\sqrt{\hat{\sigma}^2 / \sum_i (x_i - \hat{x}_i)^2}} = \frac{3.216}{\sqrt{13.426/41.6}} = 5.661$$

and therefore, this is way further than our critical value of 2.16, meaning that we reject the null hypothesis.

Note that when multicolinearity is present, then $\sum_i (x_i - \hat{x}_i)^2$ will be very small causing the denominator to blow up, and therefore you cannot place too much emphasis on the interpretation of these statistics. While it is hard to see for the single linear regression case, we know that some eigenvalue of $(\mathbf{X}^T\mathbf{X})^{-1}$ will blow up, causing the diagonal entries $(\mathbf{X}^T\mathbf{X})^{-1}_{ii}$ to be very small. When we calculate the standard error by dividing by this small value, the error blows up.

Theorem 2.1 ()

We can compute this t-statistic w.r.t. just the sample size n and the correlation coefficient ρ as such.

$$t = \frac{\hat{\beta} - 0}{\text{SE}(\hat{\beta})}$$

and the denominator is simply

$$\text{SE}(\hat{\beta}) = \sqrt{\frac{\frac{1}{n-1}\sum(y_i - \hat{y})^2}{\sum(x_i - \bar{x})^2}} \implies t = \frac{\hat{\beta}\sqrt{\sum(x_i - \bar{x})^2}\sqrt{n-1}}{\sqrt{\sum(y_i - \hat{y})^2}} = \frac{\hat{\beta}\sqrt{\sum(x_i - \bar{x})^2}\sqrt{n-1}}{\sqrt{(1-\rho^2)}\sqrt{\sum(y_i - \bar{y})^2}}$$
$$= \frac{\rho}{\sqrt{1-\rho^2}}\sqrt{n-1}$$

where the residual sum of squares on the top can be substituted according to our theorem. Therefore

$$t = \frac{\rho}{\sqrt{1-\rho^2}}\sqrt{n-1} \tag{62}$$

2.2 F Test

Given that you have n data points that have been fit on a linear model, the F-statistic is based on the ratio of two variances.

3 Ridge Regression

Ridge regression is used both in the high dimensional case or when our function space is too large/complex, which leads to overfitting. In the overfitting case, we have seen that either decreasing our function space or getting more training data helps. Another popular way is to add a *regularizing term* to the loss function in order to discourage the coefficients from reaching large values, effectively limiting the variance over $\mathcal{D}$. These are called *shrinkage models*, which "shrinks" the parameters to 0.

Definition 3.1 (Ridge Regression)

Ridge regression[a] refers to a linear model minimized with the *ridge loss*.

$$L(f, x, y) = (y - f(x))^2 + \lambda \|\beta\|^2 \tag{63}$$

where we penalize according to the L^2 norm of the coefficients.

[a] Also called weight decay in machine learning or Tikhonov regularization in signal processing.

Therefore, this regularization term effectively controls the variance that our estimator could have, which inevitably trades off with the bias. Therefore, λ acts as sort of a tuning knob between bias and variance. Think of the extreme cases when $\lambda \to \infty$. Then, all weights would be 0, and we would have extreme bias but no variance. On the other hand if $\lambda = 0$, then we are back to OLS.

Lemma 3.1 (Risk)

The prediction risk is of f is

$$R(f) = \mathbb{E}_{x,y} \left[(y - f(x))^2 + \|\beta\|^2 \right] = \mathbb{E}_{x,y} \left[(y - \beta^T x)^2 + \|\beta\|^2 \right] \tag{64}$$

and the empirical prediction risk is

$$\hat{R}(f) = \frac{1}{n} \left(\sum_{i=1}^n (y^{(i)} - f(x^{(i)}))^2 \right) + \lambda \|\beta\|^2 \tag{65}$$

Again, we should question why we should choose *this* form of the risk? Sure we should find some function that shrinks x to 0, but why the L^2 norm? One reason is that it is convenient and has a lot of nice properties as we will see later. Another is that later, in the Bayesian interpretation, this is equivalent to having a Gaussian prior on the parameter space. Other than these two reasons, I still have not yet found a good derivation, e.g. the analogue of the Gauss-Markov theorem or even some distributional assumptions that lead to this loss.

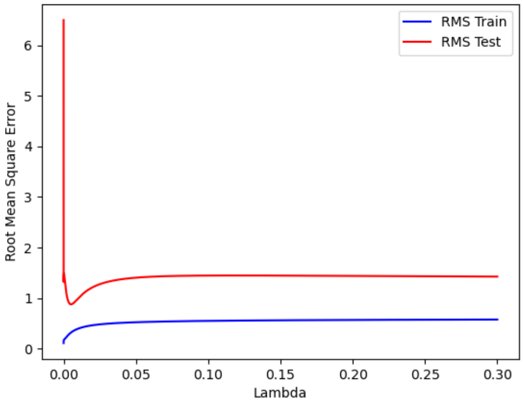

Figure 1: Even with a slight increase in the regularization term λ, the RMS error on the testing set heavily decreases.

3.1 Least Squares Solution

Now that we have this form, we might as well just solve it.

Theorem 3.1 (Least Squares Solution for Ridge Regression)

The minimizer of the ridge loss is

$$\hat{\beta} = (X^T X + \lambda I)^{-1} X^T Y \tag{66}$$

Proof.

TBD

Code 3.1 (MWS of Ridge Regression in scikit-learn)

```python
import numpy as np
from sklearn.linear_model import Ridge

X = np.random.randn(10, 5)
y = np.random.randn(10)
# regularization parameter
model = Ridge(alpha=1.0)
model.fit(X, y)
print(model.score(X, y))
print(model.intercept_)
print(model.coef_)
print(model.predict(np.array([[1, 2, 3, 4, 5]])))
```

```
0.8605535024325397
-0.28291076492665157
[-0.10400521 -0.7587073
    -0.05116735  1.16236649
    -0.0401323 ]
[2.39097184]
.
.
.
.
.
.
```

3.2 Bias Variance Tradeoff

Theorem 3.2 (Bias Variance Decomposition of Ridge Regression)

TBD

From a computational point of view, we can see that by adding the λI term, it *dampens* the matrix so that it does become invertible (or well conditioned), allowing us to find a solution. The higher the λ term, the higher the damping effect.

3.3 Concentration Bounds

The next theorem compares the performance of the best ridge regression estimator to the best linear predictor.

Theorem 3.3 (Hsu, Kakade, Zhang, 2014 [HKZ14])

Suppose that $||X_i|| \leq r$ and let $\beta^T x$ be the best linear approximation to $m(x)$. Then, with probability at least $1 - 4e^{-t}$, we have

$$r(\hat{\beta}) - r(\beta) \leq \bigg(1 + O\bigg(\frac{1 + r^2/\lambda}{n}\bigg)\bigg) \frac{\lambda ||\beta||^2}{2} + \frac{\sigma^2}{n} \frac{\text{Tr}(\Sigma)}{2\lambda} \tag{67}$$

We can see that the λ term exists in the numerator on $\frac{\lambda||\beta||^2}{2}$ and in the denominator on $\frac{\text{Tr}(\Sigma)}{2\lambda}$. This is the bias variance tradeoff. The first term is the bias term, which is the penalty for not being able to fit the data as well. The second term is the variance term, which is the penalty for having a more complex model. So our optimal λ in the theoretical sense would be the one that minimizes the sum of these two terms. In practice, it's not this clean since we have unknown quantities in the formula, but just like how we did cross validation over the model complexity, we can also do cross validation over the λ. The decomposition above just gives you a theoretical feeling of how these things trade off.

