Lecture 7: Regression, Part 2
Modeling Social Data, Spring 2017
Columbia University

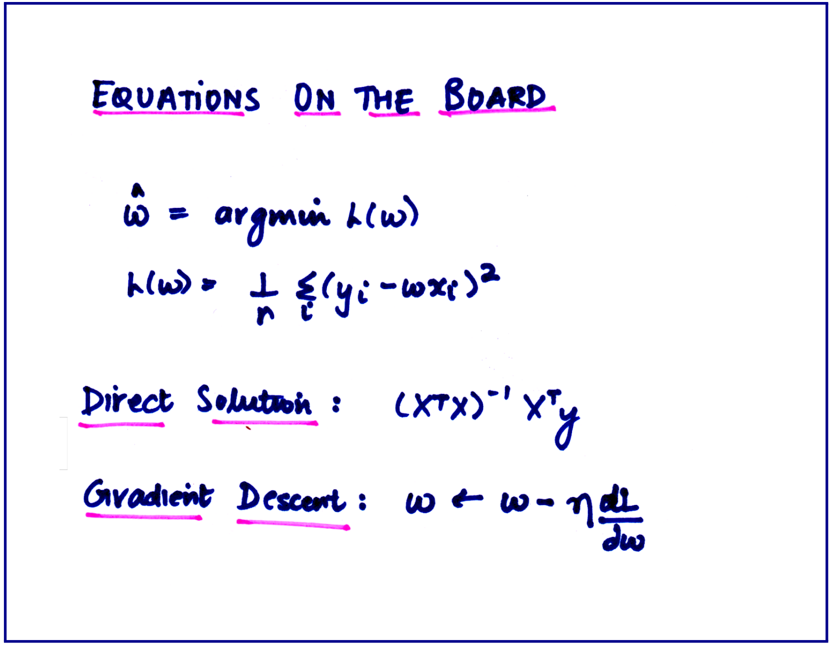
March 3, 2017

Notes from aa3917

1 Bits from the last lecture

The least squares closed form solution is expensive. It scales in a way such that it would need cubic time ($\mathcal{O}(n^3)$) to perform the operations. On huge datasets, cubic times gets really bad. For example, suppose there are 100000 features, we would be processing a 100000 x 100000 matrix. So, *Gradient Descent* is another option. It should get us to the bottom as the closed form solution. Each one of the steps takes $n * k$ operations. It is computationally cheap though it has certain downsides. It is iterative and you need to be careful of selecting learning rate η. For really wide datasets, even $n * k$ operations are a lot. In this case, we opted for *Stochastic Gradient Descent* . In SGD, we approximate the gradient with a random sample. It helps us control how many points to look at before computing the gradient.

Figure 1: Least Square Solutions



EQUATIONS ON THE BOARD

$$\hat{w} = \text{argmin } L(w)$$

$$L(w) = \frac{1}{n}\sum_i (y_i - wx_i)^2$$

Direct Solution : $(X^TX)^{-1}X^Ty$

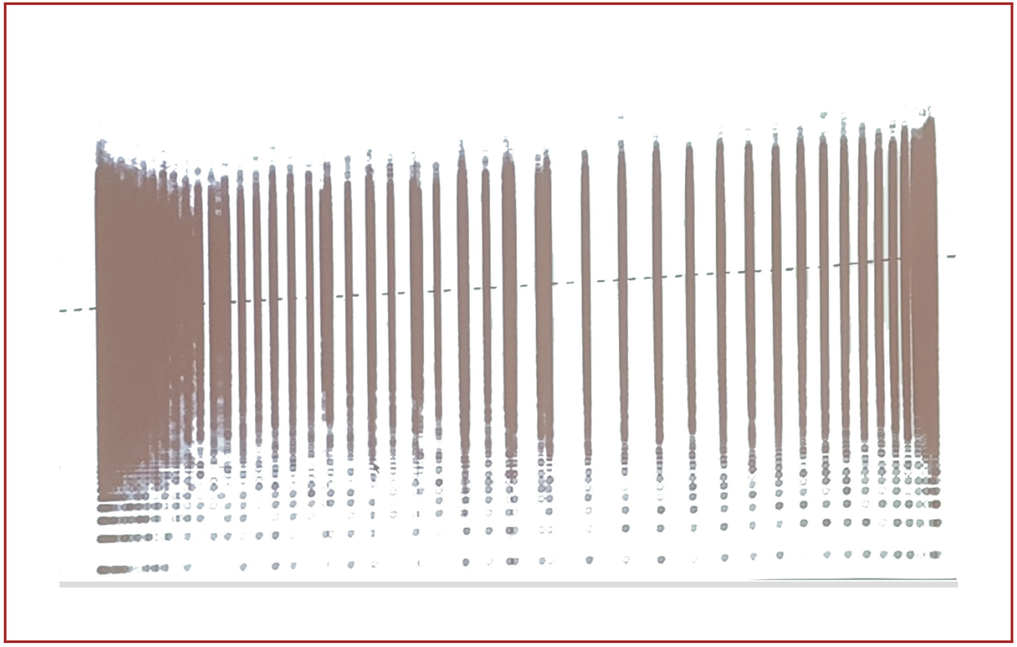
Gradient Descent : $w \leftarrow w - \eta\frac{dL}{dw}$

2 Model Selection and Evaluation

The question arises that *When should we stop fitting?* . Here, in the example where we were trying to model internet viewing using age and gender interactions, we knew that we had to fit a parabola. Here we already had the model and we were figuring out how to select the parameters. We did not discuss how to select the model. We usually find the best parameters and visually inspect the fit by plotting the predicted and the actual values.

