

# Midterm 2 Prep

Monday, November 2, 2020 7:12 AM

By: Chris Dedow

- 11.2 Models for Variable Selection
  - Good for initial analysis, often don't perform as well on other data
    - Forward selection
    - Backward elimination
    - Stepwise Regression (most common out of three)
  - Slower but better prediction
    - Lasso - add constraint to standard regression equation (basically limit allotment for sum of coefficients)
      - Some coefficients forced to 0 to simplify model
      - Scale data if you're going to be limiting coefficients (units of data will otherwise skew data)
      - If the data is not scaled, the coefficients can have artificially different orders of magnitude, which means they'll have unbalanced effects on the lasso constraint. At each step, the stepwise regression fits a different model. However, different lasso models can be found by varying  $T$ , and  $R$  has a function to automatically generate multiple lasso models.
    - Elastic Net - constrain absolute value of coefficients and their squares (underline is different than Lasso)
      - Advantages:
        - ◆ Variable selection benefits of Lasso
        - ◆ Predictive benefits of Ridge Regression
      - Disadvantages:
        - ◆ Arbitrarily rules out some correlated variables like Lasso
        - ◆ Underestimates coefficients of very predictive variables like Ridge Regression
    - Ridge Regression - constrain squares of coefficients (doesn't do variable selection but can lead to better predictive models)
      - Coefficients shrink toward 0 to reduce variance in estimate
      - Ridge regression will choose smaller (in an absolute sense) non-zero coefficients for both models. By nature, it may underestimate the effect of the factors.
- 11.3 Choosing a Variable Selection Model
- 12.1 Introduction to Design of Experiments
  - Which ad campaign to run? Which product to show?
  - Survey
  - Comparison and Control
  - Blocking - something that could cause variation (sports cars more likely to be red)
- 12.2 A/B Testing
  - Choose between two alternatives
    - Collect data quickly
    - Data must be representative
    - Amount of data is small compared to the whole population
- 12.3 Factorial Designs
  - Full factorial design
    - 2 fonts \* 2 wordings \* 2 backgrounds = 8 combinations
    - ANOVA (analysis of variance)
    - Can add up quickly if 7 factors each with 3 choices =  $3^7 = 2,187$  combinations
  - Fractional factorial design

# Factorial design

## Fractional factorial design

- Test subset of combinations
- Balanced design
  - Test each choice the same # of times
  - Test each pair of choices the same # of times



### Full Factorial Design

Font	Wording	Background
Arial	MS in Analytics	White
Arial	MS in Analytics	Gold
Arial	Master of Science in Analytics	White
Arial	Master of Science in Analytics	Gold
Roboto	MS in Analytics	White
Roboto	MS in Analytics	Gold
Roboto	Master of Science in Analytics	White
Roboto	Master of Science in Analytics	Gold

### Fractional Factorial Design

Font	Wording	Background
Arial	MS in Analytics	White
Arial	Master of Science in Analytics	Gold
Roboto	MS in Analytics	Gold
Roboto	Master of Science in Analytics	White

- Independent Factors
  - Test subset of combinations
  - Use regression to estimate effects
    - Ex: background color (gold, blue, white) and font size (10,12,14)

## Independent Factors

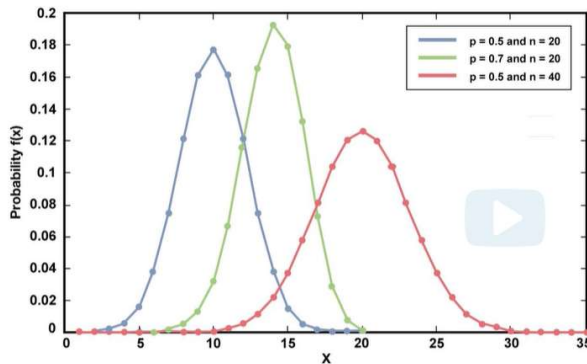
- Test subset of combinations
- Use regression to estimate effects
- Example:
  - Background color (gold, blue, white)
  - Font size (10, 12, 14 point)



