STA 35C: Statistical Data Science III

Lecture 26: Conclusion

Dogyoon Song

Spring 2025, UC Davis

Announcement

Final exam on Fri, June 6 (1:00 pm–3:00 pm) in Wellman Hall 26 (=classroom)

- **Instructions:**
  - **Arrive on time**: The exam starts at 1:00 pm and ends at 3:00 pm sharp
  - **Up to three hand-written cheat sheets:** Letter-size (8.5"×11"), double-sided
  - **Calculator**: A simple (non-graphing) scientific calculator is allowed
  - **No other materials:** No textbooks, notes, etc., beyond the cheat sheets
  - **SDC accommodations:** Confirm your schedule with AES *ASAP*
- **Preparation:**
  - *Cumulative* coverage: Lectures 1–25
  - A practice final and brief answer key are available on the course webpage; previous midterms (+solution) and homework are also available
  - Discussion section materials and homework solution are on Canvas
  - Office hours:
    - *Instructor:* Wed, June 4 (4:00–6:00pm, extended)
    - *TA:* Thu, June 5, 1–2pm

Course evaluation: Please share your feedback comments by Thu, June 5

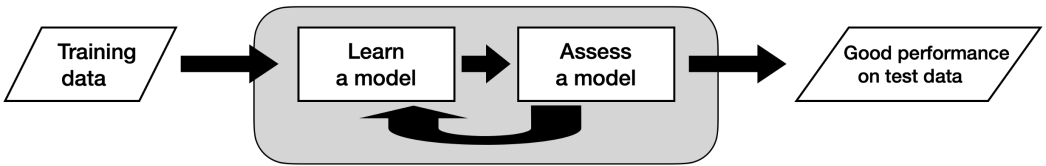
Today's topics

Review of key topics:

- Statistical learning
- Regression
- Classification
- Model assessment & selection:
  - Cross-validation
  - Bootstrap
  - Subset selection
  - Regularization
- Unsupervised learning
  - Principal component analysis
  - Clustering

Also,see mid-course review (Lecture 12 & a part of Lecture 13)

Statistical learning and STA 35C



- **Core idea**: Learn a model from training data, evaluate its performance, and refine it
  - Aim for good predictions or insights on ***new, unseen*** **data**
  - Rely on probability and statistical principles to measure uncertainty and avoid overfitting
- Learning objectives in **STA 35C**:
  - When and how to use different supervised or unsupervised learning methods
  - How to assess and interpret models (cross-validation, bootstrap, model selection)
  - Our focus is on **first principles**, rather than advanced machine learning techniques

Supervised vs. unsupervised learning

Supervised learning

$$\underbrace{X}_{\text{predictors}} \rightarrow \underbrace{Y}_{\text{response}}$$

- **Goal:** Estimate $f : X \rightarrow Y$ so that $y \approx f(x)$
- **Why?**
  - *Prediction:* e.g., forecasting sales, predicting house prices
  - *Inference:* identifying significant predictors, relationships among variables
- **Depending on the type of Y,**
  - *Regression*: Y is numeric
  - *Classification*: Y is categorical

Unsupervised learning: Learn structure in X (no Y)

- *Dimension reduction*: Extract a small subset or combine features for compression
- *Clustering*: Cluster customers by purchasing behavior

Regression: Basics

Problem setup

$$\underbrace{X}_{\text{predictors}} \longrightarrow \underbrace{Y}_{\text{numeric}} \in \mathbb{R}$$

Goal: Estimate $f : X \to Y$ to fit a regression line (or curve)

If we knew the distribution of (X, Y)...