3.4 Tuning the Regularization Coefficient

4 Stepwise Regression

Now we move to *sparse* linear regression.

Now suppose that $d > n$, then the first problem is that we can no longer use least squares since $X^T X$ is no longer invertible and the same problem happens with maximum likelihood. This is known as the **high dimensional** or **large p, small** n problem. The most straightforward way is simply to reduce the covariates to a dimension smaller than n. This can be done with three ways.

1. We perform PCA on the X and use the first k principal components where $k < n$.
2. We cluster the covariates based on their correlation. We can use one feature from each cluster or take the average of the covariates within each cluster.
3. We can screen the variables by choosing the k features that have the largest correlation with Y.

Once this is done, we are back in the low dimensional regime and can use least squares. Essentially, this is a way to find a good subset of the covariates, which can be formalized by the following. Let S be a subset of $[d]$ and let $X_S = (X_j : j \in S)$. If the size of S is not too large, we can regress Y on X_S instead of X.

4.1 Best Subset Regression

Definition 4.1 (Best Subset Regression)

Fix $k < d$ and let $\mathcal{S}_k$ denote all subsets of size k. For a given $S \in \mathcal{S}_k$, let β_S be the best linear predictor for the subset S. **Best subset regression** is a linear regression model that wants to solve the best subset S that minimizes the loss

$$\mathbb{E}[(Y - \beta_S^T X_S)^2] \tag{68}$$

which is equivalent to finding

$$\operatorname*{argmin}_{\beta} \mathbb{E}[(Y - \beta^T X)^2] \text{ subject to } ||\beta||_0 \leq k \tag{69}$$

where $||\beta||_0$ is the number of non-zero entries in β.

There will be a bias variance tradeoff. As k increases, the bias decreases but the variance increases.

The minimization of the empirical error is over all subset of size k, and we can expect bad news.

Theorem 4.1 (Best Subset Regression is NP-Hard)

Solving the best subset loss is NP-hard.

Proof.

Even though best subset regression is infeasible, we can still approximate best subset regression in two different ways.

1. A greedy approximation leads to *forward stepwise regression*.
2. A convex relaxation of the problem leads to the *Lasso* regression.

It turns out that the theoretical guarantees and computational time for both are the same, but the Lasso is much more popular. It may be due to a cleaner form or that it's easier to study, but who knows.

4.2 Forward Stepwise Regression

Forward stepwise regression is a greedy algorithm that starts with an empty set of covariates and adds the covariate that most improves the fit. It avoids the NP-hardness of the best subset regression by adding covariates one by one.

Definition 4.2 (Greedy Forward Stepwise Regression)

Given your data $\mathcal{D}$, let's first standardize it to have mean 0 and variance 1.[a] You start off with a set $\mathcal{Q} = \{\}$ and choose the number of parameters K.

1. With each covariate $X = (X_1, \ldots, X_n)$, we compute the correlation between it and the Y, which reduces to the inner product (since we standardized).

$$\rho_j = \langle Y, X_{:,j} \rangle = \frac{1}{n} \sum_{i=1}^n Y_i X_{ji} \tag{70}$$

2. Then, we take the covariate index that has the highest empirical correlation with Y, add it to $\mathcal{Q}$ and regress Y only on this covariate.

$$q_1 = \underset{j}{\operatorname{argmax}}\, \rho_j, \quad \mathcal{Q} = \{q_1\}, \quad \hat{\beta}_{q_1} = \underset{\beta}{\operatorname{argmin}} \frac{1}{n} ||Y - X_{:,q_1}\beta||^2 \tag{71}$$

3. Then you repeat the process. You take the residual values $r = Y - X_{:,q_1}\hat{\beta}_{q_1} \in \mathbb{R}^n$ compute the correlation between r and the remaining covariates, and pick our the maximum covariate index q_2. Then, you *repeat the regression from start* with these two covariates

$$q_2 = \underset{j}{\operatorname{argmax}} \langle r, X_{:;j} \rangle, \quad \mathcal{Q} = \{q_1, q_2\}, \quad \hat{\beta}_{q_1,q_2} = \underset{\beta}{\operatorname{argmin}} \frac{1}{n} ||Y - X_{:,[q_1,q_2]}\beta||^2 \tag{72}$$

   Note that you're not going to get the same coefficient for $\hat{\beta}_{q_1}$ as before since you're doing two variable regression.

4. You get out the residual values $r = Y - X_{:,[q_1,q_2]}\hat{\beta}_{q_1,q_2} \in \mathbb{R}^n$ and keep repeating this process until you have K covariates in $\mathcal{Q}$.

[a] This may or may not be a good idea, since the variance of each covariate can tell you a lot about the importance of the covariate.

Again, there is a bias variance tradeoff in choosing the number of covariates K, but through cross-validation, we can find the optimal K. It is also easy to add constraints, e.g. if we wanted to place a restriction that two adjacent covariates can't be chosen, we can easily add this to the algorithm.

Theorem 4.2 (Rate of Convergence for Stepwise Regression)

Let $\hat{f}_K$ be the optimal regressor we get from K covariates in stepwise regression. Then, we have something like

$$\|f - \hat{f}\|^2 \leq c\|f - f_K\|^2 + \frac{\log n}{\sqrt{n}} \tag{73}$$

This turns out to be the optimal rate.

What this is saying that forward stepwise gets to within about $\frac{1}{\sqrt{n}}$ of what you would get if you did the perfect best subset regression. Another interesting property is that this bound is dimensionless, which makes sense since we are approximating the best K-term linear predictor (not the true regressor), which is a weaker claim. On the other hand, if we use nonparametric estimators to estimate the true regressor, then we will get the curse of dimensionality.

4.3 Bias Variance Tradeoff

4.4 Stagewise Regression

Stagewise regression is a variant of forward stepwise regression where we add the covariate that most improves the fit, but we only take a small step in that direction. This is useful when we have a lot of covariates and we don't want to overfit.

5 Lasso Regression

The Lasso approximates the best subset regression by using a convex relaxation. In particular, the norm $||\beta||_0$ is not convex, but the L1 norm $||\beta||_1$ is. Therefore, we want relax our constraint equation as such:

$$\operatorname*{argmin}_{||\beta||_0 \leq L} r(\beta) \mapsto \operatorname*{argmin}_{||\beta||_1 \leq L} r(\beta) \tag{74}$$

This gives us a convex problem, which we can then solve. In fact, it turns out that optimizing the risk given the L1 restriction on the norm is equivalent to minimizing the risk plus a L1 penalty, as this is the Lagrangian form of the original equation (this is in convex optimization). Therefore, there exists a pair (L, λ) for which the two problems are equivalent

$$\operatorname*{argmin}_{||\beta||_1 \leq L} r(\beta) = \operatorname*{argmin}_{\beta} r(\beta) + \lambda ||\beta||_1 \tag{75}$$

Definition 5.1 (LASSO Regression)

In **lasso regression**, we minimize the loss defined

$$\hat{R}(\beta) = \frac{1}{n} \sum_{i=1}^n (y^{(i)} - \beta^T x^{(i)})^2 + \lambda ||\beta||_1 \tag{76}$$

where we penalize according to the L1 norm of the coefficients.

