Grades

Final grades will be renormalized based on grades

- **Renormalization**: centered around A- (or B+) with a variance reaching A+ and B- (or C+).
- **Bottom line**: cut-off line for failing a class: E. Cut-off line for losing credits: D. Under what conditions you will get D? If you just continue your current efforts, none of you will get D.

However, it would be ideal for you to focus on learning new things, rather than grading.

Feedbacks on research ideas

1. **Submission process**. Submission from one member of a team is good enough.
2. **Page limit**. **Why did I set up a page limit?**
   - You see it all the time in your life. (e.g. IEEE)
   - Fairness in grading.
   - I did not deduct any point for exceeding the page limit, but you will **lose** points in the proposal or final report when you **exceed** the page limit.
   - How to handle a page limit? Step up & down approach.
3. **Advanced approaches**.
   - CV and spatiotemporal modeling (GNN) – you are encouraged to explore but **make sure** you can deliver a course project!
4. **Team**
   - Principle: same grades for all the team members.
   - Rare exception: different grades for some team members.
   - Please coordinate within a team.

Next step: proposal and mid-term presentation

Information

- Proposal deadline: **March 3**1, 2023 (3 days after the presentation on **March 28**)
- Submission: both slides and proposal.
- Page limit on proposal: 3 pages.
- Page limit on presentation: 10 slides.

Proposal (6 pts)

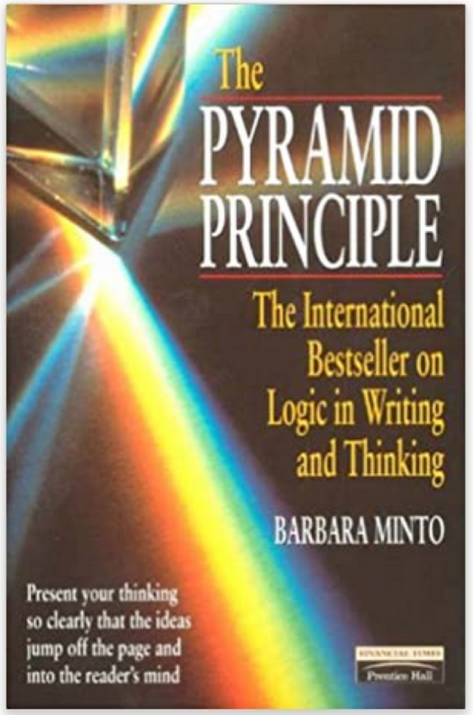
- **Introduction**.
- **Background**. Please include at least **ten** research articles.
- **Approaches**: **data and method**.
- **Preliminary findings (2pts)**. This section needs to be **at least one page** long.
- **Conclusion and discussions**.

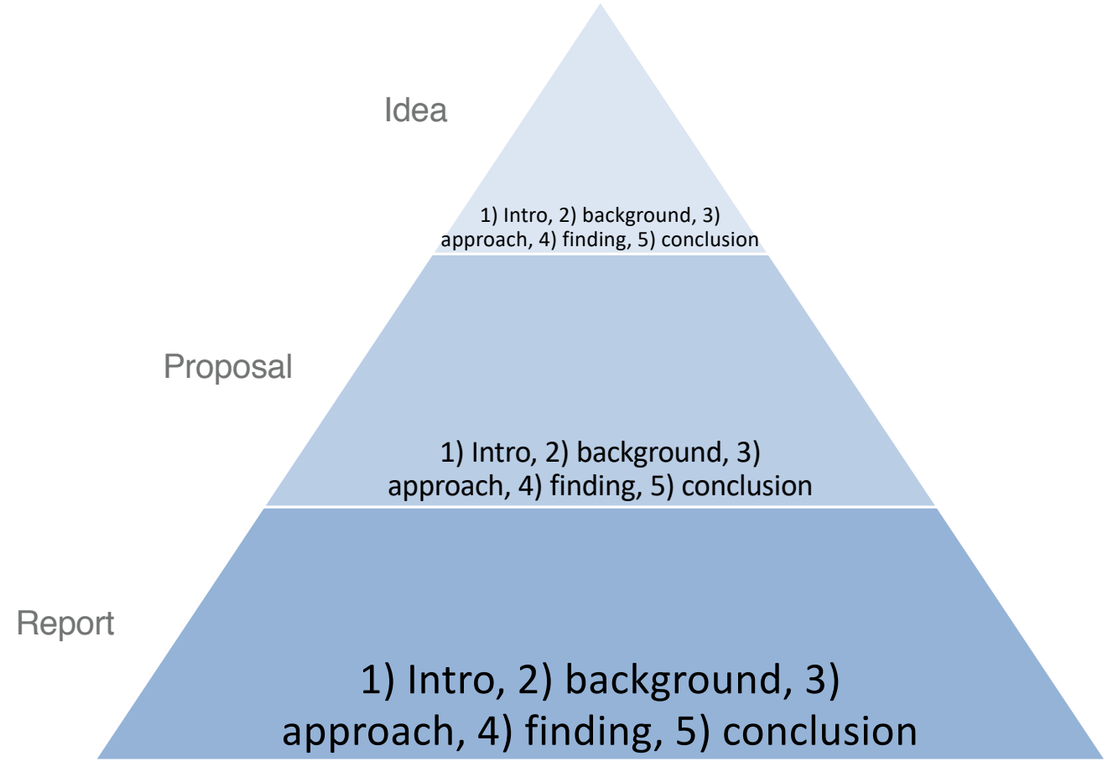
Presentation (4pts)

- The mid-term presentation is limited to **ten slides** and **ten minutes**, and the structure should be corresponding to that of your proposal, including (1) introduction, (2) background, (3) approaches, (4) preliminary findings, and (5) conclusion and discussions. The preliminary findings should have **at least three slides**. Please use the section names as titles on your slides.

Question: Why do I seem to *force* you to develop preliminary findings?

Pyramid principle in project development





Why pyramid principle? (1) result-oriented, (2) balanced, and (3) facilitate iteration.

Q&A about Psets, Projects, and Grading?

Review lecture 07

[Parts 1 and 2]

1

Spatial autocorrelation

(X, A)

$$\sum_{i=1}^{N}\sum_{j=1}^{N} a_{ij}(x_i - \bar{x})(x_j - \bar{x})$$

2

Spatial regressions for fitting node features

$(X, A) \rightarrow y_i$

$$\sum_{j} a_{ij}x_j$$

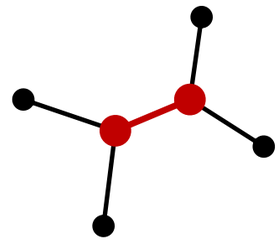
Continuing lecture 07

[Parts 3, 4, and 5]

1	2	3	4	5
Spatial autocorrelation (X, A)	Spatial regressions for fitting node features $(X, A) \rightarrow y_i$	Gravity models for fitting edge features $(X, A) \rightarrow y_{ij}$	Community detection algorithms $(X, A) \rightarrow z_i$	Summarizing urban network analysis

Part 3. Gravity model for fitting edge features

- Using node features to enhance the analysis of edges.