GTx

- Factorial design summary
  - Use before collecting data
  - Determine effects of factors
  - Full factorial design
    - Test all combinations
  - Partial factorial design
    - Estimate all effects by comparing some combinations
- 12.4 Multi-Armed Bandits
  - Exploration vs. Exploitation
    - Multi-armed bandit models use the best answer (exploitation) the more they're sure it's best. If the model is less sure what's best, it's more likely to concentrate on trying many options (exploration)
  - Start testing with k alternatives
    - Update probabilities with new information from testing
  - Help you learn faster on the fly and create more value along the way
- 13.1 Introduction to Advance Probability Distributions
- 13.2 Bernoulli, Binomial and Geometric Distribution
  - Bernoulli - probability mass Function  $P(X=1) = p$

## Binomial Distribution



Probability Mass Function:

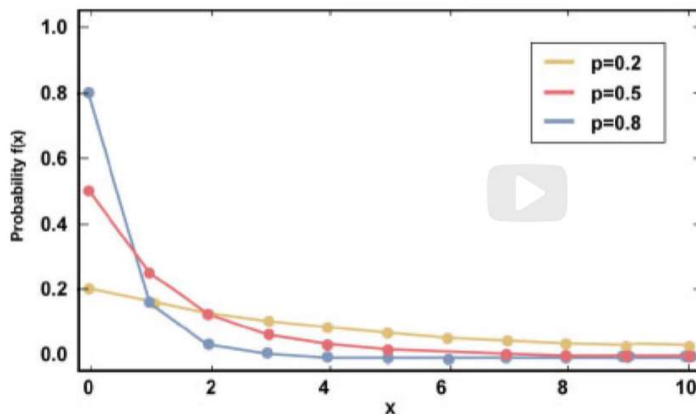
$$P(X = x) = \binom{n}{x} p^x (1-p)^{n-x}$$

$$= \left( \frac{n!}{(n-x)!x!} \right) p^x (1-p)^{n-x}$$

- Probability of getting  $x$  successes out of  $n$  independent identically distributed Bernoulli( $p$ ) trials
- Large  $n$ 
  - Binomial distribution converges to Normal distribution

GTx

## Geometric Distribution



Probability Mass Function:

$$P(X = x) = (1-p)^x p$$

- Probability of having  $x$  Bernoulli( $p$ ) failures until first success?
  - Or, having  $x$  Bernoulli( $1-p$ ) successes until first failure

GTx

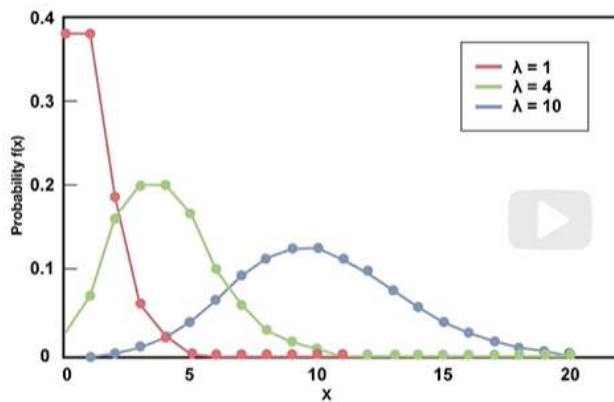
- According to the geometric distribution, the probability of having 5 successful sales calls before the first unsuccessful call is  $(1-p)^5 p$ .
  - According to the geometric distribution, the probability of having 5 successful sales calls before the first unsuccessful call is  $p^5 (1-p)$ .
- 13.3 Poisson, Exponential and Weibull Distributions
  - Won't cover (do on your own)
    - Distribution's probability, mass index, density function, expectation, variance

## Poisson Distribution



Probability Mass Function:

