Section 3: Midterm review

STATS 202: Data Mining and Analysis

Linh Tran

tranlm@stanford.edu

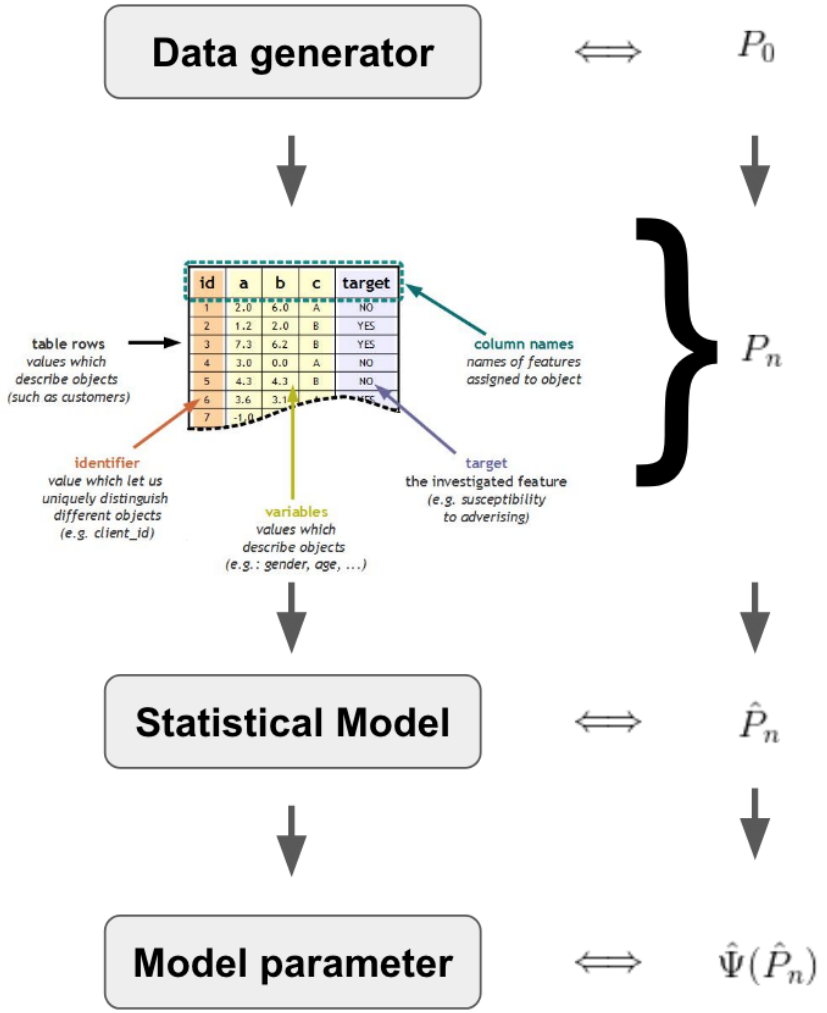
Department of Statistics
Stanford University

July 16, 2021

Announcements

- Homework 2 due today (via Gradescope)

Empirical vs true distributions



Ideally, we want $\Psi(P_0)$.

Supervised learning

Motivation: Why learn f_0?

Prediction

- Useful when we can readily get $X_1, X_2, ..., X_p$, but not Y.
- Allows us to predict what Y likely is.
- Example: Predict stock prices next month using data from last year.

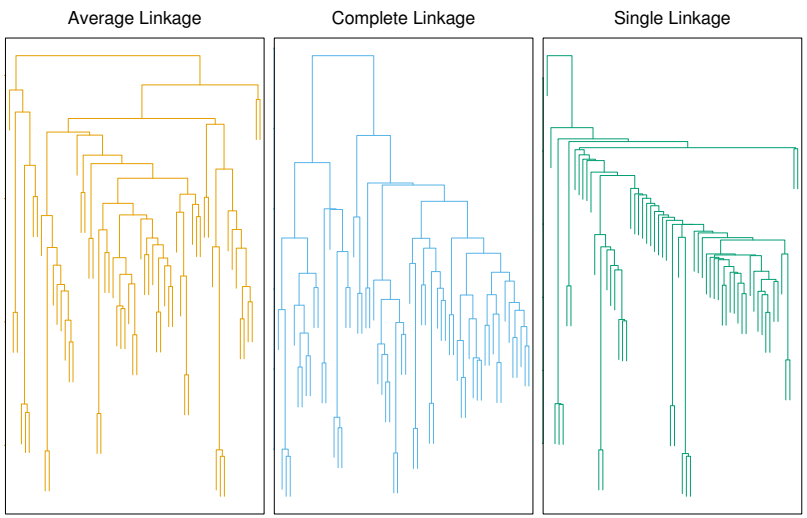
Inference

- Allows us to understand how differences in $X_1, X_2, ..., X_p$ might affect Y.
- Example: What is the influence of genetic variations on the incidence of heart disease.