Gravity model

$$M_{ij} = \frac{P_i * P_j}{d_{ij}^2}$$

- Let's use this model to analyze migration.
- M_{ij} could be the migration between nodes i and j.
- P_i is the population (or a general push factor) of origin node i.
- P_j is the population (or a general pull factor) of destination node j.
- d_{ij} is the distance (or a general friction factor) from origin i to destination j

Naming Logic

- Adapt Newton's law to social science to estimate the spatial interaction between two places

A more general form of the gravity model

$$M_{ij} = \frac{P_i^{\alpha} * P_j^{\beta}}{d_{ij}^{\gamma}}$$

- M_{ij} could be the migration between nodes i and j.
- P_i is the population (or a general push factor) of origin node i.
- P_j is the population (or a general pull factor) of destination node j.
- d_{ij} is the distance (or a general friction factor) from origin i to destination j.

- Let's generalize the baseline gravity model
- Use α, β, and γ to replace the initial parameters so that we can estimate the coefficients.
- This formula is closely related to things we learnt before - (1) linear regression, and (2) network analysis.

Gravity model and linear regression

Taking log transformation on both sides,

$$M_{ij} = \frac{P_i^{\alpha} * P_j^{\beta}}{d_{ij}^{\gamma}}$$

We obtain

$$\log M_{ij} = \alpha \log P_i + \beta \log P_j - \gamma \log d_{ij}$$

Question: Is it a linear regression? Why or why not?

Notes

- It is a linear regression because it is linear in parameters.
- It can also be transformed to a logistic regression $P(y_{ij}= 1) = \sigma(\alpha \log P_i + \beta \log P_j - \gamma \log d_{ij})$. e.g., y_{ij} measures the existence of migration/visitation between nodes i and j.
- Again, nodes i and j could be any spatial unit, e.g., country, states, counties, tracts, blocks, etc.

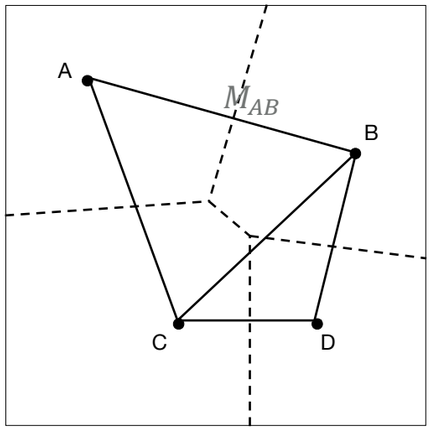
Gravity model and fitting an edge feature

With

$$M_{ij} = \frac{P_i^{\alpha} * P_j^{\beta}}{d_{ij}^{\gamma}}$$

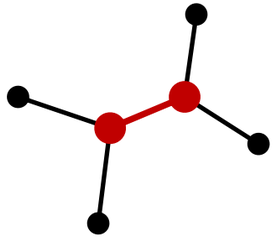
We obtain

$$\log M_{ij} = \alpha \log P_i + \beta \log P_j - \gamma \log d_{ij}$$



Notes

- The dependent variable has two indices i and j, so M_{ij} is an edge feature.
- e.g. let $i = A$ and $j = B$, then M_{ij} describes the migration from parcel A to parcel B. To predict the edge feature M_{AB}, we use the two nodes' features P_A and P_B and their distances d_{AB} as inputs.
- Sometimes, people ignore the distance factor d_{ij}. Then the regression becomes $\log M_{ij} = \alpha \log P_i + \beta \log P_j$, which uses only node features to predict an edge feature.
- Urban applications: migration, visitations, or gentrification.
- **Question**: How to estimate α, β, and γ?

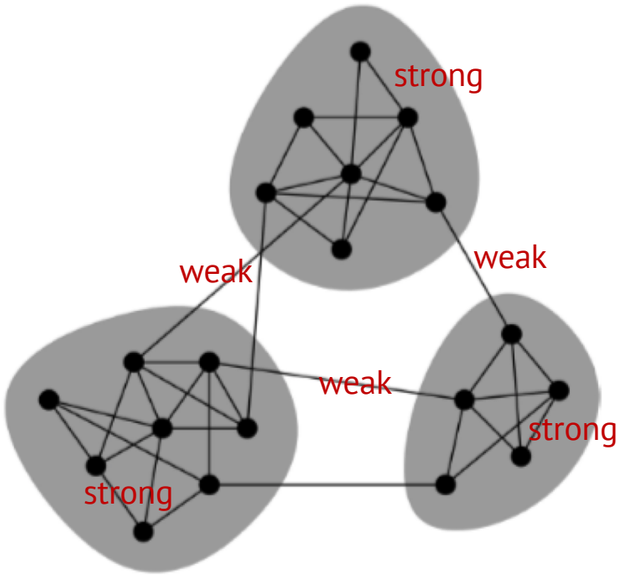


Part 4. Community Detection

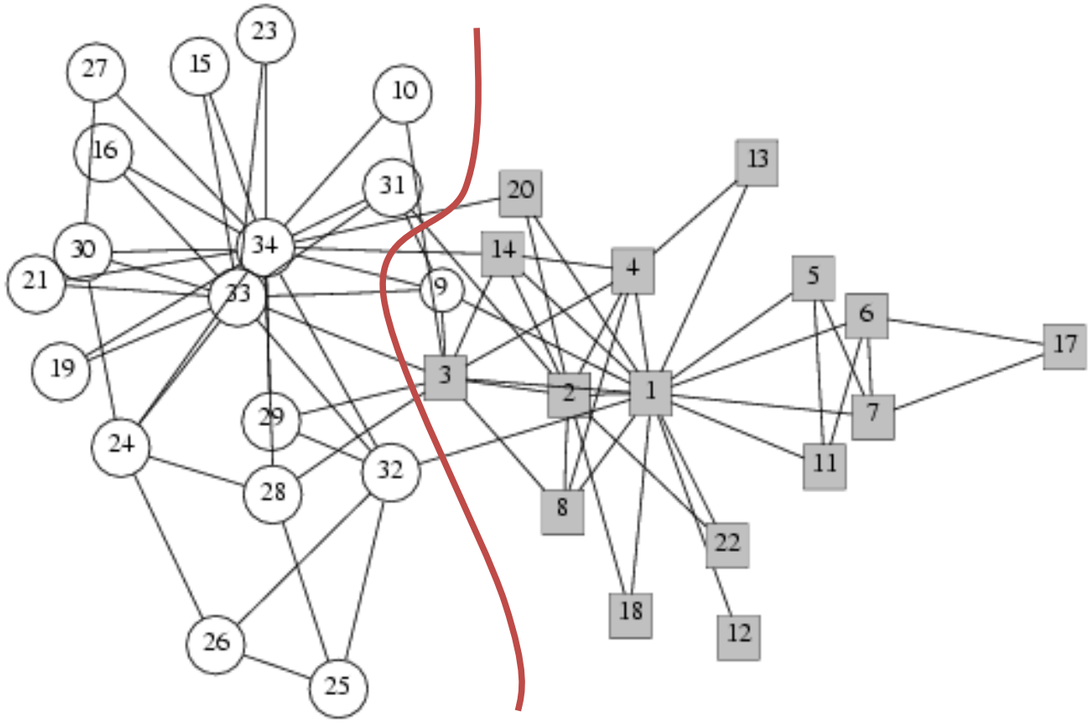
Community detection

Network communities (DGP):