A question arises: Why use the L1 norm? The motivation behind this is that we want to model the L0 norm as much as possible but at the same time we want it to be convex. This turns out to be precisely the L1 norm. Unfortunately, there is no closed form solution for this estimator, but in convex optimization, we can prove that this estimator is sparse. That is, for large enough λ, many of the components of $\hat{\beta}$ are 0. The classical intuition for this is the figure below, where the equipotential lines have "corners." In fact for any $0 < p < 1$, there are also corners, but the problem with using these p-norms is that they are not convex.

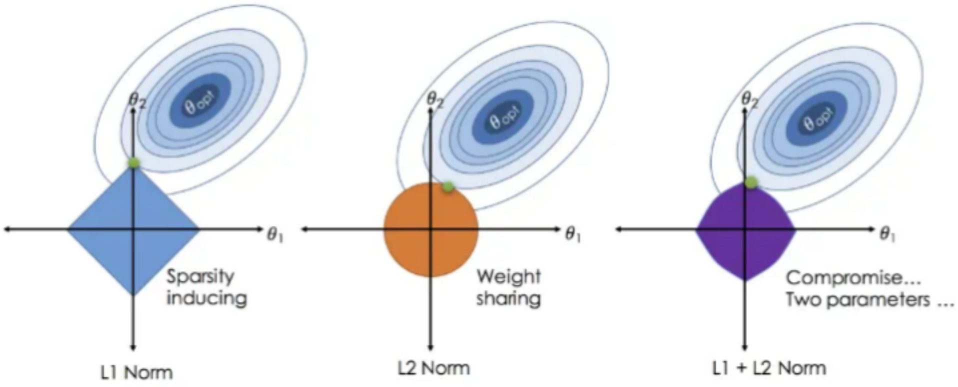

Figure 2: The ridge regularizer draws equipotential circles in our parameter space. The lasso draws a diamond, which tends to give a sparser solution since the loss is most likely to "touch" the corners of the contour plots of the regularizer. The elastic net is a linear combination of the ridge and lasso regularizers.

To motivate this even further, let us take the two vectors

$$a = \left(\frac{1}{\sqrt{d}}, \ldots, \frac{1}{\sqrt{d}} \right) \qquad b = (1, 0, \ldots, 0) \tag{77}$$

Then the L0, L1, and L2 norms of a are $d, \sqrt{d}, 1$ and those of b are $1, 1, 1$. We want to choose a norm that capture the sparsity of b and distinguishes it from b., The L0 norm clearly does this, but the L2 norm does not. The L1 norm is a good compromise between the two.

Code 5.1 (MWS of Lasso Regression in scikit-learn)

```python
from sklearn.linear_model import Lasso

X = np.random.randn(10, 5)
y = np.random.randn(10)
# regularization parameter
model = Lasso(alpha=1e-1)
model.fit(X, y)
print(model.score(X, y))
print(model.intercept_)
print(model.coef_)
print(model.predict(np.array([[1, 2, 3, 4, 5]])))
```

```
0.47590269719236045
-0.8861298412689853
[0.         0.10767647
    0.24172197 0.7427863  0.
         ]
[3.02553422]
.
.
.
.
.
```

5.1 Bias Variance Tradeoff

5.2 Concentration Bounds

This now raises the question of how to determine a suitable regularization parameter λ. The next theorem shows a nice concentration property of the Lasso for bounded covariates.

Theorem 5.1 (Concentration of Lasso)

Given (X, Y), assume that $|Y| \leq B$ and $\max_j |X_j| \leq B$. Let

$$\beta^* = \underset{||\beta||_1 \leq L}{\operatorname{argmin}} \, r(\beta) \tag{78}$$

be the best sparse linear predictor in the L1 sense, where $r(\beta) = \mathbb{E}[(Y - \beta^T X)^2]$. Let our lasso estimator be

$$\hat{\beta} = \underset{||\beta||_1 \leq L}{\operatorname{argmin}} \, \hat{r}(\beta) = \underset{||\beta||_1 \leq L}{\operatorname{argmin}} \, \frac{1}{n} \sum_{i=1}^n (Y_i - \beta^T X_i)^2 \tag{79}$$

which minimizes the empirical risk. Then, with probability at least $1 - \delta$,

$$r(\hat{\beta}) \leq r(\beta^*) + \sqrt{\frac{16(L+1)^4 B^2}{n} \log\left(\frac{\sqrt{2}d}{\sqrt{\delta}}\right)} \tag{80}$$

Proof.

5.3 Optimization

Soft Thresholding and Proximal Gradient Descent

6 Robust Regression

7 Bayesian Linear Regression

7.1 Regularization with Priors

We will now demonstrate how having a normal $\alpha\mathbf{I}$ prior around the origin in a Bayesian setting is equivalent to having a ridge penalty of $\lambda = \sigma^2/\alpha^2$ in a frequentist setting. If we have a Gaussian prior of form

$$p(\mathbf{w} \mid \alpha^2) = N(\mathbf{w} \mid \mathbf{0}, \alpha^2\mathbf{I}) = \bigg(\frac{1}{2\pi\alpha^2}\bigg)^{M/2} \exp\bigg(-\frac{1}{2\alpha^2}||\mathbf{w}||_2^2\bigg)$$

We can use Bayes rule to compute

$$\begin{aligned} p(\mathbf{w} \mid \mathbf{X}, \mathbf{Y}, \alpha^2, \sigma^2) &\propto p(\mathbf{Y} \mid \mathbf{w}, \mathbf{X}, \alpha^2, \sigma^2)\, p(\mathbf{w} \mid \mathbf{X}, \alpha^2, \sigma^2) \\ &= \bigg[\prod_{n=1}^N p(y^{(n)} \mid \mathbf{w}, \mathbf{x}^{(n)}, \alpha^2, \sigma^2)\bigg] p(\mathbf{w} \mid \mathbf{X}, \alpha^2, \sigma^2) \\ &= \bigg[\prod_{n=1}^N \frac{1}{\sqrt{2\pi\sigma^2}} \exp\bigg(-\frac{(y^{(n)} - h_{\mathbf{w}}(x^{(n)}))^2}{2\sigma^2}\bigg)\bigg] \cdot \bigg(\frac{1}{2\pi\alpha^2}\bigg)^{M/2} \exp\bigg(-\frac{1}{2\alpha^2}||\mathbf{w}||_2^2\bigg) \end{aligned}$$

and taking the negative logarithm gives us

$$\ell(\mathbf{w}) = \frac{1}{2\sigma^2}\sum_{n=1}^N \big(y^{(n)} - h_{\mathbf{w}}(\mathbf{x}^{(n)})\big)^2 + \frac{N}{2}\ln\sigma^2 + \frac{N}{2}\ln(2\pi) - \frac{M}{2}\ln(2\pi\alpha^2) + \frac{1}{2\alpha^2}||\mathbf{w}||_2^2$$

taking out the constant terms relative to $\mathbf{w}$ and multiplying by $2\sigma^2$ (which doesn't affect optima) gives us the ridge penalized error with a penalty term of $\lambda = \sigma^2/\alpha^2$.