- We might use $\hat{Y} = \mathbb{E}[Y \mid X]$
- In reality, we only have finite data, so we estimate from samples

Parameter estimation: Find β_0, β_1 that minimize

$$\mathrm{RSS} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \quad \text{where} \quad \hat{y}_i = \beta_0 + \beta_1 x_i$$

Regression: Key points

Prediction: $\hat{y}_{\text{new}} = \hat{\beta}_0 + \hat{\beta}_1 x_{\text{new}}$

- Individual outcomes may vary (noise)

Model fit:

- $R^2 = 1 - \frac{\text{RSS}}{\text{TSS}} \in [0, 1]$: proportion of variance in Y explained by the model
- Higher R^2 indicates better explanatory power
- Adding more predictors always increases R^2; R^2_{adj} penalizes for extra variables

Regression coefficient:

- Interpretation:
  - β_1: On average, Y changes by β_1 per unit increase in X
  - In multiple regression, β_1 is the effect of X_1 holding X_2 fixed (conditional effect)
- Significance test:
  - Null hypothesis $H_0 : \beta_1 = 0$ (no linear relationship)
  - If $t = \frac{\hat{\beta}_1}{\text{SE}(\hat{\beta}_1)}$ is large in magnitude, we reject H_0 and conclude significance
- Depending on the model, we may observe confounding

Classification: Basics

Problem setup

$$\underbrace{X}_{\text{predictors}} \longrightarrow \underbrace{Y}_{\text{classes}} \in \{0, 1\}$$

Goal: Estimate f to define a decision boundary between classes

Key ideas:

- If we knew $\Pr[Y = 1 \mid X]$, we could classify $Y = 1$ if $\Pr[Y = 1 \mid X] \geq p^*$
  - Decision threshold p^* matters!
- In reality, we need to estimate $\Pr[Y = 1 \mid X]$ from data, and use it
- Two approaches:
  - *Discriminative* approach: directly model $\Pr[Y = 1 \mid X]$
  - *Generative* approach: model $\Pr[X \mid Y]$, then use Bayes' theorem

Classification: Discriminative vs. generative approaches

Logistic regression is a discriminative approach:

$$\log\left(\frac{\Pr[Y=1|X]}{\Pr[Y=0|X]}\right) = \beta_0 + \beta_1 X$$

- Similar to linear regression, but the response is the log-odds of $Y = 1$
- Estimate the parameters by maximum likelihood estimation
- Prediction with a fitted model:
  - Calculate $\hat{p}_{\text{new}} = \sigma(\hat{\beta}_0 + \hat{\beta}_1 x_{\text{new}})$, where $\sigma(z) = \frac{1}{1+e^{-z}}$
  - Predict $Y = 1$ if $\hat{p}_{\text{new}} \geq p^*$

Linear discriminant analysis (LDA) is a generative approach

- **Bayes' theorem:**

$$\Pr[Y = 1 \mid X] = \frac{\Pr[Y = 1 \ \& \ X]}{\Pr[X]} = \frac{\pi_1 f_1(x)}{\pi_0 f_0(x) + \pi_1 f_1(x)}$$

- Need to model
  - $\pi_k = \Pr[Y = k]$: proportion of class k
  - $f_k(x) = \Pr[X = x \mid Y = k]$: probability of $X = x$ conditioned on class k

Classification: Key points

Decision boundary:

- The set of x where $\Pr[Y = 1 \mid X = x] = \Pr[Y = 0 \mid X = x]$
- Both logistic regression and LDA yield linear decision boundary

Choice of p^*:

- The threshold $p^* \in [0, 1]$ affects "conditional probability $\rightarrow$ class prediction"
  - Small p^*: more positive prediction
  - Large p^*: more negative prediction
- To choose optimal p^*, we balance the two types of errors (FP vs. FN)

Confusion matrix & Receiver operating characteristic (ROC) curve:

- Confusion matrix: 2-by-2 table of all possible classification outcomes
  - TP, FN, FP, TN
- ROC curve: The path of (FPR, TPR) for all $p^* \in [0, 1]$
  - Can be used to choose p^*

Model assessment: Error metrics

Regression models: Commonly use **MSE** (Mean Squared Error):

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2.$$

- Lower MSE indicates a better fit

Classification models: Often use **error rate**:

$$\text{Error Rate} = \frac{\#\,\text{Misclassified}}{\text{Total Sample Size}}$$