# Poisson Distribution



Probability Mass Function:

$$f_x(x) = \frac{\lambda^x e^{-\lambda}}{x!}$$

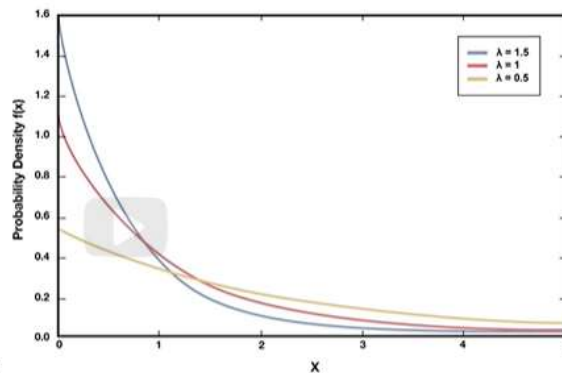
- Good at modeling random arrivals
- $\lambda$  – Average number of arrivals / time period
- Arrivals are independent and identically distributed (i.i.d)

GT<sub>x</sub>

# Exponential Distribution

Probability Mass Function:

$$f_x(x) = \lambda e^{-\lambda x}$$

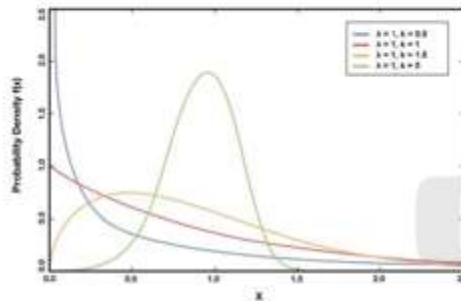


- Relation to Poisson
  - If arrivals are Poisson( $\lambda$ ), then time between successive arrivals is exponential( $\lambda$ ) distribution

Poisson arrivals  $\Leftrightarrow$  exponential inter-arrival time

GT<sub>x</sub>

# Weibull Distribution



Probability Mass Function:  
Scale parameter ( $\lambda$ ),  
Shape parameter ( $k$ )

$$f_x(x) = \frac{k}{\lambda} \left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k}$$



- Weibull: time between failures
- Geometric: number of tries between failures
- Lightbulb example
  - How many lightswitch flips on/off until bulb fails? (Geometric)
  - Leave the bulb on; how long until bulb fails? (Weibull)

GTx

## Weibull Distribution

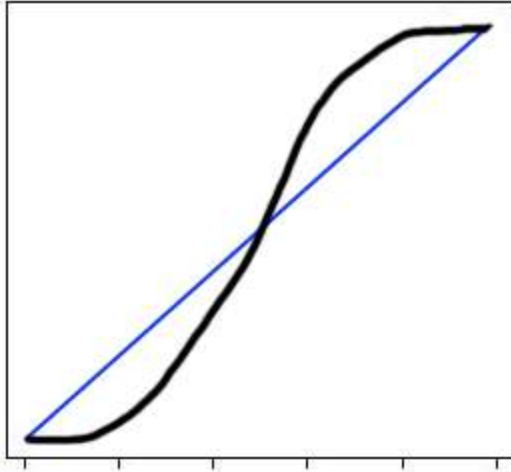
- $k < 1$ 
  - Modeling when failure rate *decreases* with time
  - "Worst things fail first" (ex: parts with defects)
- $k > 1$ 
  - Modeling when failure rate *increases* with time
  - "Things that wear out" (ex: tires)
- $k = 1$ 
  - Modeling when failure rate is *constant* with time



GTx

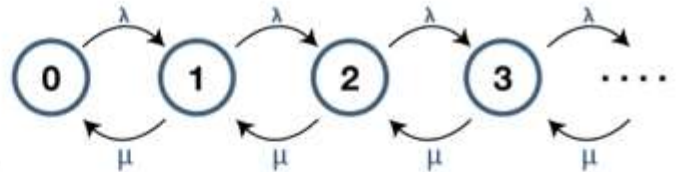
- 13.3 Poisson, Exponential and Weibull Distributions
  - Here's how these two are related. If arrivals are Poisson, with arrival rate  $\lambda$ , then the time between arrivals, known as the inter-arrival time follows the Exponential distribution, with one over  $\lambda$  as the average inter-arrival time.
  - If times between arrivals are exponentially distributed with average inter-arrival time one over  $\lambda$ , then the arrivals follow the Poisson distribution with an average of  $\lambda$  arrivals per unit time.

