Regularization Comparison
Review, 3/31 - 4/5

- Confidence intervals
- Bootstrap

- Prediction Framework: Train, Development, Test
- Overfitting: Bias versus Variance
- Feature Selection: Forward Stepwise Regression
- Ridge Regression (L2 regularization)
- Lasso Regression (L1 regularization)
Common Goal: Generalize to new data

- Training Data
- Development
- Testing Data

Model

Does the model hold up?

Set parameters
N-Fold Cross-Validation

Goal: Decent estimate of model accuracy

Iter 1
train | dev | test

Iter 2
train | dev | test | train

Iter 3
train | dev | test | train

... ...
Supervised vs. Unsupervised

**Supervised**

- Predicting an outcome $E(y|X)$
- Loss function used to characterize quality of prediction $L(y, \hat{y}) = (y - \hat{y})^2$
Supervised vs. Unsupervised

Supervised

- Predicting an outcome $E(y|X)$
- Loss function used to characterize quality of prediction $L(y, \hat{y}) = (y - \hat{y})^2$

Unsupervised

- No outcome to predict
- Goal: Infer properties of $P(X)$ without a supervised loss function.
- Often larger data.
- Don’t need to worry about conditioning on another variable.
K-Means Clustering

**Clustering**: Group similar observations, often over unlabeled data.

**K-means**: A “prototype” method (i.e. not based on an algebraic model).

Euclidean Distance:

\[
d(x_i, x_{i'}) = \sqrt{\sum_{j=1}^{m} (x_{ij} - x_{ij'})^2} = ||x_i - x_{i'}||
\]

centers = a random selection of k cluster centers
until centers converge:

1. For all \( x_i \), find the closest center (according to \( d \))
2. Recalculate centers based on mean of euclidean distance
Review 4-7

- Cross-validation
- Supervised Learning
- Euclidean distance in m-dimensional space
- K-Means clustering
K-Means Clustering

Understanding K-Means

(source: Scikit-Learn)
Dimensionality Reduction - Concept
Dimensionality Reduction - PCA

Linear approximates of data in $q$ dimensions.

Found via *Singular Value Decomposition*: $X = UDV^T$
Review 4-11

- K-Means Issues
- Dimensionality Reduction
- PCA
  - What is V (the components)?
  - Percentage variance explained
scikit-learn algorithm cheat-sheet

classification
- SVC
- Ensemble Classifiers
- KNeighbors Classifier
- SGD Classifier
- Naive Bayes
- Text Data
- Linear SVC

<100K samples

>50 samples

regression
- SGD Regressor
- Lasso
- ElasticNet
- Ridge Regression (SVR kernel='linear')

<100K samples

few features should be important

number of categories known

<10K samples

<10K samples

dimensionality reduction
- Randomized PCA
- Isomap
- Spectral Embedding
- LLE

<10K samples

predicting a category

predicting a quantity

predicting a structure

predicting a quantity

more data

labeled data

<10K samples

<10K samples

tough luck

<10K samples

Back
Classification: Regularized Logistic Regression

\[ \lambda \| \beta \|_2^2 \quad \lambda \| \beta \|_1 \]
Classification: Naive Bayes

**Bayes classifier:** choose the class most likely according to $P(y|X)$.
(y is a class label)
Classification: Naive Bayes

Bayes classifier: choose the class most likely according to $P(y|X)$. (y is a class label)

Naive Bayes classifier: Assumes all predictors are independent given y.

\[ P(Y = y | A = a, B = b, C = c) = p(y|a)p(y|b)p(y|c) \]

\[ P(y|X) = \prod_{i=1}^{m} P(y|X_i) \]
Classification: Naive Bayes

\[ P(y|X) = \frac{P(y)P(X|y)}{P(X)} \]

Bayes Rule:

\[ P(A|B) = \frac{P(B|A)P(A)}{P(B)} \]

\[ P(y|X) = \prod_{i=1}^{m} P(y|X_i) \]
Classification: Naive Bayes

\[
P(y|X) = \frac{P(y)P(X|y)}{P(X)}
\]
Classification: Naive Bayes

\[
P(y|X) \propto P(y, X_1, \ldots, X_m) \propto P(y) \prod_{i=1}^{m} P(X_i|y)
\]

**Maximum a Posteriori (MAP):** Pick the class with the maximum posterior probability.

\[
\hat{y} = \arg \max_y P(y) \prod_{i=1}^{m} P(X_i|y)
\]
Classification: Naive Bayes

\[ P(y|X) = \frac{P(y)P(X|y)}{P(X)} \]

**Prior**

\[ P(y|X) \propto P(y, X_1, \ldots, X_m) \propto P(y) \prod_{i=1}^{m} P(X_i|y) \]

**Likelihood**

**Maximum a Posteriori (MAP):** Pick the class with the maximum posterior probability.

\[ \hat{y} = \operatorname{arg\ max}_y P(y) \prod_{i=1}^{m} P(X_i|y) \]

**Unnormalized Posterior**