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^N \big(y^{(n)} - h_{\mathbf{w}}(\mathbf{x}^{(n)})\big)^2 + \frac{\sigma^2}{\alpha^2}||\mathbf{w}||_2^2$$

But minimizing this still gives a point estimate of $\mathbf{w}$, which is not the full Bayesian treatment. In a Bayesian setting, we are given the training data $(\mathbf{X}, \mathbf{Y})$ along with a new test point $\mathbf{x}'$ and want to evaluate the predictive distribution $p(y \mid \mathbf{x}', \mathbf{X}, \mathbf{Y})$. We can do this by integrating over $\mathbf{w}$.

$$\begin{aligned} p(y \mid \mathbf{x}', \mathbf{X}, \mathbf{Y}) &= \int p(y \mid \mathbf{x}', \mathbf{w}, \mathbf{X}, \mathbf{Y})\, p(\mathbf{w} \mid \mathbf{x}', \mathbf{X}, \mathbf{Y})\, d\mathbf{w} \\ &= \int p(y \mid \mathbf{x}', \mathbf{w})\, p(\mathbf{w} \mid \mathbf{X}, \mathbf{Y})\, d\mathbf{w} \end{aligned}$$

where we have omitted the irrelevant variables, along with α^2 and σ^2 to simplify notation. By substituting the posterior $p(\mathbf{w} \mid \mathbf{X}, \mathbf{Y})$ with a normalized version of our calculation above and by noting that

$$p(y \mid \mathbf{x}', \mathbf{w}) = N(y \mid h_{\mathbf{w}}(\mathbf{x}'), \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\bigg(-\frac{\big(y - h_{\mathbf{w}}(\mathbf{x}')\big)^2}{2\sigma^2}\bigg)$$

Now this integral may or may not have a closed form, but if we consider the polynomial regression with the hypothesis function of form

$$h_{\mathbf{w}}(x) = w_0 + w_1 x + w_2 x^2 + \ldots + w_{M-1}x^{M-1}$$

then this integral turns out to have a closed form solution given by

$$p(y \mid \mathbf{x}', \mathbf{X}, \mathbf{Y}) = N\big(y \mid m(x'), s^2(x')\big)$$

where

$$\begin{aligned} m(x') &= \frac{1}{\sigma^2} \boldsymbol{\phi}(x')^T \mathbf{S} \bigg(\sum_{n=1}^N \boldsymbol{\phi}(x^{(n)}) y^{(n)} \bigg) \\ s^2(x') &= \sigma^2 + \boldsymbol{\phi}(x')^T \mathbf{S} \boldsymbol{\phi}(x') \\ \mathbf{S}^{-1} &= \alpha^{-2} \mathbf{I} + \frac{1}{\sigma^2} \sum_{n=1}^N \boldsymbol{\phi}(x^{(n)}) \boldsymbol{\phi}(x')^T \end{aligned}$$

and $\boldsymbol{\phi}(x)$ is the vector of functions $\phi_i(x) = x^i$ from $i = 0, \ldots, M-1$.

8 Generalized Linear Models

Remember the linear model looked like this, where we use the conventional β notation to represent parameters.

$$Y = X^T\beta + \epsilon, \ \ \epsilon \sim N(0, \sigma^2 I) \tag{81}$$

which implies that $Y \mid X \sim N(X^T\beta, \sigma^2 I)$. Basically, given x, I assume some distribution of Y, and the value of x will help me guess what the mean of this distribution is. Note that we in here assume that only the mean depends on X. I could potentially have something crazy, like

$$Y \mid X \sim N(X^T\beta, (X^T\gamma)(XX^T + I))$$

where the covariance will depend on X, too, but in this case we only assume that that mean is dependent on X.

$$Y \mid X \sim N(\mu(X), \sigma^2 I)$$

where in the linear model, $\mu(X) = X^T\beta$. So, there are three assumptions we are making here:

1. $Y \mid X$ is Gaussian.
2. X only affects the mean of $Y \mid X$, written $\mathbb{E}[Y \mid X] = \mu(X)$.
3. X affects the mean in a linear way, such that $\mu(X) = X^T\beta$.

So the two things we are trying to relax are:

1. **Random Component**: the response variable $Y \mid X$ is continuous and normally distributed with mean $\mu = \mu(X) = \mathbb{E}[Y \mid X]$.
2. **Link**: I have a link that explains the relationship between the X and the μ, and this relationship is $\mu(X) = X^T\beta$.

So when talking about GLMs, we are not changing the fact that we have a linear function $X \mapsto X^T\beta$. However, we are going to assume that $Y \mid X$ now comes from a broader **family of exponential distributions**. Second, we are going to assume that there exists some **link function** g

$$g(\mu(X)) = X^T\beta$$

Admittedly, this is not the most intuitive way to think about it, since we would like to have $\mu(X) = f(X^T\beta)$, but here we just decide to call $f = g^{-1}$. Therefore, if I want to give you a GLM, I just need to give you two things: the conditional distribution $Y \mid X$, which can be any distribution in the exponential family, and the link function g.

We really only need this link function due to compatibility reasons. Say that $Y \mid X \sim \text{Bern}(p)$. Then, $\mu(X) = \mathbb{E}[Y \mid X]$ always lives in $[0, 1]$, but $X^T\beta$ always lives in $\mathbb{R}$. We want our model to be realistic, and we can clearly see the problem shown in Figure 3.

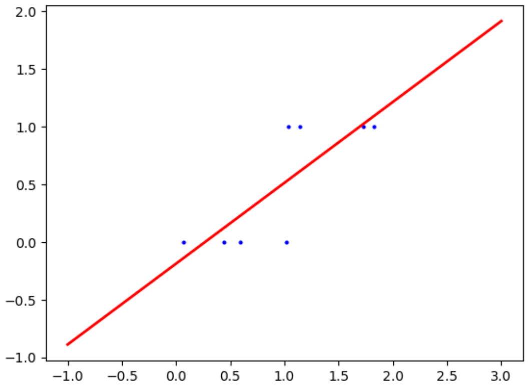

Figure 3: Fitting a linear model for Bernoulli random variables will predict a mean that is outside of $[0, 1]$ when getting new datapoints.

If $Y \mid X$ is some exponential distribution, then its support is always positive and so $\mu(X) > 0$. But if we stick to the old form of $\mu(X) = X^T\beta$, then $\text{Im}(\mu) = \mathbb{R}$, which is not realistic when we predict negative values. Let's take a couple examples:

Example 8.1 (Disease Epidemic)

In the early stages of a disease epidemic, the rate at which new cases occur can often increase exponentially through time. Clearly, $\mu(X) = \mathbb{E}[Y \mid X]$ should be positive and we should have some sort of exponential trend. Hence, if $\mu(x)$ is the expected number of cases on data x, a model of the form

$$\mu(x) = \gamma \exp(\delta x) \tag{82}$$

seems appropriate, where γ and δ are simply scaling factors. Clearly, $\mu(X)$ is not of the form $f(X^T\beta)$. So what I do is to transform μ in such a way that I can get something that is linear.

$$\log(\mu(X)) = \log(\gamma) + \delta X \tag{83}$$

which is now linear in X, of form $\beta_0 + \beta_1 X$. This will have some effects, but this is what needs to be done to have a genearlized linear model. Note that what I did to μ was take the log of it, and so the link function is $g = \log$, called the **log-link**. Now that we have chosen the g, we still need to choose what the conditional distribution $Y \mid X$ would be. This is determined by speaking with industry professionals, experience, and convenience. In this case, Y is a count, and since this must be a discrete distribution. Since it is not bounded above, we think Poisson.