- Lower error rate indicates a better fit
- **False Positives (FP)** vs. **False Negatives (FN)** may also matter
- A confusion matrix or ROC curve can help visualize these outcomes

Model assessment: Bias-variance tradeoff

Training vs. test error:

- We fit a model using training data to minimize training error
- We want the model to perform well on test data (small test error), which is not guaranteed

Bias-variance tradeoff:

- More flexible models tend to fit training data better, but can fail to generalize

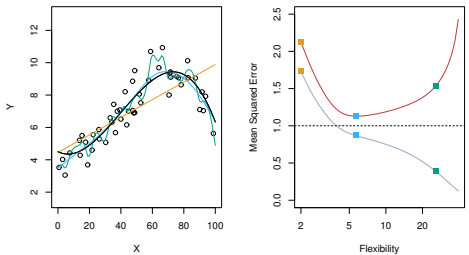


Figure: As model flexibility increases, training MSE typically goes down, while test MSE may go back up [JWHT21, Figure 2.9]

- High flexibility $\Longrightarrow$ low bias but potentially high variance
- Low flexibility $\Longrightarrow$ higher bias but lower variance
- Dashed line: irreducible error (not explainable by X)

Resampling methods: Cross-validation and bootstrap

Needs:

- Estimate test error *using only training data*
- Valid inference for *flexible or complex models beyond linear regression*

Cross-validation: Estimate test error from training data

- *Validation set approach:* Split training data into folds, hold out some for validation
- *Cross-validation*: Repeat across each fold
  - k-fold CV, LOOCV: Advantages and drawbacks

Bootstrap: Estimate sampling distribution from a single dataset

- *Resampling* from the given dataset with replacement to generate synthetic datasets
- If the original dataset is representative of the underlying distribution...
  - Bootstrap samples will look like i.i.d. sample from the nature
  - Can construct confidence interval, etc.

Model selection

In reality, we might have many predictors, unsure which are truly helpful

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \epsilon.$$

Best subset selection: Identify a relevant subset of predictors, then fit via least squares

- *Best subset selection:* Try *all* subsets of predictors, and pick the one that performs the best
  - With p predictors, there are 2^p possible subsets
  - Compare models of different sizes carefully (recall R^2 vs. R^2_{adj})
- Forward/backward stepwise selection: Computationally lighter alternatives

Regularization: Add a penalty term that favors "simpler" models

- *Ridge:* Add ℓ_2 penalty $\sum_{j=1}^{p} \beta_j^2$
  - Ridge is stable under collinearity and has simpler closed-form solutions
- *Lasso:* Add ℓ_1 penalty $\sum_{j=1}^{p} |\beta_j|$
  - Lasso can yield sparse solutions (some $\beta_j = 0$)

Hypothesis test: Basics

Single test:

- H_0: "no signal" vs. H_a: "signal"
- Reject H_0: "Discovery" of "signal"

	H_0 is true	H_0 is not true
Reject H_0	Type-I error (FP)	Correct (TP)
Not reject H_0	Correct (TN)	Type-II error (FN)

$\Longrightarrow$ Pr(Type I error) = Pr(reject a true null)

- By setting threshold α, we want to control Pr(Type I error) below α

Multiple hypothesis testing

Setting:

- Suppose we have m predictors to test simultaneously
- Each test has a per-hypothesis Type I error rate $\alpha > 0$

Problem:

- With m tests, we have m chances for false positives
- Probability of ≥ 1 false rejection $\approx 1 - (1 - \alpha)^m$, which can be large as m grows
  - e.g. at $m = 20$ and $\alpha = 0.05$, we expect ≈ 1 false positive on average

How to address?