We summarized the view as average viewing which is not the same as looking at all the ages. If we look at the genders of all the ages in the figure below, we observe that there exists a lot of individual variability conditioned on age. The variation between the age is more than the systematic individual variance. The average by age changes a tiny bit. Given that the average changed, doesn't mean every person changed. This is why it is not such a good model. We can't interpret the model just by staring at the coefficients. There is a lot that has not been explained. Sometimes, modeling the average is a good idea. If we are making an economic policy, we care about the average more than every individual. Whether we have achieved what we want, depends on the goal. We might want to do analysis **quantitatively** which would actually help us answer if the systematic shift in the mean would explain the variance in the world. Infact, it doesn't and we shall see how simply age and gender does not explain internet viewing!

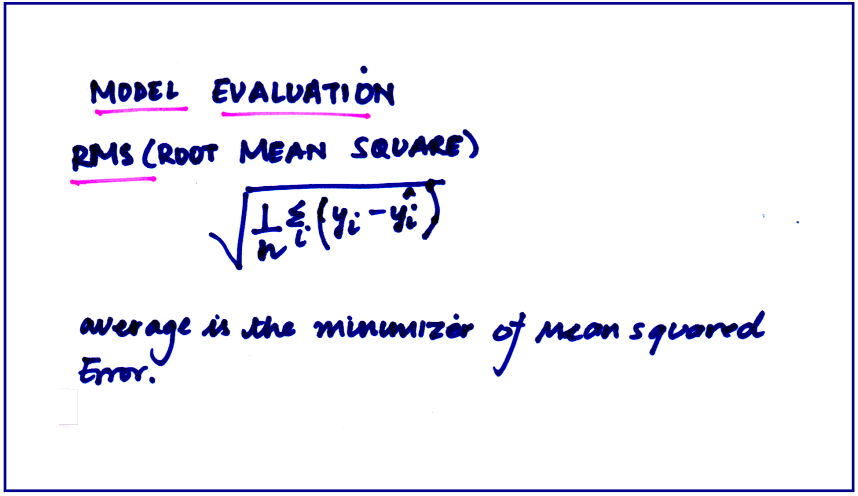
Figure 2: Internet viewing by gender of all ages



3 Performance metrics for model fitting

The most common metric for evaluating a model is *Root mean square error.* The square root in the formula is to enforce the same unit. Saying that we are 10000 square page views off doesn't make much sense.

Figure 3: RMS error



MODEL EVALUATION

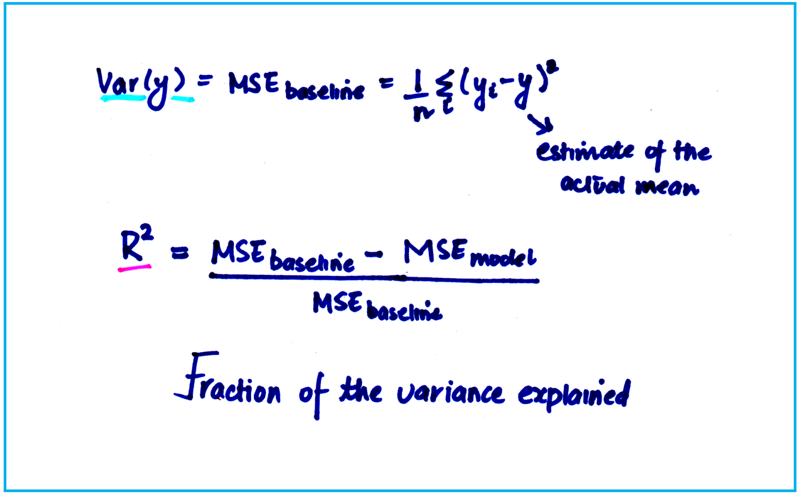
RMS (ROOT MEAN SQUARE)

$$\sqrt{\frac{1}{n}\sum_i (y_i - \hat{y}_i)}$$

average is the minimizer of mean squared Error.

The baseline is to predict the mean for each observation. For any regression problem, we usually measure the performance of the model with respect to the baseline model and ask how does it compare to the average. The following is the representation of the normalized measure we use to evaluate the model. This measure is called R^2.