- Sets of nodes with lots of internal connections but few external ones (to the rest of the network).
- Within-community edges are strong (more frequent).
- Cross-community edges are weak (more infrequent).
- However, the weak ties are powerful in terms of providing diverse sources of information, helping job hunting, bridging across communities, mitigating segregation, etc. This weak tie theory was proposed by Granovetter in the 1950s.



Story about Zachary's Karate Club



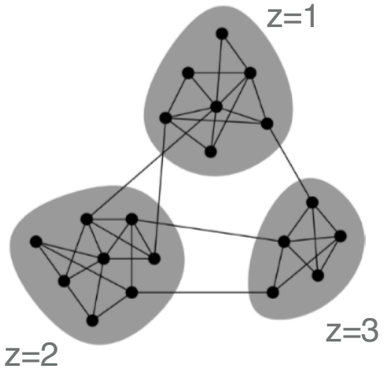
- Observed social ties & rivalries in a university karate club.
- During the study, conflicts led the group to split into two groups.
- How to detect the two communities before the split? E.g. min-cut approach

How to automatically detect communities?

Define **Modularity Q**

Given a partitioning of the network into communities $z \in S$, Modularity Q is measured by

$$Q \propto \sum_{z \in S} [\,(\#\ edges\ within\ group\ z) - (expected\ \#\ edges\ within\ group\ z)]$$



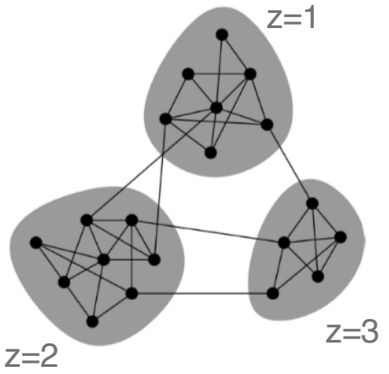
Notes

- The first term $\sum_{s \in S}[\,(\#\ edges\ within\ group\ s)]$ measures the edges within group s. When the community indices are correctly specified, then this term takes a large value. e.g., s = 1, 2, 3 in the graph above.
- The second term is a baseline edge count, which involves some configuration model that we won't discuss. But intuitively, the logic here is similar to computing a deviation from the average value, i.e., $(x_i - \bar{x})$, which is shown in correlation, regression, Moran's I, etc.
- Categorically, Q is similar to MSE or log-likelihood as an objective to be optimized.
- Eventually we want to maximize Q to learn the community structures S.

Formula for Modularity Q

Modularity Q equals to

$$Q(G,S) = \frac{1}{2m}\sum_{z\in S}\sum_{i,j\in z}\left(A_{ij} - \frac{k_i k_j}{2m}\right),$$



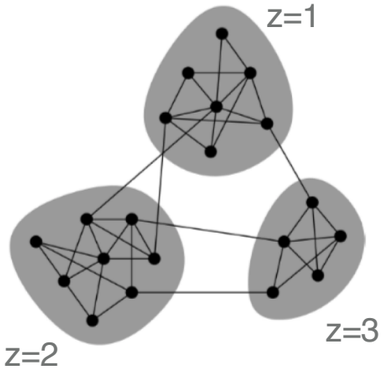
Where

- m is the number of edges.
- A_{ij} is the value from the adjacency matrix.
- k_i and k_j are the node degrees of nodes i and j.
- The first term $\frac{1}{2m}\sum_{z\in S}\sum_{i,j\in z}A_{ij}$ is a normalized counts for the within-community edges. If the community indices are correctly specified, this value should be large.
- The second term $\frac{1}{2m}\sum_{z\in S}\sum_{i,j\in z}\frac{k_i k_j}{2m}$ measures the baseline edge counts conditioning on a fixed node degrees. I will skip detail here.
- When the $Q(G,S)$ is larger, we are more likely to have specified the correct community structures.

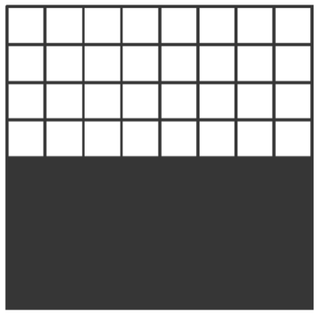
Modularity Q for community detection

Therefore, we seek to learn z_i by maximizing the modularity Q:

$$Q(G,S) = \frac{1}{2m}\sum_{z\in S}\sum_{i,j\in z}\left(A_{ij} - \frac{k_i k_j}{2m}\right),$$



- Modularity values take range [-1, 1].
- It is positive if the number of edges within groups exceeds the expected number
- Empirically, when Q is greater than 0.3-0.7, it means a significant community structure.
- Louvain algorithm is the most common method to detect the communities.
- Similar to all our previous lectures, it is one line of script in python.
- Urban applications: community detection is critical for understanding spatial segregation: (1) strong connection within communities, and (2) weak connections across communities.



Different goal

- The goal of community detection is quite different from the regressions because z_i is unobserved data while y_i is observed data.
- This perspective will be revisited in the unsupervised learning class. (next class)