Unsupervised learning

- In unsupervised learning, all the variables are on equal standing, no such thing as an input and response.
- Clustering is typically applied
  - Hierarchical clustering (single, complete, or average linkage).
  - K-means clustering.
  - Expectation maximization (using Gaussian mixtures).

Hierarchical clustering



- Agglomerative algorithm produces a *dendrogram*.
- At each step we join the two clusters that are "closest":
  - **Complete:** distance between clusters is maximal distance between any pair of points.
  - **Single:** distance between clusters is minimal distance.
  - **Average:** distance between clusters is the average distance.
- Height of a branching point = distance between clusters joined.

K-means clustering

- The number of clusters is fixed at K.
- Goal is to minimize the average distance of a point to the average of its cluster.
- The algorithm starts from some assignment, and is guaranteed to decrease this average distance.
- This find a local minimum, not necessarily a global minimum, so we typically repeat the algorithm from many different random starting points.

Supervised learning

We're interested in a response variable Y associated to each vector of predictors $\mathbf{X}$.

Regression: $f_0 = \mathbb{E}_0[Y|X_1, X_2, ..., X_p]$

- A scalar value, i.e. $f_0 \in \mathbb{R}$
- $\hat{f}_n$ therefore gives us estimates of y

Classification: $f_0 = \mathbb{P}_0[Y = y|X_1, X_2, ..., X_p]$

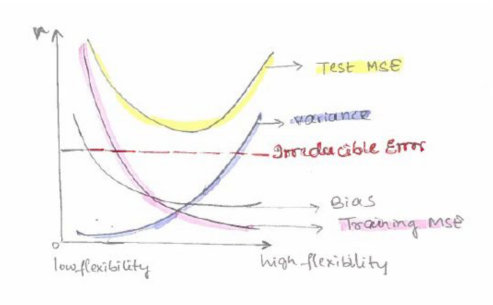
- A vectored value, i.e. $f_0 = [p_1, p_2, ..., p_K] : p_j \in [0, 1], \sum_K p_j = 1$
- n.b. In a binary setting this simplies to a scalar, i.e. $f_0 = p_1 : p_1 = \mathbb{P}_0[Y = 1|X_1, X_2, ..., X_p] \in [0, 1]$
- $\hat{f}_n$ therefore gives us predictions of each class
- Can take the arg max, giving us Bayes Classifier

Bias variance decomposition

Let x_0 be a fixed point, $y_0 = f_0(x_0) + \epsilon$, and $\hat{f}_n$ be an estimate of f_0 from $(x_i, y_i) : i = 1, 2, ..., n$.

The MSE at x_0 can be decomposed as

$$\begin{aligned} MSE(x_0) &= \mathbb{E}_0[y_0 - \hat{f}_n(x_0)]^2 && (1) \\ &= \text{Var}(\hat{f}_n(x_0)) + \text{Bias}(\hat{f}_n(x_0))^2 + \text{Var}(\epsilon_0) && (2) \end{aligned}$$



Loss functions

Regression:

- MSE ($(y_i - \hat{y}_i)^2$)
- AIC, BIC, R^2, Adjusted R^2

Classification:

- Cross-entropy ($(y_i \log(\hat{p}_i))$
- 0-1 loss ($\mathbb{I}(y_i \neq \hat{y}_i)$)
- Confusion matrix
- Receiver operating characteristic curve (& AUC)

Linear models

- Coefficients, standard errors, and hypothesis testing
- Interactions between predictors
- Non-linear relationships
- Correlation of error terms
- Non-constant variance of error (heteroskedasticity)
- Outliers
- High leverage points
- Collinearity
- Mis-specification

Regression methods

- Multiple linear regression
- Stepwise selection methods (e.g. forward, backward, etc.)
- Ridge regression, Lasso, and elastic net
- Nearest neighbors regression

Comparing Linear Regression to K-nearest neighbors

Linear regression: prototypical parametric method
KNN regression: prototypical nonparametric method Long story short:

- KNN is only better when the function f_0 is not linear (and plenty of data)
  - **Question**: What if the true function f_0 IS linear?

Comparing Linear Regression to K-nearest neighbors

Linear regression: prototypical parametric method
KNN regression: prototypical nonparametric method Long story short:

- KNN is only better when the function f_0 is not linear (and plenty of data)
  - **Question**: What if the true function f_0 IS linear?
- When n is not much larger than p, even if f_0 is nonlinear, linear regression can outperform KNN.