Figure 4: R^2



$$\text{Var}(y) = \text{MSE}_{\text{baseline}} = \frac{1}{n}\sum_i (y_i - y)^2$$

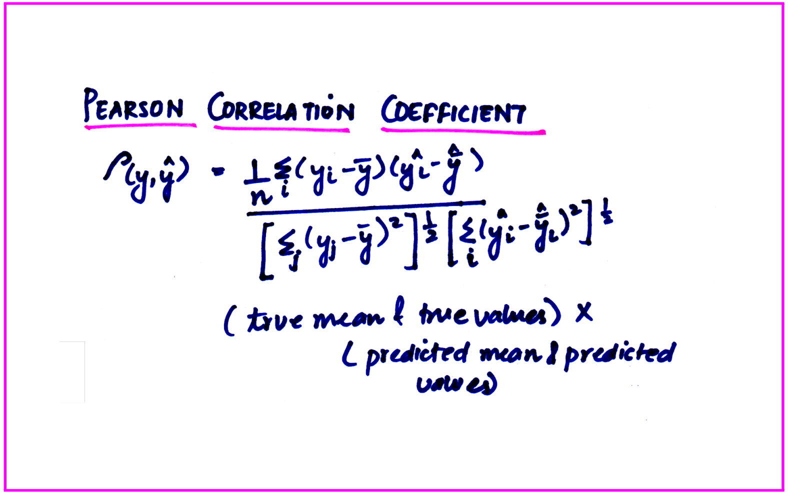
estimate of the actual mean

$$R^2 = \frac{\text{MSE}_{\text{baseline}} - \text{MSE}_{\text{model}}}{\text{MSE}_{\text{baseline}}}$$

Fraction of the variance explained

R^2 explains the fraction of variance in the model. The R value of the above model was 0.019 . It means that this model can capture only 2% of the variance of the whole dataset. Terrible! If R is 0, it is pretty bad while if it is 1, the model is great. This value is not unrelated to correlation that is represented by the Pearson's coefficient. The Pearson's coefficient shows how correlated are the deviations of the true and predicted values are, from the mean. We can use features like glance in R to look at the various coefficients. It is very well known that RMS is extremely sensitive to outliers and that big deviations contribute a lot to mean squared error.

Figure 5: Pearson's Correlation Coefficient



PEARSON CORRELATION COEFFICIENT

$$\rho(y,\hat{y}) = \frac{\frac{1}{n}\sum_i (y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\left[\sum_j (y_j - \bar{y})^2\right]^{\frac{1}{2}}\left[\sum_i (\hat{y}_i - \bar{\hat{y}}_i)^2\right]^{\frac{1}{2}}}$$

(true mean & true values) x (predicted mean & predicted values)

4 The goal is not to get the training error down to 0!

While trying to explain the past in your model, you should be careful of not trying to focus on explaining it too well. The goal is to **generalize** i.e predict well on the unseen data and not try to fit a complex decision boundary that goes through all or most of the training data points. We can use the function $poly(age, k)$ and add more degrees to the decision boundary but this will lead to overfitting. In doing this in *R*, you might get an NA which means that your data points are extremely correlated such that *R* treats them as identical. The thing to remember is that you can do *worse* in the future if pay too much attention to the past.

4.1 Cross Validation

The dataset is usually split into three sets -

1. Training - This data is used to fit the model and find the best set of parameters.
2. Test - This should **not be touched** until we have tested our model on the validation set.
3. Validation - This data is used to test the performance of the training data before it actually exposed to the test set. We should be careful and should not overfit to the validation set.

A famous method for validation is **K-Fold Validation**. Here we select a value of K(generally 5 or 10). A training row is randomly selected and is added to the fold and we go through each fold. For each fold, the other K - 1 folds acts as the training data and the fold selected is treated as the test set. We then select the model with the lowest cross validation error.

5 Bias/Variance Tradeoff

5.1 Bias?

Bias is the true error of the best classifier in the concept class (e.g, best linear separator, best decision tree on a fixed number of nodes). Basically, you approximate a function $\hat{f}$ and you compare how much it is off from the true function *f*.

1. High Bias : If your model has high bias, it means that it assumes something that might not be true, so you could always be off even if you have an infinite amount of training data!
2. Low Bias : If your model has low bias, it means that it can capture complicated patterns with enough data. For example, if the world is really 9th degree, eventually you will find the right model.

5.2 Variance?

Variance is the error of the trained classifier with respect to the best classifier in the concept class. It is a measure of how much your estimate changes across the different datasets($\hat{f}_1$,$\hat{f}_2$,$\hat{f}_3$)

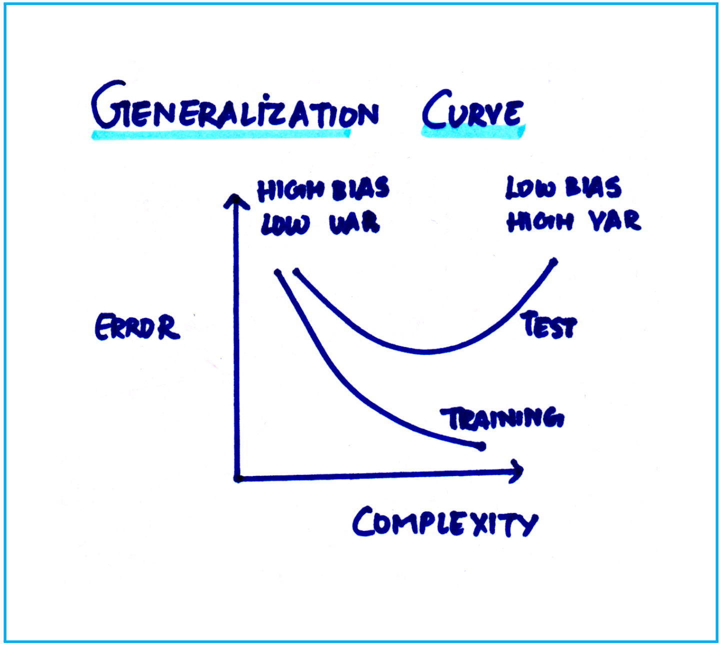
1. High Variance : If your model has high variance, it means your model changes a lot with different training datasets.
2. Low Variance : If your model has low variance, it means that the estimate doesn't change a lot with different datasets.