Example 8.2 (Prey Capture Rate)

The rate of capture of preys, Y, by a hunting animal, tends to increase with increasing density of prey X, but eventually level off when the predator is catching as much as it can cope with. We want to find a perhaps concave function that levels off, and suitable model might be

$$\mu(X) = \frac{\alpha X}{h + X} \tag{84}$$

where α represents the maximum capture rate, and h represents the prey density at which the capture rate is half the maximum rate. Again, we must find some transformation g that turns this into a linear function of X, and what we can do it use the **reciprocal-link**.

$$\frac{1}{\mu(X)} = \frac{h+X}{\alpha X} = \frac{h}{\alpha}\frac{1}{X} + \frac{1}{\alpha} \tag{85}$$

The standard deviation of capture rate might be approximately proportional to the mean rate, suggesting the use of a Gamma distribution for the response.

Example 8.3 (Kyphosis Data)

The Kyphosis data consist of measurements on 81 children following corrective spinal surgery. The binary response variable, Kyphosis, indicates the presence or absence of a postoperative deforming. The three covariates are: age of the child in months, number of the vertebrae involved in the operation, and the start of the range of the vertebrae involved. The response variable is binary so there is no choice: $Y \mid X$ is Bernoulli with expected value $\mu(X) \in (0,1)$. We cannot write $\mu(X) = X^T\beta$ because the right hand side ranges through $\mathbb{R}$, and so we find an invertible function that squishes $\mathbb{R}$ to $(0,1)$, and so we can choose basically any CDF.

For clarification, when writing a distribution like Bernoulli(p), or Binomial(n,p), Poisson(λ), or $N(\mu, \sigma^2)$, the hyperparameters that we usually work with we will denote as $\boldsymbol{\theta}$, and the space that this $\boldsymbol{\theta}$ lives in will denote Θ. For example, for the Bernoulli, $\Theta = [0,1]$, and for Poisson, $\Theta = [0,+\infty)$.

Ultimately, a GLM consists of three steps:

1. The observed input X enters the model through a linear function $\beta^T X$.
2. The conditional mean of response, is represented as a function of the linear combination
$$\mathbb{E}[Y \mid X] = \mu = f(\beta^T X) \tag{86}$$
3. The observed response is drawn from an exponential family distribution with conditional mean μ.

8.1 Exponential Family

We can write the pdf of a distribution as a function of the input x and the hyperparameters θ, so we can write $P_\theta(x) = p(\theta, x)$. For now, let's think that both $x, \theta \in \mathbb{R}$. Think of all the functions that depend on θ and x. There are many of them, but we want θ and x to interact in a certain way. The way that I want them to interact with each other is that they are multiplied within an exponential term. Now clearly, this is not a very rich family, so we are just slapping some terms that depend only on θ and only on x.

$$p_\theta(x) = \exp(\theta x) h(x) c(\theta)$$

But now if $\theta \in \mathbb{R}^k$ and $x \in \mathbb{R}^q$, then we cannot simply take the product nor the inner product, but what we can do is map both of them into a space that has the same dimensions, so I can take the inner product. That is, let us map $\boldsymbol{\theta} \mapsto \boldsymbol{\eta}(\boldsymbol{\theta}) \in \mathbb{R}^k$ and $\mathbf{x} \mapsto \mathbf{T}(\mathbf{x}) \in \mathbb{R}^k$, and so our exponential distribution form would be generalized into something like

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \exp\big[\boldsymbol{\eta}(\boldsymbol{\theta}) \cdot \mathbf{T}(\mathbf{x})\big] h(\mathbf{x}) c(\boldsymbol{\theta})$$

We can think of $c(\boldsymbol{\theta})$ as the normalizing term that allows us to integrate the pdf to 1.

$$\int_{\mathcal{X}} p_{\boldsymbol{\theta}}(\mathbf{x}) = c(\boldsymbol{\theta}) \int \exp\big[\boldsymbol{\eta}(\boldsymbol{\theta}) \cdot \mathbf{T}(\mathbf{x})\big] h(\mathbf{x})\, d\mathbf{x}$$

We can just push the $c(\boldsymbol{\theta})$ term into the exponential by letting $c(\boldsymbol{\theta}) = e^{-\log(c(\boldsymbol{\theta}))^{-1}}$ to get our definition.

Definition 8.1 (Exponential Family)

A **k-parameter exponential family** is a family of distributions with pdf/pmf of the form

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \exp\big[\boldsymbol{\eta}(\boldsymbol{\theta}) \cdot \mathbf{T}(\mathbf{x}) - B(\boldsymbol{\theta})\big] h(\mathbf{x})$$

The h term, as we will see, will not matter in our maximum likelihood estimation, so we keep it outside the exponential.

1. $\boldsymbol{\eta}$ is called the **canonical parameter**. Given a distribution parameterized by the regular hyperparameters $\boldsymbol{\theta}$, we would like to parameterize it in a different way $\boldsymbol{\eta}$ under the function $\boldsymbol{\eta} : \Theta \to \mathbb{R}$
2. $\mathbf{T}(\mathbf{x})$ is called the **sufficient statistic**.
3. $h(\mathbf{x})$ is a nonnegative scalar function.
4. $B(\boldsymbol{\theta})$ is the normalizing factor.

Let's look at some examples.

Example 8.4 (Gaussian)

If we put the coefficient into the exponential and expand the square term, we get

$$p_\theta(x) = \exp\left(\frac{\mu}{\sigma^2} \cdot x - \frac{1}{2\sigma^2} \cdot x^2 - \frac{\mu^2}{2\sigma^2} - \log(\sigma\sqrt{2\pi})\right)$$

where

$$\boldsymbol{\eta}(\boldsymbol{\theta}) = \begin{pmatrix} \mu/\sigma^2 \\ -1/2\sigma^2 \end{pmatrix},\ T(x) = \begin{pmatrix} x \\ x^2 \end{pmatrix},\ B(\theta) = \frac{\mu^2}{2\sigma^2} + \log(\sigma\sqrt{2\pi}),\ h(x) = 1$$

This is not a unique representation since we can take the $\log(\sqrt{2\pi})$ out of the exponential, but why bother to do this when we can just stuff everything into B and keep h simple.