- *Family-Wise Error Rate (FWER)* ensures probability of *any* false positive is $\leq \alpha$
  - Bonferroni correction, Holm's method
- *False Discovery Rate (FDR)* limits the *proportion* of false positives among all rejections
  - Benjamini-Hochberg procedure
- Review Midterm2 & homework for definition of FWER/FDR and further details

Beyond linear models: Basis functions

Linear regression is powerful but can sometimes be restrictive

- Assumes $Y \approx \beta_0 + \sum_{j=1}^{p} \beta_j X_j$, i.e. a purely linear combination of predictors
- Real data often exhibits more complex, nonlinear relationships

Linear regression with basis functions: Transform X to construct new features $\{b_1(X), \ldots, b_K(X)\}$, then fit a linear model in those features:

$$Y \approx \beta_0 + \beta_1 b_1(X) + \cdots + \beta_K b_K(X)$$

- *Polynomials*: $b_1(X) = X$, $b_2(X) = X^2, \ldots$
- *Step functions*: $b_1(X) = I(c_1 < X \leq c_2)$, $b_2(X) = I(c_2 < X \leq c_3), \ldots$
- *Splines*: piecewise polynomials with continuity constraints
  - Best of both polynomials and step functions
  - *Piecewise polynomials* of degree d, joined at *knots* (cutpoints)
  - Degree-d spline: *continuity constraints* at each knot, up to $(d-1)$-th derivatives

Principal component analysis

Problem Setup:

- We have data of $X \in \mathbb{R}^p$, where p is possibly large
- We want to *reduce dimension* to $r \ll p$ while retaining most "information"

PCA approach:

- **Project data (X) onto an r-dimensional subspace** (spanned by r vectors)
- These r vectors (=PCs) are chosen to capture maximum variance in X
  - **First PC:** a unit vector $\mathbf{u}_1 \in \mathbb{R}^p$ that maximizes variance, i.e.,

$$\mathbf{u}_1 = \underset{\|\mathbf{u}\|=1}{\operatorname{argmax}} \frac{1}{n} \sum_{i=1}^{n} \left(\mathbf{u} \cdot \mathbf{x}_i\right)^2$$

  - Subsequent PCs are defined analogously, each orthogonal to all preceding PCs
- Unsupervised learning: no Y is used

Proportion of variance explained (PVE) and scree plot:

- Tradeoff between keeping too few vs. too many principal components
- "Elbow" in a scree plot

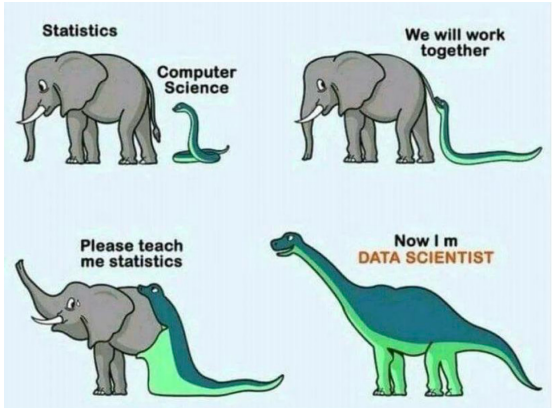
Clustering

Setup:

- *Data:* $\mathcal{X} = \{X_1, \ldots, X_n\} \subset \mathbb{R}^p$
- *Goal:* Partition a dataset (no response labels) into subgroups of "similar" observations
- *Unsupervised*: Typically used for exploratory analysis or hypothesis generation
- No single "correct" distance or method; different choices lead to different clusterings

K-means	**Hierarchical**
- Partition data into K clusters - Minimizes within-cluster variation	- Builds a *dendrogram* from bottom-up - Cut at a certain height to obtain clusters
- Simple, computationally fast - Easy-to-interpret "centroids" for each cluster	- No need to specify K in advance - One dendrogram can yield many clusterings
- Must pre-specify K - Local search can yield suboptimal solutions	- Greedy merges rely on linkage choice - Nested clusters may be less optimal

Conclusion & Best Wishes



- **Keep learning:** Continue learning, explore more advanced topics, and stay curious
- **Best of luck** in your upcoming exams and in all your future endeavors!

References

Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani.
An Introduction to Statistical Learning: with Applications in R, volume 112 of *Springer Texts in Statistics*.
Springer, New York, NY, 2nd edition, 2021.