5.3 Tradeoff?

If we make the concept class more complicated (e.g, linear classification to quadratic classification, or increase number of nodes in the decision tree), then bias decreases but variance increases(Overfitting). On the other hand, if try to lower the variance, we are making the model too simplistic that it cannot capture the true distribution(Underfitting). Thus there is a bias-variance tradeoff.

The equation that captures the mean squared error is **MSE** $=$ **Bias**^{2} $+$ **Variance** - **Irreducible Error**

Figure 6: Generalization Curve

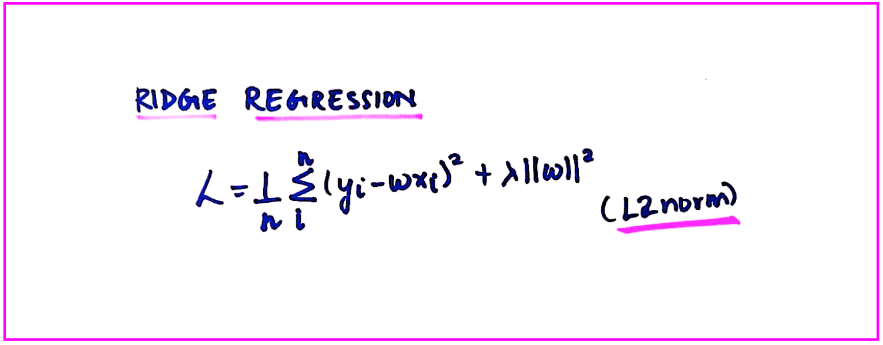


6 Regularization

Regularization artificially discourages complex or extreme explanations of the world even if they fit the what has been observed better. The idea is that such explanations are unlikely to generalize well to the future; they may happen to explain a few data points from the past well, but this may just be because of accidents of the sample. It, technically attempts to solve the overfitting problem in statistical models.

1. Ridge Regression - Ridge regression is a way of shrinking weights. It penalizes the loss function using a parameter λ times the sum of squared weights. It establishes that it doesn't believe in the weights too much and tries to bring them down in an attempt to generalize well.

Figure 7: Ridge Regression

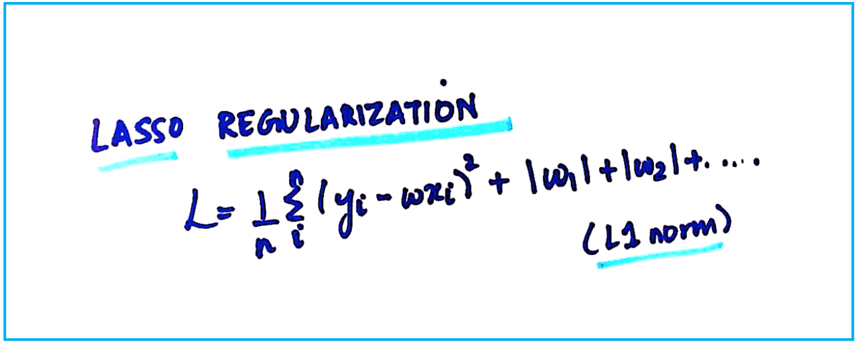


RIDGE REGRESSION

$$L = \frac{1}{n} \sum_{i}^{n} (y_i - w x_i)^2 + \lambda \|w\|^2 \quad \text{(L2norm)}$$

2. Lasso Regression -Lasso is a way to perform feature selection. Here, instead of taking the squared norm,we take the absolute norm also known as $L1$ norm. Lasso ropes in a bunch of things and sets them explicitly to zero. It believes that if it had to choose then it would zero out a coefficient than make it small.

Figure 8: Lasso Regularization



LASSO REGULARIZATION

$$L = \frac{1}{n}\sum_{i}^{n}(y_i - wx_i)^2 + |w_1| + |w_2| + \dots$$

(L1 norm)

Notes from hv2169

1 How to Evaluate Models

We try to find the best parameters by inspecting the fit of the model:

1. Plot the actual and model-predicted values
2. Break this down using a different color for each gender
3. We can also color by age and wrap by gender for clarity

2 Modeling Variability

This is done by summarizing the average of each gender-age combo, which causes us to overlook variability in the observations. Plotting every point shows variability in age is greater than variability across age, revealing issues with the model.