Example 8.5 (Gaussian with Known Variance)

If we have known variance, we can write the Gaussian pdf as

$$p_\theta(x) = \exp\left[\frac{\mu}{\sigma} \cdot \frac{x}{\sigma} - \frac{\mu^2}{2\sigma^2}\right] \cdot \frac{1}{\sigma\sqrt{2\pi}} e^{x^2/2\sigma^2}$$

where

$$\eta(\theta) = \frac{\mu}{\sigma},\ T(x) = \frac{x}{\sigma},\ B(\theta) = \frac{\mu^2}{2\sigma^2},\ h(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{x^2/2\sigma^2}$$

Example 8.6 (Bernoulli)

The pmf of a Bernoulli with θ is

$$\begin{aligned} p_\theta(x) &= \theta^x (1-\theta)^{(1-x)} \\ &= \exp\big[x\log(\theta) + (1-x)\log(1-\theta)\big] \\ &= \exp\left(x \log\left[\frac{\theta}{1-\theta}\right] - \log\left[\frac{1}{1-\theta}\right]\right) \end{aligned}$$

where

$$\eta(\theta) = \log\left[\frac{\theta}{1-\theta}\right],\ T(x) = x,\ B(\theta) = \log\left[\frac{1}{1-\theta}\right],\ h(x) = 1$$

Example 8.7 (Binomial with Known Number of Trials)

We can transform a binomial with known N as

$$\begin{aligned} p_\theta(x) &= \binom{N}{x} \theta^x (1-\theta)^{1-x} \\ &= \exp\bigg[x \ln\Big(\frac{\theta}{1-\theta}\Big) + \ln(1-\theta) \bigg] \cdot \binom{N}{x} \end{aligned}$$

where

$$\eta(\theta) = \ln\Big(\frac{\theta}{1-\theta}\Big),\ T(x) = x,\ B(\theta) = \ln(1-\theta),\ h(x) = \binom{N}{x}$$

Example 8.8 (Poisson)

The pmf of Poisson with θ can be expanded

$$\begin{aligned} p_\theta &= \frac{\theta^{-x}}{x!} e^{-\theta} \\ &= \exp\big[-\theta + x\log(\theta) - \log(x!) \big] \\ &= \exp\big[x\log(\theta) - \theta \big] \frac{1}{x!} \end{aligned}$$

where

$$\eta(\theta) = \log(\theta),\ T(x) = x,\ B(\theta) = \theta,\ h(x) = \frac{1}{x!}$$

However, the uniform is not in here. In fact, any distribution that has a support that does not depend on the parameter is not an exponential distribution.

Let us now focus on one parameter families where $\theta \in \Theta \subset \mathbb{R}$, which do not include the Gaussian (with unknown mean and variance, Gamma, multinomial, etc.), which has a pdf written in the form

$$p_\theta(x) = \exp\big[\eta(\theta)\, T(x) - B(\theta)\big]\, h(x)$$

8.1.1 Canonical Exponential Family

Now a common strategy in statistical analysis is to reparamaterize a probability distribution. Suppose a family of probability distributions $\{P_\theta\}$ is parameterized by $\theta \in \Theta \subset \mathbb{R}$. If we have an invertible function $\eta : \Theta \to \mathcal{T} \subset \mathbb{R}$, then we can paramaterize the same family with η rather than θ, with no loss of information. Typically, it is the case that η is invertible for exponential families, so we can just reparameterize the whole pdf and write

$$p_\eta(x) = \exp\big[\eta\, T(x) - \phi(\eta)\big]\, h(x)$$

where $\phi = B \circ \eta^{-1}$.

Definition 8.2 (Canonical One-Parameter Exponential Family)

A family of distributions is said to be in **canonical one-parameter exponential family** if its density is of form

$$p_\eta(x) = \exp\big[\eta\, T(x) - \phi(\eta)\big]\, h(x)$$

which is a subfamily of the exponential family. The function ψ is called the **cumulant generating function**.

Before we move on, let us just provide a few examples.

Example 8.9 (Poisson)

The Poisson can be represented as

$$p_\theta(x) = \exp\big[x \log\theta - \theta\big]\,\frac{1}{x!}$$

Now let $\eta = \log\theta \implies \theta = e^\eta$. So, we can reparamaterize the density as

$$p_\eta(x) = \exp\big[x\eta - e^\eta\big]\,\frac{1}{x!}$$

where $P_\eta = \text{Poisson}(e^\eta)$ for $\eta \in \mathcal{T} = \mathbb{R}$, compared to $P_\theta = \text{Poisson}(\theta)$ for $\theta \in \Theta = \mathbb{R}^+$.

Example 8.10 (Gaussian)

Recall that the Gaussian with known parameter σ^2 and unknown $\theta = \mu$ is in the exponential family, since we can expand it as

$$p_\theta(x) = \exp\left[\frac{\mu}{\sigma^2}\cdot x - \frac{\mu^2}{2\sigma^2}\right]\cdot\frac{1}{\sigma\sqrt{2\pi}}e^{x^2/2\sigma^2}$$

We can perform the change of parameter $\eta = \mu^2/2\sigma^2 \implies \mu = \sigma^2\eta$, and substituting this in will give the canonical representation

$$p_\eta(x) = \exp\left[\eta x - \frac{\sigma^2\eta^2}{2}\right]\cdot\frac{1}{\sigma\sqrt{2\pi}}e^{x^2/2\sigma^2}$$

where now $P_\eta = N(\sigma^2\eta, \sigma^2)$ for $\eta \in \mathcal{T} = \mathbb{R}$, compared to $P_\theta = N(\theta, \sigma^2)$ for $\theta \in \Theta = \mathbb{R}$, which is basically the same space.

Example 8.11 (Bernoulli)

The Bernoulli has an exponential form of

$$p_\theta(x) = \exp\left[x\log\left(\frac{\theta}{1-\theta}\right) + \log(1-\theta)\right]$$

Now setting $\eta = \log\left(\frac{\theta}{1-\theta}\right) \implies \theta = \frac{1}{1+e^{-\eta}}$, and so $B(\theta) = -\log(1-\theta) = -\log\left(\frac{e^{-\eta}}{1+e^{-\eta}}\right) = \log(1+e^\eta) = \psi(\eta)$, and so the canonical paramaterization is

$$p_\eta(x) = \exp\big[x\eta - \log(1+e^\eta)\big]$$

We present two useful properties of the exponential family.

Theorem 8.1 (Moments)

Let random variable X be in the canonical exponential family P_η

$$p_\eta(x) = e^{\eta T(x) - \psi(\eta)}h(x)$$

Then, the expectation and variance are encoded in the cumulant generating function in the following way

$$\mathbb{E}[T(X)] = \psi'(\eta) \qquad \text{Var}[T(X)] = \psi''(\eta)$$

Proof.

Example 8.12 ()

We show that this is consistent with the Poisson, normal, and Bernoulli distributions.