The geometric distribution models how many tries it takes for something to happen, while the Weibull distribution models how long it takes.
- 13.4 QQ Plots



- Heavy tailed distribution
- 13.5 Queuing

## Queuing Example



- Arrival Rate (calls) =  $\lambda$
- Service Rate (calls) =  $\mu > \lambda$
- Transition Equations ( $\geq 1$  calls in the queue)
  - $P(\text{Next event is an arrival}) = \frac{\lambda}{\lambda + \mu}$
  - $P(\text{Next event is finished call}) = \frac{\mu}{\lambda + \mu}$
- Can calculate:
  - Expected fraction of time employee is busy =  $\frac{\lambda}{\mu}$
  - Expected waiting time before talking to employee =  $\frac{\lambda}{\mu(\mu - \lambda)}$
  - Expected number of calls waiting in queue =  $\frac{\lambda^2}{\mu(\mu - \lambda)}$



# Memoryless Property

- Memoryless exponential distribution
  - Distribution of *remaining* call time = *initial* distribution of call time
- Memoryless Poisson distribution
  - Distribution of time to next arrival = initial distribution of time to next arrival
  - Poisson interarrival times are exponentially distributed
- Data fits exponential distribution → memoryless
- Not memoryless → not exponential

GT<sub>x</sub>

## Memoryless Property

- Law firm example
  - Should tire manufacturer pay damages for accident that happened when tire with 10,000 miles failed?
- Probability(tire fails at 10,000 miles) = ?
  - Tires are more likely to fail, the more worn out they are
  - Not memoryless
  - Cannot model with the exponential distribution
  - Maybe try Weibull with  $k > 1$



GT<sub>x</sub>

# Queuing Models

- Potential queuing model parameters
  - General arrival distribution [A]
  - General service distribution [S]
  - Number of servers [c]
  - Size of the queue [K]
  - Population size [N]
  - Queuing discipline [D]
- Kendall notation (e.g. "M/M/1 queue")
- Model extensions: potential "hang-ups", balking, etc.



- Memoryless Property Distribution
- Distribution of remaining call time = initial distribution of call time
- 13.6 Simulation Basics
  - **Deterministic simulations** do not incorporate elements of randomness and so their output for given inputs does not vary.
  - **Stochastic simulations** incorporate randomness and thus might produce different outputs. They are generally more useful in analytics. Stochastic simulations must be replicated because one result may not be characteristic.
  - In **continuous-time simulations**, changes can happen continuously. In **discrete event simulations**, changes only happen at discrete time points when something happens.

## Simulation

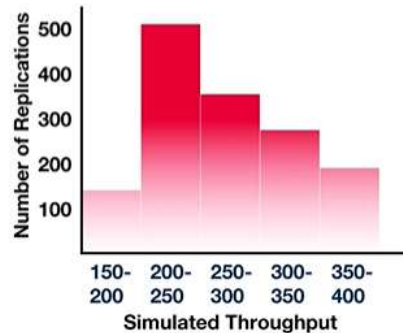
- Simulation software
  - Elements of model include
    - Entities: things that move through simulation (e.g., bags, people, etc.)
    - Modules: parts of process (e.g., queues, storage, etc.)
    - Actions
    - Resources (e.g., workers)
    - Decision points
    - Statistical tracking
    - Etc.
  - Often "drag and drop" programming style