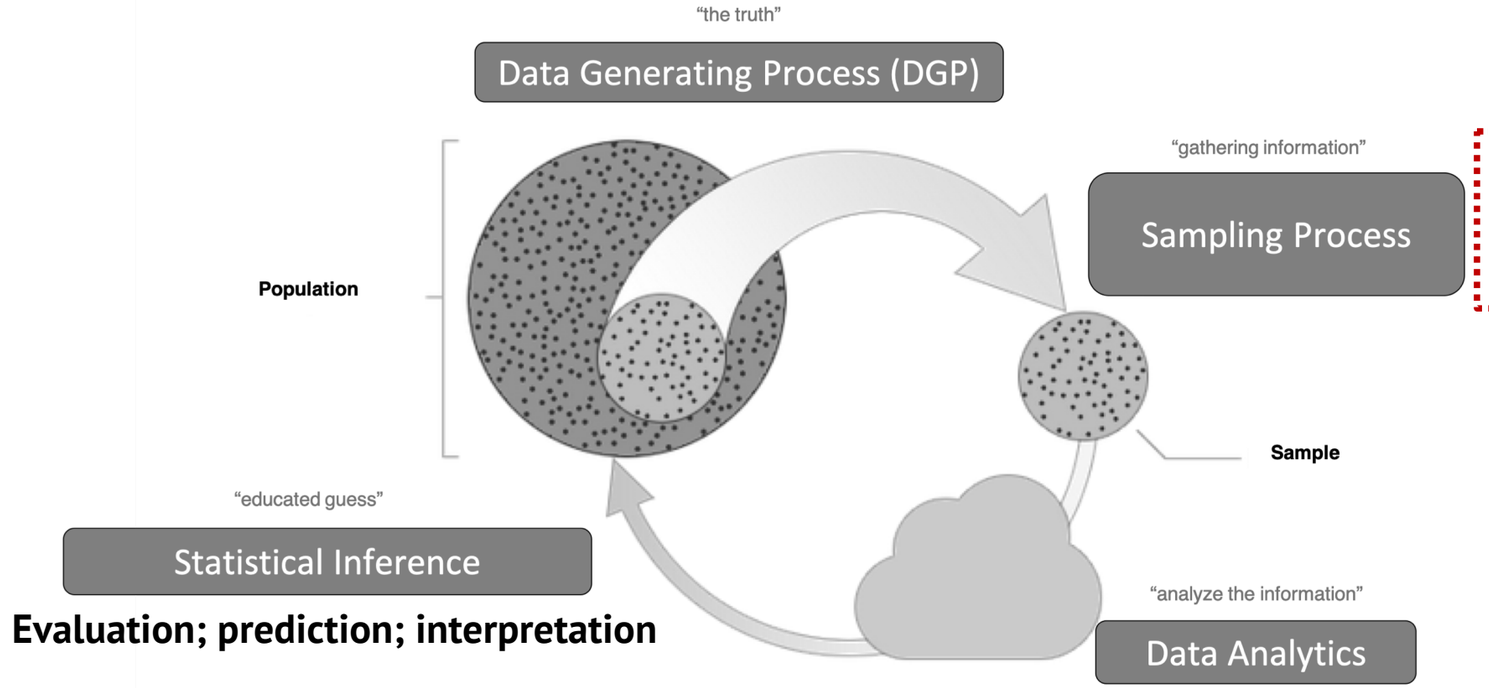
Example. community detection in our class

1. Detect communities with observed edges. (community detection) – example with modularity.
2. Detect communities/clusters with node features. (unsupervised learning) – example with feature similarity

Part 5. Summarizing urban network analysis

Reviewing lecture 07

Regression: recovering $E[Y|X; A]$



Data: ~~i.i.d.~~
Graph: (V, E)
Graph Features: (X, A)

Evaluation; prediction; interpretation

(1) Spatial regression $(X, A) \rightarrow y_i$; (2) gravity model $(X, A) \rightarrow y_{ij}$;
(3) community detection $(X, A) \rightarrow z_i$;
Estimation/Training/Learning: OLS & MLE

Back to the general diagram

Univariate Linear Regression

1. Establish the goal (DGP)
   e.g. recovering $E[Y|X]$
2. Make modeling assumptions (e.g. i.i.d.)
   e.g. $E[y_i|x_i] = \beta_0 + \beta_1 x_i$
3. Estimate the model by minimizing an objective
   e.g. $\underset{\beta_0,\beta_1}{\operatorname{argmin}} \sum_{i=1}^{N} (y_i - \beta_0 - x_i\beta_1)^2$
4. Examine the performance
   e.g. R^2
5. Use the model (interpretation, prediction, etc.)
   e.g. $\hat{\beta}_0, \hat{\beta}_1$

Spatial regression

1. Establish the goal (DGP)
   e.g. recovering $E[Y|X; A]$
2. Make modeling assumptions (e.g. i.i.d.)
   e.g. node $E[y_i|x_i] = \beta_0 + \beta_1 x_i + \rho \sum_j a_{ij} x_j$
   e.g. edge $E[y_{ij}|x_i] = \sigma(\beta_0 + \beta_1 x_i + \beta_2 x_j)$
3. Train the model by minimizing an objective
   e.g. OLS or MLE
4. Examine the performance
   e.g. R^2, log-likelihood, accuracy, etc.
5. Use the model (interpretation, prediction, etc.)
   e.g. $\hat{\beta}_0, \hat{\beta}_1, \ldots \hat{\beta}_k; \rho$

Last comments about the terminology in urban network analysis

"Message passing", "random walk", etc. are not the common terms in the traditional GIS, spatial econometrics, and transportation modeling. Hence try to minimize the use of these terms in your academic writing if you target only the traditional audience.

However, these terms are useful because they present a general network analytical perspective.

The terminology is used so that we can pave the way for the discussion about CNN & GNN in deep learning.

Survey (20 minutes)
https://forms.gle/H6UNp9avPcSheAXe9

Modifying a statement in the statistical analysis

Linear regression

Recovering $\mathrm{E}[\mathrm{y_i}|\mathrm{x_i}]$ with OLS

Logistic regression

Recovering $\mathrm{P}[y_i|x_i]$ with MLE

Notes

- Intuition about $\mathrm{P}[y_i|x_i]$
- The relationship between $\mathrm{E}[\mathrm{y_i}|\mathrm{x_i}]$ and $\mathrm{P}[y_i|x_i]$.
- Urban applications: equality, equal opportunity, and risk allocation.
- Clarifying that we could seek to recover things beyond a conditional mean function. e.g. $\mathrm{P}[y_i|x_i]$, $\mathrm{P}[x_i, y_i]$, etc. Example. ChatGPT

General Form

$x \rightarrow y$

URP 6931. Introduction to Urban Analytics

Lecture 08: Supervised Learning - Classification

Instructor: Shenhao Wang
Assistant Professor, Director of Urban AI Lab
Department of Urban and Regional Planning
University of Florida

Outline Lecture 08: Supervised Learning

Today focus on only Part 1.