1. In the Poisson, $\psi(\eta) = e^\eta$, and so $\psi'(\eta) = e^\eta = \theta = \mathbb{E}[X]$. Taking the second derivative gives $\psi''(\eta) = e^\eta = \theta = \mathrm{Var}[X]$, too.
2. In the Normal with known variance σ^2, we have $\psi(\eta) = \frac{1}{2}\sigma^2\eta^2$. So

$$\begin{aligned} \mathbb{E}[X] &= \psi'(\eta) = \sigma^2\eta = \mu \\ \mathrm{Var}[X] &= \psi''(\eta) = \sigma^2 \end{aligned}$$

3. In the Bernoulli, we have $\psi(\eta) = \log(1 + e^{-\eta})$. Therefore,

$$\begin{aligned} \mathbb{E}[X] &= \psi'(\eta) = \frac{x^\eta}{1 + x^\eta} = \frac{1}{1 + e^{-\eta}} = \theta \\ \mathrm{Var}[X] &= \psi''(\eta) = -\left(\frac{1}{1 + e^{-\eta}}\right)^2 e^{-\eta} \cdot -1 = \theta^2 \cdot \frac{1 - \theta}{\theta} = \theta(1 - \theta) \end{aligned}$$

Theorem 8.2 (Convexity)

Consider a canonical exponential family with density

$$p_\eta(x) = e^{\eta T(x) - \psi(\eta)} h(x)$$

and natural parameter space $\mathcal{T}$. Then, the set $\mathcal{T}$ is convex, and the cumulant generating function ψ is convex on $\mathcal{T}$.

Proof.

This can be proven using Holder's inequality. However, from the theorem above, note that $\mathrm{Var}[T(X)] = \psi''(\eta)$ must be positive since we are talking about variance. This implies that the second derivative of ψ is positive, and therefore must be convex.

We will look at a subfamily of the exponential family. Now remember that we introduce the functions $\boldsymbol{\eta}$ and $\mathbf{T}$ so that we can capture a much broader range of distributions, but if we have one parameter $k = 1$, then we can just set $\boldsymbol{\eta}(\boldsymbol{\theta})$ to be the new parameter θ. The **canonical exponential family** for $k = 1, y \in \mathbb{R}$, is defined to have the pdf

$$f_\theta(y) = \exp\left(\frac{y\theta - b(\theta)}{\phi} + c(y, \phi)\right) \tag{87}$$

where

$$h(y) = \exp\big(c(y, \phi)\big) \tag{88}$$

If ϕ is known, this is a one-parameter exponential family with θ being the **canonical parameter**, and if ϕ is unknown, the $h(y)$ term will not depend on θ, which we may not be able to split up into the exponential pdf form. In this case ϕ is called the **dispersion parameter**. For now, we will always assume that ϕ is known.

We can prove this for all other classes, too. We can think of the $c(y, \phi)$ as just a term that we stuff every other term into. What really differentiates the different distributions of the canonical exponential family is the $b(\theta)$. The form of b will determine whether this distribution is a Gaussian, or a Bernoulli, or etc. This b will capture information about the mean, the variance, the likelihood, about everything.

8.2 Cumulant Generating Function

Definition 8.3 (Score)

The **score** is the gradient of the log-likelihood function with respect to the parameter vector. That is, given that $L(\boldsymbol{\theta})$ is the likelihood, then

$$s(\boldsymbol{\theta}) \coloneqq \frac{\partial \log L(\boldsymbol{\theta}; \mathbf{x})}{\partial \boldsymbol{\theta}}$$

which gives a row covector.

Now, remember that the score also depends on the observations $\mathbf{x}$. If we rewrite the likelihood as a probability density function $L(\boldsymbol{\theta}; \mathbf{x}) = f(\mathbf{x}; \boldsymbol{\theta})$, then we can say that the expected value of the score is equal to 0, since

$$\begin{aligned}
\mathbb{E}[s(\boldsymbol{\theta})] &= \int_{\mathcal{X}} f(\mathbf{x}; \boldsymbol{\theta}) \frac{\partial}{\partial \boldsymbol{\theta}} \log L(\boldsymbol{\theta}; \mathbf{x})\, d\mathbf{x} \\
&= \int_{\mathcal{X}} f(\mathbf{x}; \boldsymbol{\theta}) \frac{1}{f(\mathbf{x}; \boldsymbol{\theta})} \frac{\partial f(\mathbf{x}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\, dx \\
&= \frac{\partial}{\partial \boldsymbol{\theta}} \int_{\mathcal{X}} f(\mathbf{x}; \boldsymbol{\theta})\, d\mathbf{x} \\
&= \frac{\partial}{\partial \boldsymbol{\theta}} 1 = 0
\end{aligned}$$

where we take a leap of faith in switching the derivative and integral in the penultimate line. Furthermore, we can get the second identity

$$\mathbb{E}\bigg[\frac{\partial^2 \ell}{\partial \theta^2}\bigg] + \mathbb{E}\bigg[\frac{\partial \ell}{\partial \theta}\bigg]^2 = 0$$

We can apply these two identities as follows. Since

$$\ell(\theta) = \frac{Y\theta - b(\theta)}{\phi} + c(Y; \phi)$$

therefore

$$\frac{\partial \ell}{\partial \theta} = \frac{Y - b'(\theta)}{\phi}$$

which yields

$$0 = \mathbb{E}\bigg[\frac{\partial \ell}{\partial \theta}\bigg] = \frac{\mathbb{E}[Y] - b'(\theta)}{\phi} \implies \mathbb{E}[Y] = \mu = b'(\theta)$$

On the other hand, we have

$$\frac{\partial^2 \ell}{\partial \theta^2} + \bigg(\frac{\partial \ell}{\partial \theta}\bigg)^2 = -\frac{b''(\theta)}{\phi} + \bigg(\frac{Y - b'(\theta)}{\phi}\bigg)^2$$

and from the previous result, we get

$$\frac{Y - b'(\theta)}{\phi} = \frac{Y - \mathbb{E}[Y]}{\phi}$$

together with the second identity, yields

$$0 = -\frac{b''(\theta)}{\phi} + \frac{\mathrm{Var}(Y)}{\phi^2} \implies \mathrm{Var}(Y) = \phi''(\theta)$$

Since variance is always positive, this implies that $b'' > 0$ and therefore b must be convex.

8.3 Link Functions

Now let's go back to GLMs. In linear models, we said that the conditional expectation of Y given $X = \mathbf{x}$ must be a linear function in x

$$\mathbb{E}[Y \mid X = \mathbf{x}] = \mu(\mathbf{x}) = \mathbf{x}^T \beta$$

But if the conditional distribution takes values in some subset of $\mathbb{R}$, such as $(0, 1)$, then it may not make sense to write this as a linear function, since $X^T\beta$ has an image spanning $\mathbb{R}$. So what we need is a link function that relates, i.e. transforms the restricted subset of μ, onto the real line, so that now you can express it of the form $X^T\beta$.

$$g\big(\mu(X)\big) = X^T \beta$$

Again, it is a bit more intuitive to talk about g^{-1}, which takes our $X^T\beta$ and transforms it to the values that I want, so we will talk about both of them simultaneously. If g is our link function, we want it to satisfy 3 requirements:

1. g is continuously differentiable
2. g is strictly increasing
3. $\text{Im}(g) = \mathbb{R}$, i.e. it spans the entire real line

This implies that g^{-1} exists, which is also continuously differentable and is strictly increasing.

Example 8.13 ()

If I have a conditional distribution...

1. that is Poisson, then we want our μ to be positive, and so we need a link function $g : \mathbb{R}^+ \to \mathbb{R}$. One choice would be the logarithm

$$g(\mu(X)) = \log\big(\mu(X)\big) = X^T\beta$$

2. that is Bernoulli, then we want our μ to be in $(0, 1)$ and we need a link function $g : (0, 1) \to \mathbb{R}$. There are 2 natural choices, which may be the **logit** function

$$g(\mu(X)) = \log\left(\frac{\mu(X)}{1 - \mu(X)}\right) = X^T\beta$$

or the **probit** function

$$g(\mu(X)) = \Phi^{-1}\big(\mu(X)\big) = X^T\beta$$

where Φ is the CDF of a standard Gaussian. The two functions can be seen in Figure 4.