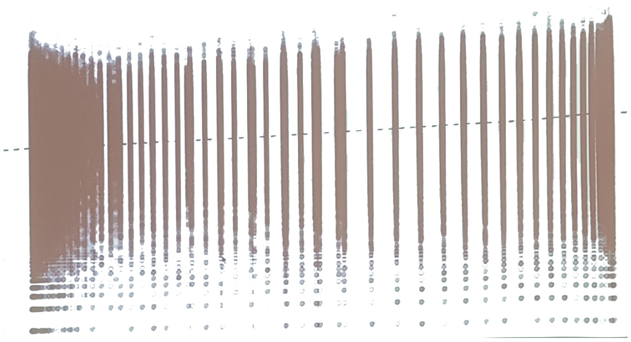


Figure 9: Observations Plot

Traditional methods of reporting statistics by significance overlook this variability as well. We can thus explain overall trends, but not individual observations that may be seen. There may also be unexplored transformations that could reduce variability, or other confouding variables that affect the parameters but have not necessarily been recorded via the model.

3 How Good is a Model?

There are two metrics used to evaluate a model's fit:

- RMS (root-mean-squared) error (equation below)
- Pearson's Correlation Coefficient (equation below)

RMSE $= \sqrt{\frac{1}{N}\sum_i^N (y_i - \hat{y})^2}$
Pearson's r $= \frac{\sum_i (y_i - \bar{y})(\hat{y_i} - \hat{\bar{y}})}{[\sum_i (y_i - \bar{y})^2]^{\frac{1}{2}}\,[\sum_i (\hat{y_i} - \hat{\bar{y}})^2]^{\frac{1}{2}}}$

Note that the square of the Pearson coefficient is the R^2 of the model, the fraction of variance explained.

4 Explaining the Past vs. Predicting the Future

Adding degrees to a polynomial model adds accuracy to its predictive power due to increased flexibility. The R function `poly(x, degree, raw=true)` adds successive polynomial transformations of a variable to a linear model. However, this succeeds mostly in predicting the past but does not necessarily help in effectively predicting the future. Such transformations increase variability in results, and they lead to overfitting of the model. Different observations in the past therefore result in entirely different models. It is important to consider both the effects of overfitting (high variance) and underfitting (high bias) when obtaining a model with predictive power. In other words, the models should be complex enough to explain the past, but simple enough to predict the future.

We achieve this by splitting the data into three parts: training set, validation set, and testing set. We fit the models based on the training set and evaluate the performance using the validation set. Finally, conclusions should be obtained based on performance on the testing set. It is extremely important not to calibrate the model using the testing set (called peeking) as this leads to overfitting of the model.

5 Bias and Variance

Polynomial models have high variance because different observation sets lead to distinct prediction models. However, there is a tradeoff between variance and bias. The following equation captures the relationship between the two.

$$MSE = Bias^2 + Variance + IrreducibleError$$

Although a liear model has low variance, it is likely to have high bias since the actual model itself may not be linear. On the other hand, an n-degree polynomial model is likely low bias. A generalization curve evaluates model complexity based on predictive power on training and test sets. This can be used to find a point of divergence between two lines to find a middle ground between bias and variance.

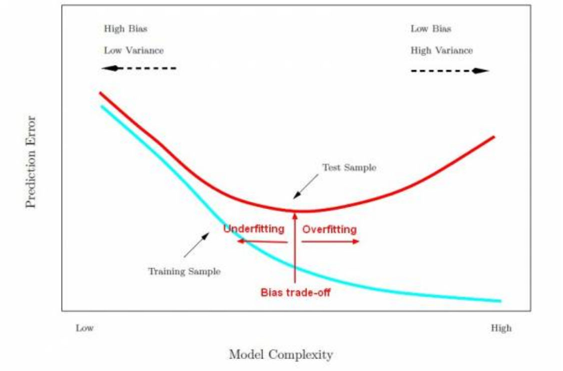


Figure 10: Generalization Curve

6 Choosing the Training and Validation Sets

In R the function `sample()` can be used to select indices of observations we want to include in either set. It is important to use random sampling to avoid orderings inherent to the dataset, thus avoiding trends that affect the bias of the models. It is also important, however, to use as much of the dataset as possible in order to arrive at a model with the greatest predictive power. One solution is to use K-fold cross validation, which involves using different subsets of the training and validation data so that each observation is part of the validation at least one. Then, average across all the runs to find the best predictive model.

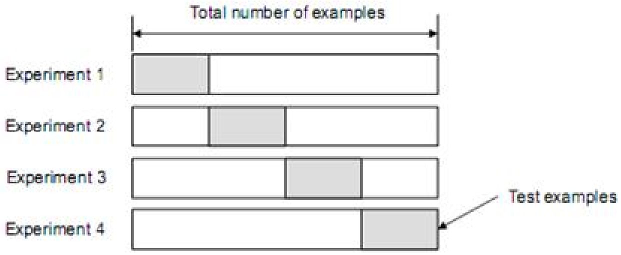


Figure 11: K-Folds Cross Validation with K = 4

7 Regularization

Regularization penalizes complexity in the loss function for a specific choice to map it in the model. Do so by using Ridge Regularization or Lasso Regularization.

Ridge Regularization

$$\mathcal{L} = \frac{1}{n}\sum_{i}^{n}(y_i - wx_i)^2 + \lambda||w||^2$$

As λ increases, the coefficient decrease. This reduces the weighting of the model fit.