1

Revisiting logistic regression

- i.i.d. data structure
- Minor differences - new terminology
- Major differences
- Computational practice

2

Regularization

3

Diverse supervised learning algorithms

Part 1. Revisiting logistic regression with a ML perspective

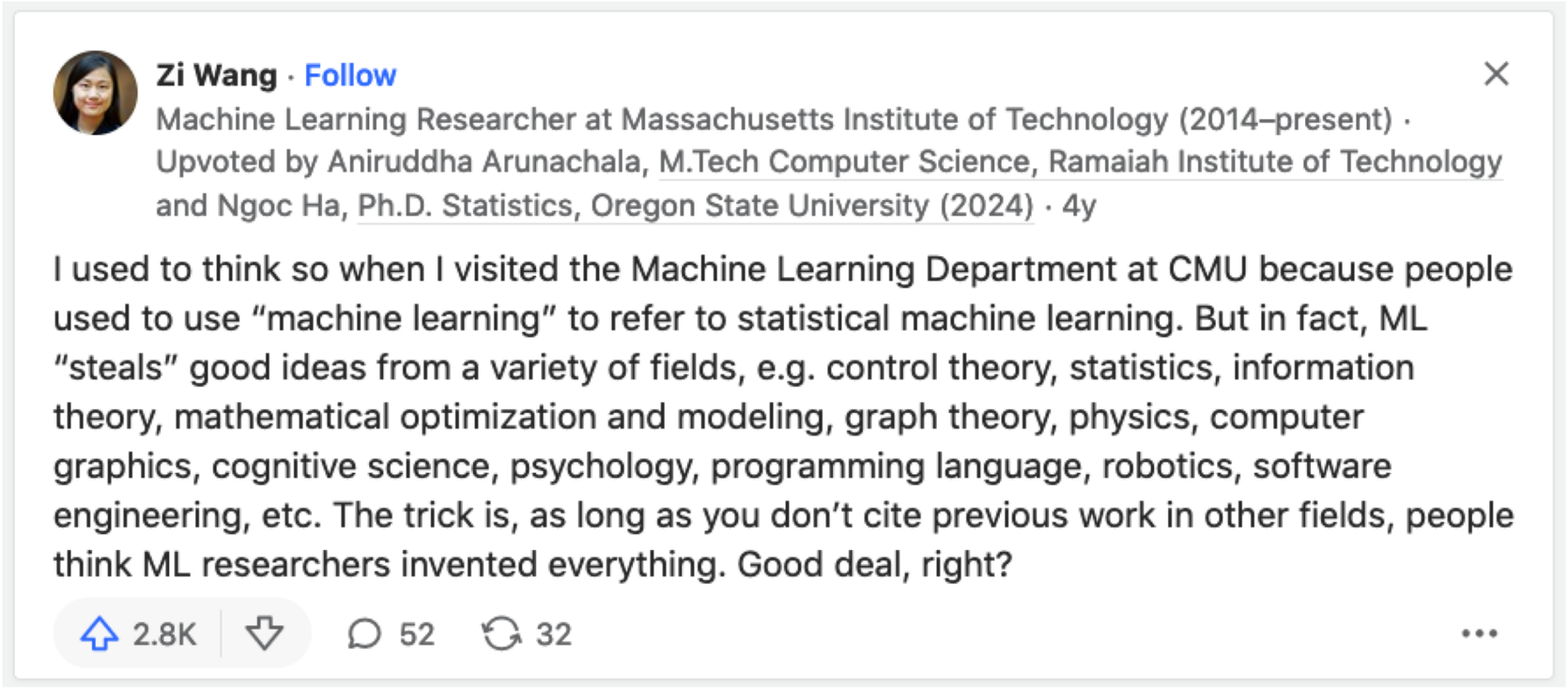
Change terminology from the statistical modeling

- Supervised learning: a machine learning paradigm with labeled data y_i (as opposed to unsupervised learning).
- Feature vectors/inputs: $\mathrm{x} = [x_1, x_2, \dots x_d] \in R^d$
- Labels/outputs: $y \in \{-1, 1\}$
- Training set: $\mathrm{S_n} \in \{x_i, y_i\}, i = 1, \dots, N$
- Parameters: θ
- Classifier/hypothesis $\mathrm{h}: \mathrm{R^d} \to \{-1, 1\}$
- Learning the classifier.
- Training error $L_N(h) = \frac{1}{N}\sum_{i=1}^{N}\left[[h_\theta(x_i) \neq y_i]\right];\ i = 1, \dots, N$
- Testing error (aka generalization error) $L(h) = \frac{1}{N}\sum_{i=N+1}^{N'}\left[[h_\theta(x_i) \neq y_i]\right];\ N' \to +\infty$

Notes

- The set up of supervised learning and regressions are highly similar.
- Terminology change is a hurdle but not essential. e.g. machine learning: use machine (computer) to learn (estimate) your model.
- However, there are also key differences. e.g. training vs. testing.

Quora question: Is Machine Learning just glorified statistics?



Zi Wang · Follow

Machine Learning Researcher at Massachusetts Institute of Technology (2014–present) · Upvoted by Aniruddha Arunachala, M.Tech Computer Science, Ramaiah Institute of Technology and Ngoc Ha, Ph.D. Statistics, Oregon State University (2024) · 4y

I used to think so when I visited the Machine Learning Department at CMU because people used to use “machine learning” to refer to statistical machine learning. But in fact, ML “steals” good ideas from a variety of fields, e.g. control theory, statistics, information theory, mathematical optimization and modeling, graph theory, physics, computer graphics, cognitive science, psychology, programming language, robotics, software engineering, etc. The trick is, as long as you don’t cite previous work in other fields, people think ML researchers invented everything. Good deal, right?

2.8K 52 32

Classification Intuition

Labels/outputs: $y \in \{-1, 1\}$

Feature vectors/inputs: $\mathrm{x} = [x_1,\ x_2] \in R^d$

Example

$y \in \{-1, 1\}$. $y = 1$ (ride hailing); $y = -1$ (public transit);