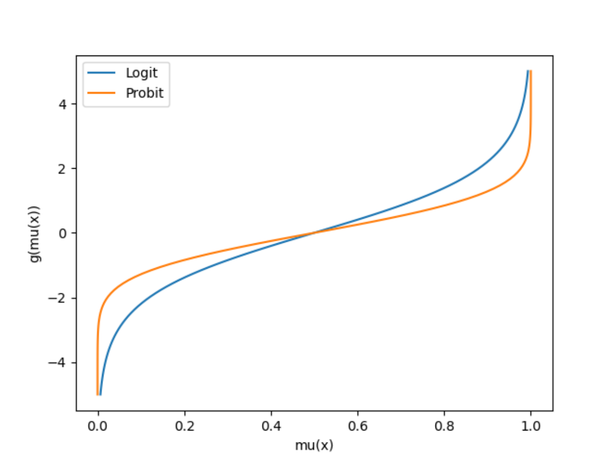

Figure 4: Logit and Probit Functions

Now there are many choices of functions we can take. In fact, if μ lives in $(0, 1)$, then we can really just take our favorite distribution that has a density that is supported everywhere in $\mathbb{R}$ and take the inverse cdf as our link. So far, we have no reason to prefer one function to another, but in the next section, we will see that there are more natural choices.

8.3.1 Canonical Link Functions

Now let's summarize what we have. We assume that the conditional distribution $Y \mid X = x$ follows a distribution in the exponential family, which we can completely characterize by the cumulant generating function ψ. For different values of x, the conditional distribution will be paramaterized by different $\eta(x)$, and the resulting distribution P_η will have some mean $\mu(x)$, which is usually not the natural parameter η. Now, let's forget about our knowledge that $\psi'(\eta) = \mu$, but we know that there is some relationship between $\eta \leftrightarrow \mu$.

Given an x, I need to use the linear predictor $x^T \beta$ to predict $\mu(x)$, which can be done through the link function g.

$$g\big(\mu(x)\big) = x^T \beta$$

Now what would be a natural way of choosing this g? Note that our natural parameter η for this canonical family takes value on the entire real line. I must construct a function g that maps μ onto the entire real line, and so why not make it map to η. Therefore, we have

$$\eta(x) = g\big(\mu(x)\big) = x^T \beta$$

Definition 8.4 (Canonical Link)

The function g that links the mean μ to the canonical parameter θ is called the **canonical link**.

$$g(\mu) = \theta$$

Now using our knowledge that $\psi'(\eta) = \mu$, we can see that

$$g = (\psi')^{-1}$$

This is indeed a valid link function.

1. $\psi'' > 0$ since it models the variance, and so ψ' is strictly increasing and so $g = (\psi')^{-1}$ is also strictly increasing.
2. The domain of ψ' is the real line since it takes in the natural parameter η which exists over $\mathbb{R}$, so $\text{Im}(g) = \mathbb{R}$.

So, given our cumulant generating function ψ and our link function g, both satisfying

$$\psi'(\eta) = \mu \text{ and } g(\mu) = x^T \beta$$

we can combine them to get

$$(g \circ \psi')(\eta) = g(\mu) = x^T \beta$$

and so, even though the mean of the response variable is not linear with respect to x, the value of $(g \circ \psi')(\eta)$ is indeed linear. In fact, if we choose the canonical link, then the equation

$$\eta = x^T \beta$$

means that the natural parameter of our conditional distribution in the exponential family is linear with respect to x! From this we can find the conditional mean $\mu(x)$.

The reason we focus on canonical link functions is because, when the canonical link is used, the components of the model (the parameters of the linear predictor) have an additive effect on the response variable in the transformed (linked) scale, which makes the interpretation of the results easier. It's also worth noting that while using the canonical link function has some desirable properties, it is not always the best or only choice, and other link functions may be used if they provide a better fit for the data or make more sense in the context of the problem at hand.

Let us evaluate some canonical link functions.

Example 8.14 ()

The Bernoulli has the canonical exponential form of

$$p_\eta(x) = \exp\big[x\eta - \log(1 + e^\eta)\big]$$

where $\eta = \log\big(\frac{\theta}{1-\theta}\big)$. Since we have prior knowledge that $\theta = \mu$ (i.e. the expectation of a Bernoulli is the hyperparameter θ itself), we have a function that maps $\mu \mapsto \eta$.

$$\eta = g(\mu) = \log\left(\frac{\mu}{1-\mu}\right)$$

which gives us our result. We can also take the inverse of $\psi' = \frac{e^\eta}{1+e^\eta}$ to get our result

$$g(\mu) = (\psi')^{-1}(\mu) = \log\left(\frac{\mu}{1-\mu}\right)$$

8.4 Likelihood Optimization

Now let us have a bunch of data points $\{(x_n, y_n)\}_{n=1}^N$. By our model assumption, we know that the conditional distribution $Y \mid X = x_n$ is now of an exponential family with parameter $\eta_n = \eta(x_n)$ and density

$$p_{\eta_n}(y_n) = \exp\big[y_n \eta_n - \psi(\eta_n)\big] h(y_n)$$

Now we want to do likelihood optimization on β (not η or μ), and to do this, we must rewrite the density function in a way so that it depends on β. Given a link function g, note the following relationship between β and η:

$$\begin{aligned}\eta_n = \eta(x_n) &= (\psi')^{-1}(\mu(x_n)) \\ &= (\psi')^{-1}\big(g^{-1}(x_n^T \beta)\big) \\ &= h(x_n^T \beta)\end{aligned}$$

where for shorthand notation, we define $h \coloneqq (g \circ \psi')^{-1}$. Subtituting this into the above likelihood, taking the product of all N samples, and logarithming the equation gives us the following log likelihood to optimize over β.

$$\ell(\beta) = \log \prod_{n=1}^N p_{\eta_n}(y_n) = \sum_{n=1}^N y_n h(x_n^T \beta) - \psi(h(x_n^T \beta))$$

where we dropped the $h(y_n)$ term at the end since it is a constant and does not matter. If g was the canonical link, then h is the identity, and we should have a linear relationship between $\eta(x_n) = x_n^T \beta$. This means that the η_n reduces only to $x_n^T \beta$, which is much more simple to optimize.

$$\ell(\beta) = \log \prod_{n=1}^N p_{\eta_n}(y_n) = \sum_{n=1}^N y_n x_n^T \beta - \psi(x_n^T \beta)$$

Note that the first term is linear w.r.t β, and ψ is convex, so the entire sum must be concave w.r.t. β. With this, we can bring in some tools of convex optimization to solve.

Bibliography

[GKKW02] L. Györfi, M. Kohler, A. Krzyzak, and H. Walk. *A Distribution-Free Theory of Nonparametric Regression*. Springer Series in Statistics. Springer New York, 2002.

[HKZ14] Daniel Hsu, Sham M. Kakade, and Tong Zhang. Random design analysis of ridge regression, 2014.