Lasso Regularization

$$\mathcal{L} = \frac{1}{n}\sum_{i}^{n}(y_i - wx_i)^2 + ||w_1|| + ||w_2|| + ...$$

This performs variable selection by ignoring coefficients that are near zero.

Notes from ih2309

1 From the Previous Lecture

We have learned the general mathematical formulas and reasoning behind why we use linear regression for the presentation of the data. In many cases, linear regression can be a great tool in presenting the general trend of the data. Using the same data we used at the end of the last lecture, we can see that each dot on the graph is the average for a certain age group. By looking at these data points and the general trend curve, we can make a prediction(which in fact is our end goal in finding the "best" model). We can also manipulate data so that we can select specific data based on different category. After plotting predicted and actual data, we can use different colors to plot the trend curves by gender and age.

2 Evaluating the fit of model

One of the key information in the data we miss out with this trend curve is variation in the data. It might be the case that two different sets of data with almost identical trend curve have drastically different data. This can be due to the difference in the variability, which has the real world implication. So we need to take standard deviation into consideration when we evaluate the fit of model. There are two commonly used formula: root mean square error and Pearson's correlation coefficient.

- Root mean square error (RMSE)

$$\sqrt{\frac{1}{N}\sum_{i}^{N}(y_i - \hat{y})^2}$$

  The RMSD represents the sample standard deviation of the differences between predicted values and observed value.

- Pearson's correlation coefficient (PCC)

$$\frac{\sum_i (y_i - \bar{y})(\hat{y}_i - \hat{\bar{y}})}{[\sum_i (y_i - \bar{y})^2]^{\frac{1}{2}}[\sum_i (\hat{y}_i - \hat{\bar{y}})^2]^{\frac{1}{2}}}$$

  The PCC is a measure of the linear dependence (correlation) between two variables X and Y. In a nutshell, PCC is the covariance of the two variables divided by the product of their standard deviations.

3 Trade-off between Accuracy and Prediction

When fitting a curve to the data, we often have to deal with a question of how much accurate do we want the function to be. While making the curve fit very accurately to the data can look good and seem to be a good representation, a too accurate model can perform badly in predicting future since it is overly sensitive to noise, which is the definition of "overfitting." On the other hand, we would not want our curve to be too loose to the extent that the given data does little in predicting future, "underfitting". This complexity is represented by the order of a function – the higher it is, the more complex the model gets.

As we can see from the above graph, the optimal point of accuracy happens somewhere in-between overfitting and underfitting.

4 Bias- and Variance Trade-off

$$MSE = Bias^2 + Variance + IrreducibleError$$

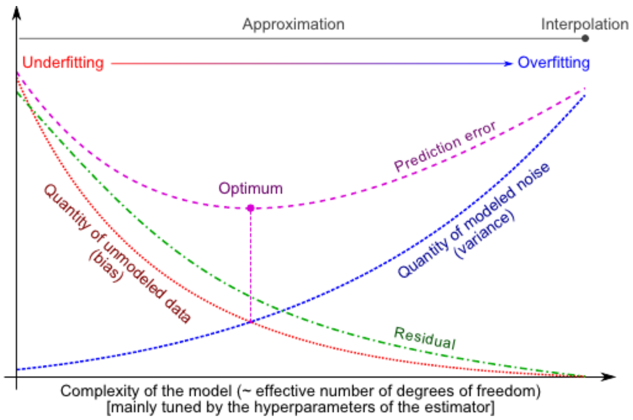


Figure 12: source: http://www.brnt.eu/phd/node14.html

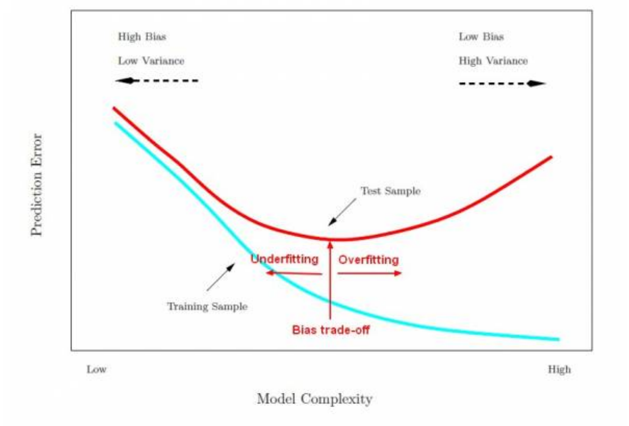


Figure 13: source: http://gerardnico.com/wiki/

Low variance: It does not change a lot as it gets difficult.
High variance: Model changes a lot with different draws of the data.
High Bias: Assumes something that might not be true, so it could always be wrong even with infinite amount of data
Low Bias: Can capture complicated patterns with enough data.

5 Setting complexity

So now we know choosing the right complexity, i.e. degree of functions is important. To evaluate this, we fit the models on the "training set", and test with "validation set". These two sets should be randomly chosen so that we can test the model.
More generally, to find a good model

- randomly split our data into 3 sets
- fit models on the training set
- Use the validation set to find the best model (different data)
- Quote final performance of this model on the test set

One technique we can make a good use of is "K-fold cross validation." We take a subset of data as a validation data out of training data, and run the test. Then we take another subset out of the training data as a new validation data. Do this until every data has been used as a training data once. Finally, average all runs. Beware that we need to make sure the data is randomized. Then we can generate many test cases based on the real examples. We choose the order of the function that is minimum. It does not have to be strictly the minimum, however, as it depends on each test. Rather, find the lowest order that is among the flat low line by looking at it.