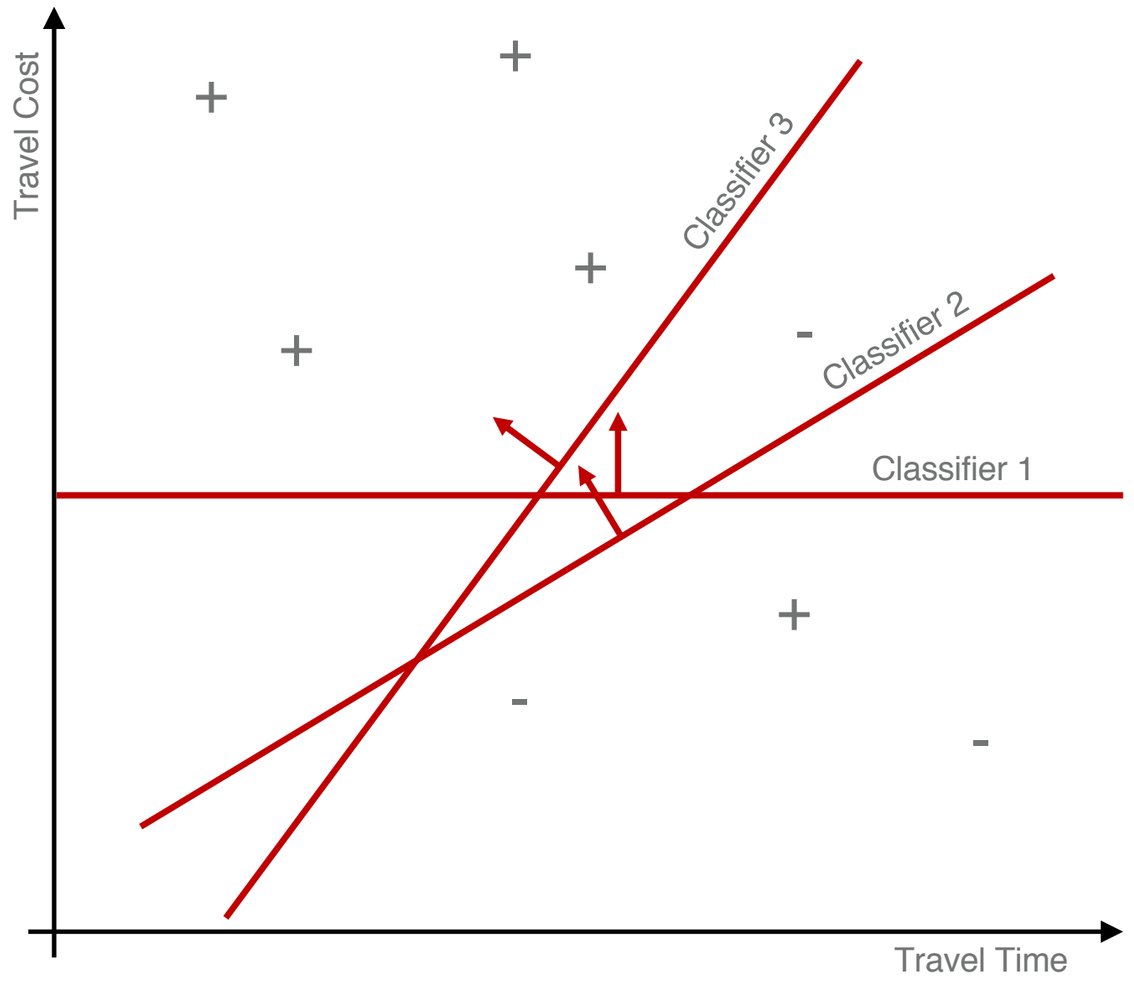
x_1: travel time; x_2: travel cost

Task

Find a linear classifier (h) that can minimize the testing error

Q: How many errors are there for each classifier?

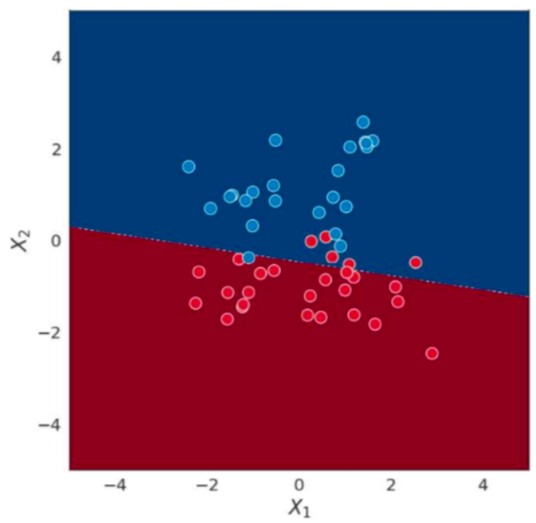
- Classifier 1. 2 errors
- Classifier 2. 2 errors
- Classifier 3. 1 error



Two types of classifiers

Comparing deterministic and probabilistic classifiers

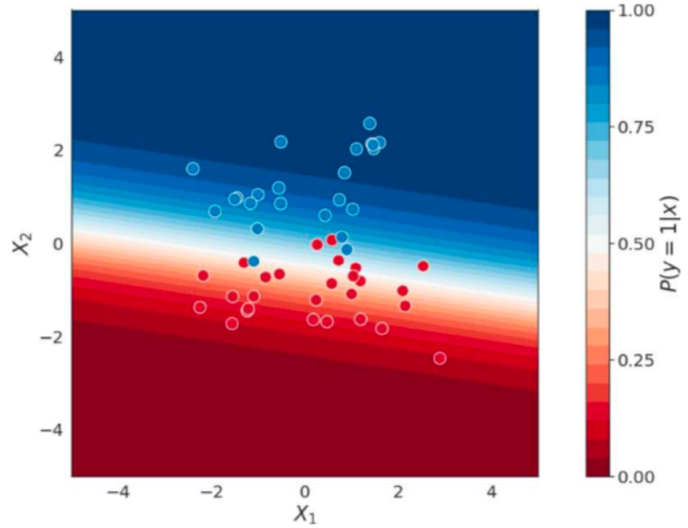
- Probabilistic classifiers can be turned into deterministic ones.
- Borderline cases in deterministic classifiers are problematic
- Goal of probabilistic classifiers – learning $p(y_i|x_i)$



Deterministic classifiers

e.g. The person will use not use public transit if the price is higher than $1.0

example: decision Tree



Probabilistic classifiers

e.g. The person has only 20% chance to use public transit if the price is $1.0

example: logistic regression

Let's discuss the logistic regression in machine learning…

Let $\theta^T x = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ ... + \theta_d x_d$

$\theta^T x$ can take real values, so we cannot use it directly for classification.

Therefore, we introduce the sigmoid (logistic) function:

$$\sigma(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

We can obtain the logistic regression as

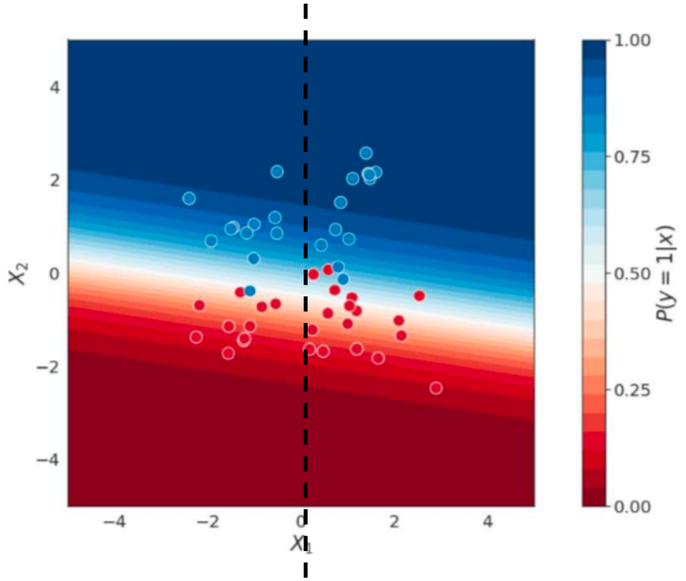
$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