# Simulation

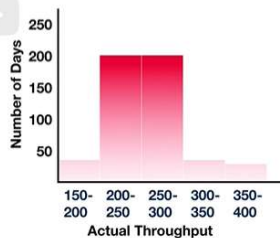
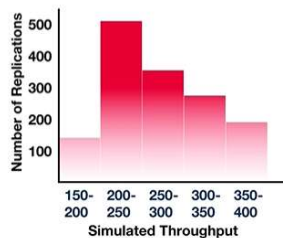
- Replications: number of runs of simulation
  - One replication = one data point (may be unrepresentative)
  - Run multiple times to get distribution of outcomes
  - Example: simulating average daily throughput



GTx

## Simulation

- Simulation Validation
  - Use real data to validate your simulation is giving reasonable results
    - Real and simulated averages don't match → problem
    - Averages match, variances don't match → problem



GTx

- Stochastic simulation should be run many times because one random outcome might not be representative of system performance in the range of different situations that could arise
- A stochastic simulation is meant to show the performance of a system over a range of random events that could happen.
- If the simulation isn't a good reflection of reality, then any insights we gain from studying the simulation might not be applicable in reality.
- This is such an important point about simulation that I wanted to make sure you all clicked on it. I've overseen a lot of analytics projects that included simulation, and my experience is that it's easy to rely too much on simulated insights that might not be true in reality. It's critical to make sure that the simulation is a good-enough model of reality that insights from the simulation can effectively be transferred to reality
- 13.7 Prescriptive Simulation
- 13.8 Markov Chains
  - The next state of a 'memoryless' process doesn't depend on previous states, but it does depend on the current state

# Markov Chains

- Long-run probability of rainy days?

- Calculate  $((\pi P)P)P \dots = \pi P^\infty$  ?
- Instead, use "steady state"
  - Apply P, and get initial vector back:  
 $\pi^* P = \pi^*$
- Solve for such a  $\pi^*$  vector
  - $\pi^* P = \pi^*$ , and  $\sum_i \pi_i^* = 1$

P	Sunny	Cloudy	Rainy
Sunny	.75	.15	.10
Cloudy	.20	.40	.40
Rainy	.40	.30	.30

- $\pi^*$  might not always exist
  - Can't have cyclic behavior
  - Every state must be reachable from all others

GTx

- 14.1 Missing Data
- 14.2 Methods That Do Not Require Imputation
  - Two different ways dealing with missing data
    - Remove
    - Add categorical variables
  - Avoid estimating what missing data may be

## Categorical variable approach

Quantitative variable has missing data

- All missing values = 0
- New categorical variable
  - Missing data could be biased
  - Include interactions

ID#	Height	Missing?	# kids	#kids,missing
1	5'6"	No	2	2
2	6'1"	No	3	3
3	5'11"	No	5	5
4	0	Yes	4	0
5	5'10"	No	1	1
6	0	Yes	5	0

Split into two models

Height data			No height data	
ID#	Height	# kids	ID#	# kids
1	5'6"	2	4	4
2	6'1"	3	6	5
3	5'11"	5		
5	5'10"	1		

GTx

- 14.3 Imputation Methods
  - Imputation - estimating missing values
    - Mean, median (numeric) or mode (categorical)
      - Advantage
        - Hedge against being too wrong
        - Easy to compute
      - Biased imputation
        - Ex: people with high income less likely to answer survey
        - Mean/median will underestimate the missing value
    - Second approach: regression

- True! It's usually not recommended to impute more than 5% of values, and advanced methods like multivariate imputation by chained equations (MICE) can impute multiple factor values together

## Imputation Approaches

- Second approach: regression
  - Reduce or eliminate the problem of bias
  - Income example
    - Factors: zip code, profession, number of cars owned, etc.

- Disadvantages
  - Complex: build, fit, validate, test to estimate missing value
  - Does not capture all the variability

Income	Zip code	Profession	Cars Owned
\$15,000	24001	Fast food server	0
\$31,000	24330	Telemarketer	1
\$59,000	17330	Executive Assistant	1
\$98,000	24335	Plumber	1
\$120,000	30228	Car Salesman	2
\$240,000	34509	Financial Advisor	3
\$403,000	67767	Dentist	4

GTx

## Imputation with variability

- Imputation does not capture all variability
  - Even with regression model
- Impute with added variability
  - Add perturbation to each imputed value
    - Ex: normally-distributed variation
    - Less accurate on average
    - More-accurate variability



GTx

- 15.1 Introduction to Optimization
  - Regression classification, etc.
    - Software automates solution and model building
  - Optimization
    - Software automates solution
    - Model building is up to you!
  - Statistical software can both build and solve regression models. Optimization software only solves models; human experts are required to build optimization models.