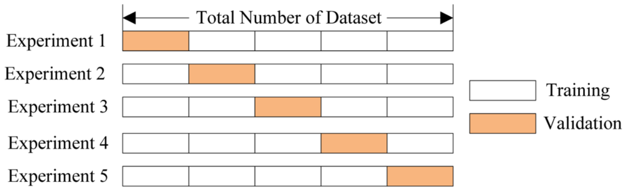


Figure 14: K-fold Cross Validation (source: http://sdsawtelle.github.io/blog/output/week6-andrew-ng-machine-learning-with-python.html)

Another technique is adding a random noise to the existing data and use that as training/validation data. The noise-generating technique is more advance and was briefly discussed during the lecture.

6 Regularization

Lastly, we discussed on ways to penalize complexity of the function.

- Ridge Regularization:

$$\mathcal{L} = \frac{1}{n}\sum_{i}^{n}(y_i - wx_i)^2 + \lambda||w||^2$$

  As λ increases, the coefficients decrease and reduces the weight of the model fit.

- Lasso Regularization:

$$\mathcal{L} = \frac{1}{n}\sum_{i}^{n}(y_i - wx_i)^2 + ||w_1|| + ||w_2|| + ...$$

  We can ignore near-zero coefficients.

Notes from yn2277

1 How to Evaluate Models

Currently, we try to find the best parameters by visually inspecting the fit:

1. We start by plotting the *predicted* and *actual* values according to the model
2. We can break this down further using a different *color* for each gender
3. We can color by age, and *wrap* by gender to make this clearer

Note: *Discrete variables* (such as gender) are automatically colored with a discrete color scale, while *continuous variables* (such as age) are automatically colored with a continuous color scale.

2 Modeling Variability

However, all this is done by summarizing the *average* of each gender/age combination, which might cause us to overlook some variability in our observations. For instance, plotting every point shows that variability *within* age is higher than variability *across* age in the figure below), which reveals the deficiencies of our model.

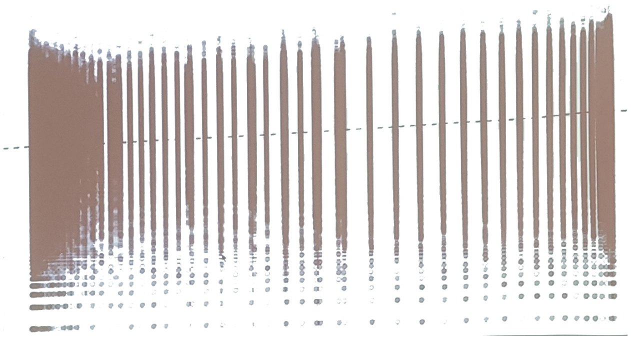


Figure 15: Complete Plot of Observations

Source: Class notes, Jake Hofman

Traditional methods of reporting statistics (by significance value) also overlook this variability, because while it is true that you have explained average changes by age, it is also true that you have *not* explained the rest of the variability in the model. In other words, we can explain *collective* trends, but not *individual* observations that we might see in the real world.

There might also be *unexplored transformations* to the data that might reduce this variability, or *other variables* such as internet connection that have not been recorded.

3 How Good is a Model?

There are two main metrics we can use to evaluate the fit of a model:

- Root mean squared error (RMSE)

$$\sqrt{\frac{1}{N}\sum_{i}^{N}(y_i - \hat{y})^2}$$

  - R syntax: `sqrt(mean((pred - actual)^2)))`
  - We take the square root because we are calculating the squared difference between predicted and observed values.
  - To obtain a baseline MSE, we calculate the difference between the each observation and the mean of the observed data (i.e. the variance of y).
  - We then compare this to the MSE of the model with

$$\frac{MSE_{baseline} - MSE_{model}}{MSE_{baseline}} = R^2$$

    where R^2 is also the fraction of variance explained.

- Pearson's correlation coefficient

$$\frac{\sum_i (y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{[\sum_i (y_i - \bar{y})^2]^{\frac{1}{2}} [\sum_i (\hat{y}_i - \bar{\hat{y}})^2]^{\frac{1}{2}}}$$

  - R syntax: `cor(pred, actual)`
  - We calculate the deviations from the mean of observed and actual data, and then scale them accordingly by the denominator
  - The square of Pearson's correlation coefficient is also equal to to R^2 in the MSE, or the fraction of variance explained.

Applying the above tools to our previous model, we get a low R^2 of 0.019, which shows that our model in fact explains *very little* of the data. However, we need to remember that MSE is also highly susceptible to outliers, because a big difference between predicted and actual outcomes may skew the results.

4 Explaining the Past vs. Predicting the Future

As we add degrees to a polynomial, we add accuracy to the predictive power of a model because of the increased flexibility. The `poly(x, degree, raw=true)` function in R automatically adds successive polynomial transformation (until a user-defined k degree) of a variable to a linear model. We need to remember to remove the original variable, because this transformation will also give results for `k=1`.