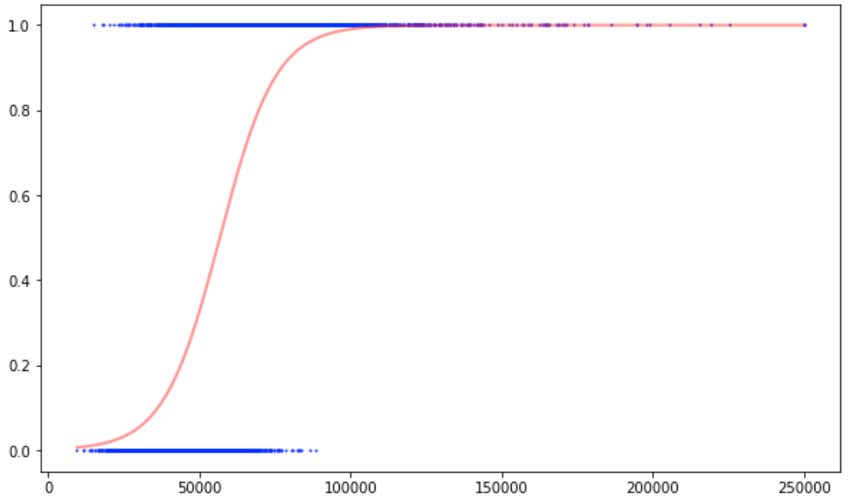
Logistic regression in machine learning

Logistic regression

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

2D and 1D visualization





Constructing the loss function with MLE

The likelihood of the data set is given by:

$$l(\theta) = \prod_{i=1}^{N} P(y_i|x_i;\theta)$$

We want to identify the $\hat{\theta}$ that maximizes the likelihood

$$\hat{\theta} = \text{argmax}_\theta \, l(\theta)$$

We could take log transformation on both sides:

$$\hat{\theta} = \text{argmax}_\theta \sum_{i=1}^{N} \log P(y_i|x_i;\theta)$$

Which you already saw in the previous lecture. The objective is the log-likelihood, and the negative value is called cross-entropy loss.

Learning parameters by minimizing the cross-entropy loss

For binary discrete classification, we could use the following cross-entropy loss:

$$J(\theta) = -\sum_{i=1}^{N} y_i \log h_\theta(x_i) + (1 - y_i) \log(1 - h_\theta(x_i))$$

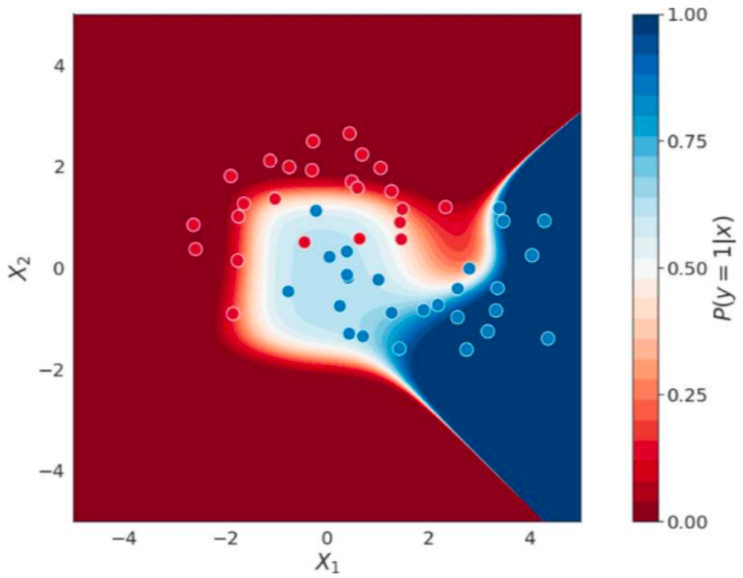
Then learning the parameters by

$$\hat{\theta} = \operatorname{argmin}_\theta J(\theta)$$

How to enrich the model?

We can introduce a nonlinear decision boundary by transforming the features.

$$x = \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_1 x_2 \\ x_1^2 \\ x_2^2 \\ x_1^2 x_2 \\ x_2 x_2^2 \\ \dots \end{bmatrix}$$



So far, there is nothing different from the statistical approach…But let's discuss three differences.

1. How to learn the parameters in ML?

Gradient decent for training the logistic regression.

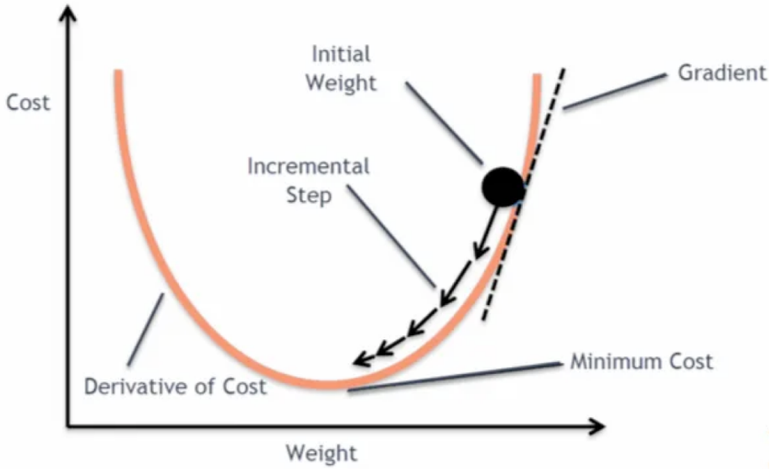
Initialize θ

$\forall\, j$, repeat until the following converges

$$\theta_j \leftarrow \theta_j - \alpha \sum_{i=1}^{N} (h_\theta(x_i) - y_i) x_{ij}$$

in which the second term is the gradient of the objective function.

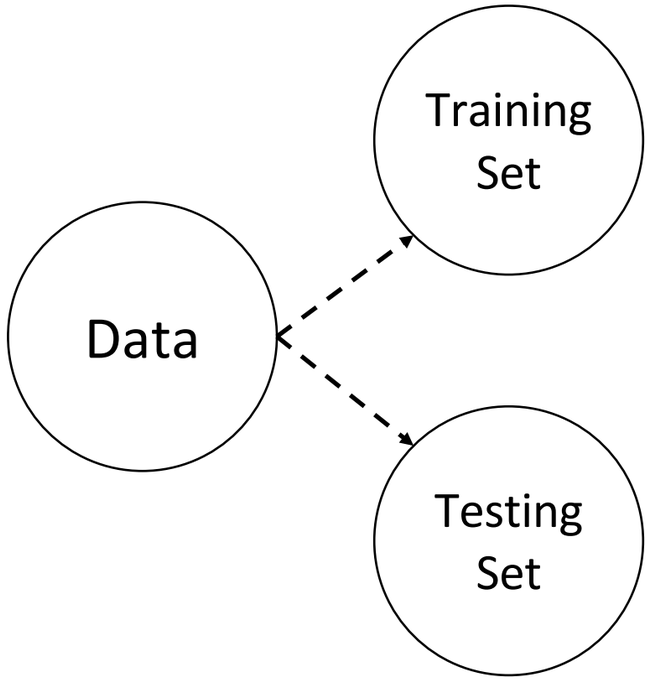
The image shows the intuition of gradient descent.



Notes

- It is “cheap” but FAST! As a result, (stochastic) gradient descent is widely used in deep learning.
- ML does not have the traditional statistical properties (e.g., statistical significance) partially due to GD.