Comparing Linear Regression to K-nearest neighbors

Linear regression: prototypical parametric method
KNN regression: prototypical nonparametric method Long story short:

- KNN is only better when the function f_0 is not linear (and plenty of data)
  - **Question**: What if the true function f_0 IS linear?
- When n is not much larger than p, even if f_0 is nonlinear, linear regression can outperform KNN.
- KNN has smaller bias, but this comes at a price of (much) higher variance (c.f. overfitting)

Classification methods

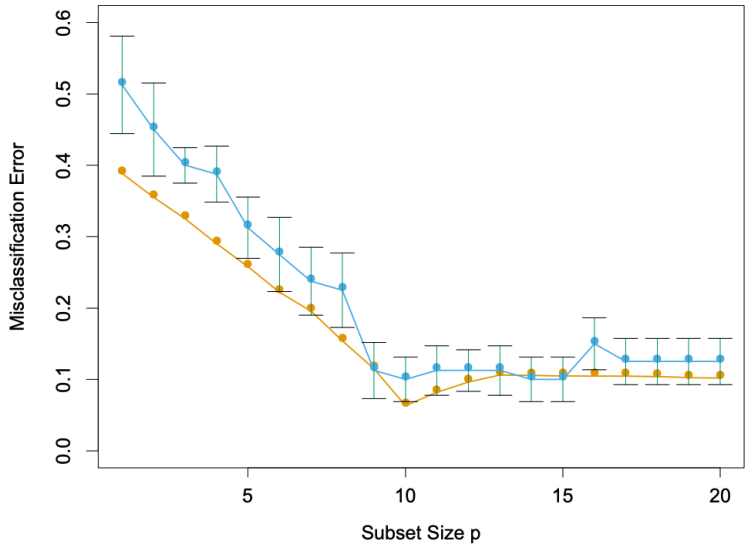
- Nearest neighbors classification
- Naive Bayes
- Logistic regression
- LDA and QDA
- Stepwise selection methods

How do we estimate the test error?

- Our main technique is to split the data.
- Different approaches:
  1. **Validation set:** Split the data in two parts, train the model on one subset, and compute the test error on the other.
  2. k-**fold:** Split the data into k subsets. Average the test errors computed using each subset as a validation set.
  3. **LOOCV:** k-fold cross validation with $k = n$.
- No approach is superior to all others.
- What are the main differences? How do the bias and variance of the test error estimates compare? Which methods depend on the random seed?

How do we choose a model?

Forward stepwise selection



Blue: 10-fold cross validation
Yellow: True test error

- A number of models with $9 \leq p \leq 15$ have the same CV error.
- The vertical bars represent 1 standard error in the test error from the 10 folds.
- **Rule of thumb:** Choose the simplest model whose CV error is no more than one standard error above the model with the lowest CV error.

The Bootstrap

- **Main idea:** If we have enough data, the empirical distribution is similar to the actual distribution of the data.
- Resampling with replacement allows us to obtain pseudo-independent datasets.
- They can be used to:
  1. Approximate the standard error of a parameter (say, β in linear regression), which is just the standard deviation of the estimate when we repeat the procedure with many independent training sets.
  2. Compute confidence intervals (e.g. normal-based, quantile, etc.).
  3. Estimate out of bag error.
  4. Estimate bias.
  5. **Bagging**: By averaging the *predictions* $\hat{y}$ made with many independent data sets, we eliminate the variance of the

The Jackknife

If $d > 1$:

$$\widehat{SE}_B(\hat{\alpha}_n) = \sqrt{\frac{n-d}{d\binom{n}{d}} \sum_z \left(\hat{\alpha}_n^{*,z} - \frac{1}{\binom{n}{d}} \sum_{z'} \hat{\alpha}_n^{*,z'} \right)^2} \tag{3}$$

When $d = 1$, this simplifies to:

$$\widehat{SE}_B(\hat{\alpha}_n) = \sqrt{\frac{n-1}{n} \sum_{i=1}^{n} \left(\hat{\alpha}_n^{*,i} - \frac{1}{n} \sum_{i'=1}^{n} \hat{\alpha}_n^{*,i'} \right)^2} \tag{4}$$

- Is a linear approximation to the bootstrap (though asymptotically equivalent)
- Can be less computationally expensive; esp for large data sets
- Doesn't work well for sample quantiles like the median

Self testing questions

For each of the regression and classification methods:

1. What are we trying to optimize?
2. What does the fitting algorithm consist of, roughly?
3. What are the tuning parameters, if any?
4. How is the method related to other methods, mathematically and in terms of bias, variance?
5. How does rescaling or transforming the variables affect the method?
6. In what situations does this method work well? What are its limitations?