However, while this succeeds in predicting the *past*, it does not necessarily have a bearing on how accurately we predict the *future*. Polynomial transformations also increase variability in our results, such that different observation sets might lead us to drastically different conclusions (see *Bias and Variance*).

Our models should thus be *complex* enough to explain the past, but *simple* enough to generalize the future. We can achieve this by randomly splitting our data into three sets: training set, validation set, and testing set.

1. Fit the models based on the *training set*
2. Use the *validation set* to evaluate their performance
3. Quote final performance on the *testing set*

If the model does not perform well on the testing set, we should not amend it using the same dataset, but wait for new data to come in. Unfortunately, this is often ignored in practice. A possible solution is a *reusable holdout set* which adds noise to the results of the testing set, rather than revealing the testing data directly.

5 Bias and Variance

As discussed earlier, polynomial models have *high variance* because different observation sets can lead to drastically different predictions. However, there is often a tradeoff between variance (predictiveness of model with *different* observations) and bias (predictiveness of model with *more* observations).

Loosely speaking, the relationship between variance and bias can be captured in the following equation, such that for a given MSE (and irreducible error), an increase in bias will result in a decrease in variance (or vice versa).

$$MSE = Bias^2 + Variance + IrreducibleError$$

While a linear model is *low variance*, it is likely to be *high bias* because there are no guarantees that the relationship is indeed linear. In other words, even with infinite observations, a model may still be wrong. By contrast, a n-degree polynomial model is likely to be *low bias*, because with infinite observations we are more likely to arrive (trivially) at an accurate representation of the world.

A *generalization curve* evaluates the complexity of the model based on its predictive power on the training sample and test sample. Using this, we can identify a point of divergence between the two lines to find a middle ground between bias and variance.

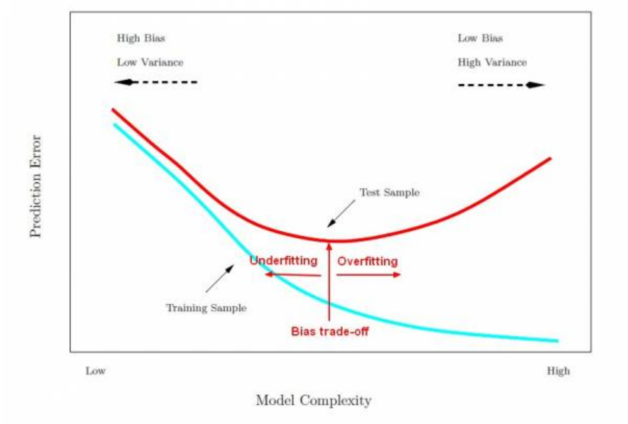


Figure 16: Generalization Curve for Training and Testing/Validation Errors

Source: http://gerardnico.com/wiki/_media/data_mining/model_complexity_error_training_test.jpg

6 Choosing the Training and Validation Sets

When shuffling the data, we can use the `sample()` function in R to select the indices of observations we want to include in either the training or test sets. However, we need to be careful of inherent orderings in the data, which might contain trends (e.g. seasonality) that would be incorrectly eliminated by random sampling.

We notice, however, that our results can vary significantly based on our assignment of the training and validation sets, which brings us to *K-fold cross validation*, which involves taking different subsets of training and validation data such that every observation is at one point part of the validation set. We can then average across all the runs to find the best possible model.

To process this in R, we simply create a variable called `num_folds` and call `sample(1:num_folds)` to split the folds evenly between the observations. We also keep track of both the average test error and standard error across all the folds, to help us evaluate the significance of any reduction in observed error.

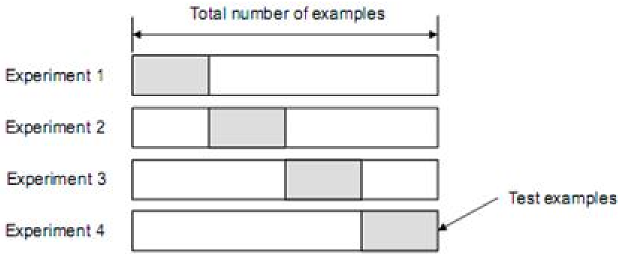


Figure 17: 4-Folds Cross Validation with Training and Testing/Validation Sets

Source: http://www.imtech.res.in/raghava/gpsr/Evaluation_Bioinformatics_Methods_files/image002.jpg

7 Regularization

Regularization involves penalizing complexity in our loss function for a specific choice to map it in the model. We can do so using either Ridge Regularization or Lasso Regularization.

- Ridge Regularization

$$\mathcal{L} = \frac{1}{n}\sum_{i}^{n}(y_i - wx_i)^2 + \lambda||w||^2$$

  As λ increases, the coefficients decrease which reduces the weighting of the model fit.

- Lasso Regularization

$$\mathcal{L} = \frac{1}{n}\sum_{i}^{n}(y_i - wx_i)^2 + ||w_1|| + ||w_2|| + ...$$

  This effectively performs variable selection by ignoring near-zero coefficients.