2. How to evaluate the model?

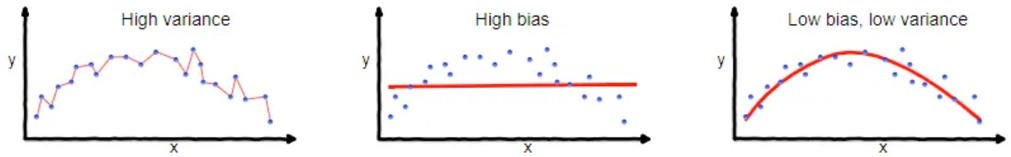


- Example. Splitting the data into training and testing sets (80% vs. 20%).
- It is a common practice for model evaluation.
- The performance in the training set is called the training error, which we don't care.
- The performance in the testing set is called the testing error, which we really care.
- Fundamentally, ML seeks to achieve a low testing error, or more accurately generalization error.
- We will train the model using the training set, and evaluate the model using only the testing set.

Why training + testing split?

3. It is because we are worried about not only a too simple but also too complex model

- When a model is too simple, we enrich the model.
- When a model is too complex, we regularize the model.
- Essentially, we want to learn the true DGP.
- Statistics: tradeoff of *bias* and variance; ML: tradeoff of approximation and *estimation* errors



overfitting **underfitting** **Good balance**

Logistic regression in statistics vs. ML

Terminology (minor)

Variables

Independent and dependent variables

Estimation

Model

etc

Terminology (minor)

Features

Inputs and outputs

Training/learning

Classifier/algorithm

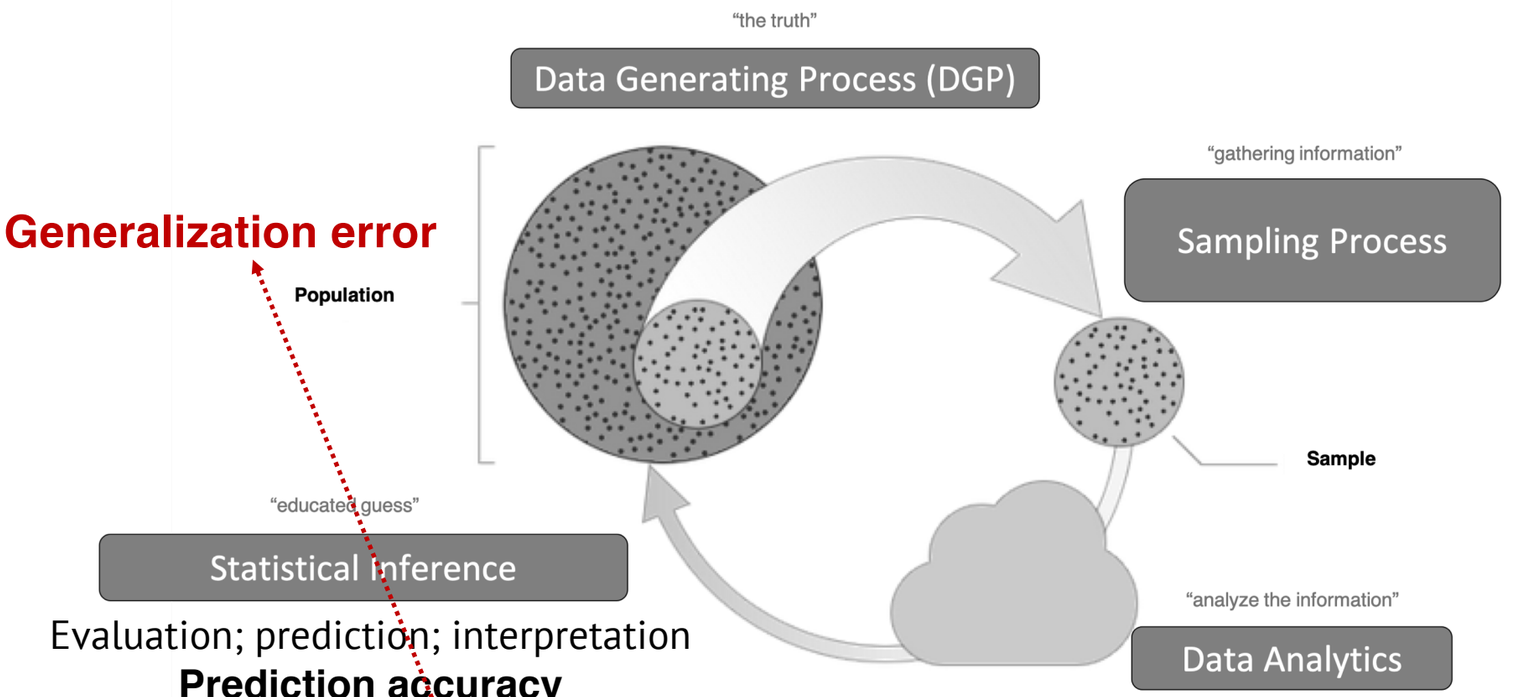
etc

New things (major)

1. Gradient descent
2. Training vs. testing set – generalization error.
3. We worry about too complex a model – regularization.
4. Various supervised learning algorithms

The general diagram for supervised learning

Supervised learning: learning true $P(Y|X)$



Generalization error

i.i.d.
Y: {-1, 1}

Evaluation; prediction; interpretation
Prediction accuracy

training and testing error

Regression: $P[Y = 1|X] = \sigma(\hat{\beta}_0 + \hat{\beta}_1 x_{1i} + \hat{\beta}_2 x_{2i} + \cdots + \hat{\beta}_k x_{ki})$
Estimation/Training/Learning: MLE

General diagram

Logistic regression in Stats

1. Establish the goal (DGP)
   e.g. recovering $P(y_i|x_i)$
2. Make modeling assumptions (e.g. i.i.d.)
   e.g. $P(y_i|x_i) = \sigma(\beta_0 + \beta_1 x_{1i} + \ ... \beta_k x_{ki})$
3. Train the model by minimizing an objective
   e.g. $\underset{\beta}{\text{argmax}}$ log-likelihood
4. Examine the performance
   e.g. log-likelihood, accuracy, confusion matrix
5. Use the model (interpretation, prediction, etc.)
   e.g. $\hat{\beta}_0, \hat{\beta}_1, ... \hat{\beta}_k$

Logistic regression in ML

1. Establish the goal (DGP)
   e.g. learning $P(y_i|x_i)$
2. Make modeling assumptions (e.g. i.i.d.)
   e.g. $P(y_i|x_i) = \sigma(\beta_0 + \beta_1 x_{1i} + \ ... \beta_k x_{ki})$
3. Estimate the model by minimizing an objective
   e.g. $\underset{\beta}{\text{argmax}}$ log-likelihood
4. Examine the performance
   e.g. training and testing errors
5. Use the model (interpretation, prediction, etc.)
   e.g. $\hat{P}(y_i|x_i)$ for prediction