## Optimization for prescriptive analytics (examples)

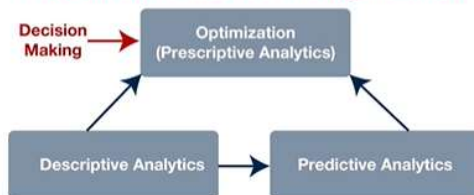
- Airplane mechanic scheduling
- Crude oil shipment planning



# Optimization for prescriptive analytics (examples)

- Airplane mechanic scheduling
- Crude oil shipment planning
- Server farm allocation
- Machine shop production
- GPS routing for cars

Provides direction for an organization



- 15.2 Elements of Optimization Models
  - Variables - decisions to be made
  - Constraints - restrictions on variable values
  - Objectives function - solution quality measure
  - Solution - values for each variables
  - Feasible solution - variable values that satisfy all constraints
  - Optimal solution - feasible solution with the best objective value

## Optimization Models

Variables – decisions to be made

Constraints – restrictions on variable values

Objective function

Example: political candidate scheduling

$x_i$  = total time spent in state  $i$

$y_i$  = number of visits to state  $i$

$z_i$  = 1 if state  $i$  is ever visited, 0 if not

$w_{id}$  = time spent in state  $i$  on day  $d$

$v_{id}$  = 1 if state  $i$  visited on day  $d$ , 0 if not

Constraints:

30 days left in campaign

$$\sum_i x_i \leq 30$$

At least 3 Florida visits in days 24-30

$$\sum_{d=24}^{30} v_{\text{Florida},d} \geq 3$$

Total visits must add up correctly

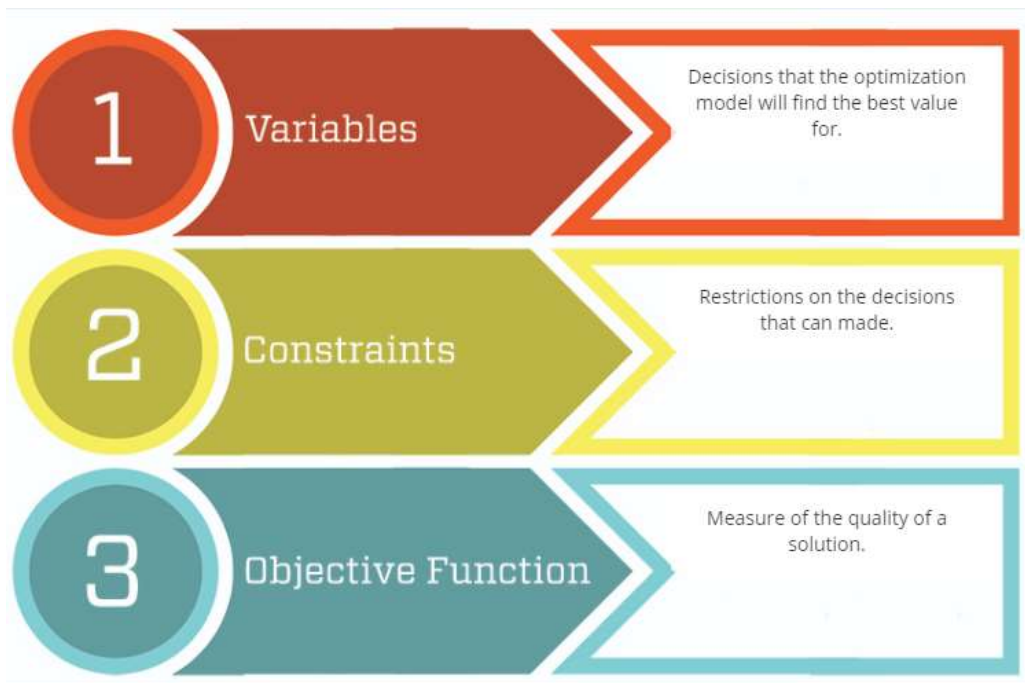
$$\sum_d v_{id} = y_i$$

Objective function

Maximize expected new votes

$$\sum_i \left( \alpha p_i \sqrt{x_i + \frac{1}{3} \sum_{j \in N(i)} x_j} + \beta v_{id} f_d \right)$$

GTx



- 15.3 Modeling is an Art: Two Examples

## Example 1

- Diet problem (US Army)  
Satisfy soldiers' nutritional requirements...  
...at minimum cost

$n$  foods  
 $m$  nutrients

$a_{ij}$  = amount of nutrient  $j$  per unit of food  $i$

$m_j$  = minimum daily intake of nutrient  $j$   
 $M_j$  = maximum daily intake of nutrient  $j$

$c_i$  = per-unit cost of food  $i$

### Optimization model

#### Variables

$x_i$  = amount of food  $i$   
in daily diet

#### Constraints

$\sum_i a_{ij}x_i \geq m_j$  for each nutrient  $j$

$\sum_i a_{ij}x_i \leq M_j$  for each nutrient  $j$

$x_i \geq 0$  for each food  $i$

#### Objective function

Minimize  $\sum_i c_i x_i$

GT\*



## Example 2

- Call center scheduling
  - Meet forecasted demand  $d_i$  for each day of the week  $i$
  - Workers work 5 days in a row, then 2 days off
  - Minimize worker-days used

### Variables

$x_i$  = number of people who start working on day  $i$

### Objective function

minimize  $5(x_{\text{Sunday}} + x_{\text{Monday}} + \dots + x_{\text{Saturday}})$

### Constraints

Meet demand:  $\sum_j \text{working on day } i \cdot x_j \geq d_i$

Ex:  $x_{\text{Fri}} + x_{\text{Sat}} + x_{\text{Sun}} + x_{\text{Mon}} + x_{\text{Tue}} \geq d_{\text{Tue}}$

**Non-negativity:**  $x_i \geq 0$  for all days  $i$

**Integrality:**  $x_i$  is integer for all days  $i$



- 15.4 Modeling with Binary Variables

## Example: Stock Market Investment

Invest to balance return and risk

$B$  = investment budget

$n$  = number of stocks available

$r_i$  = expected return of stock  $i$  relative to market

$Q_{ij}$  = covariance of returns of stocks  $i$  and  $j$

### Variables

$x_i$  = amount invested in stock  $i$

### Constraints

$$\sum_i x_i \leq B$$

$x_i \geq 0$  for all stocks  $i$

### Objective function

$$\text{Maximize } \sum_i r_i x_i - \theta \sum_i \sum_j Q_{ij} x_i x_j$$

Expected  
return

Risk

GTx



## Example: Stock Market Investment

Invest to balance return and risk

B = investment budget

n = number of stocks available

$r_i$  = expected return of stock i relative to market

$Q_{ij}$  = covariance of returns of stocks i and j

Linking constraints

Transaction fees

Fixed charge

Could be  $t_i$  if vary by stock

Can be weighted as one-time expense

Variables

$x_i$  = amount invested in stock i

$y_i = 1$  if invest in stock i, 0 if not

Constraints

$$\sum_i x_i \leq B$$

$x_i \geq 0$  for all stocks i

$x_i \leq B y_i$  for all stocks i

$y_i = 0$  means  $x_i \leq 0$  (and  $x_i \geq 0$  forces  $x_i = 0$ )

$y_i = 1$  means  $x_i \leq B$

Objective function

$$\text{Maximize } \sum_i r_i x_i - \theta \sum_i \sum_j Q_{ij} x_i x_j - \sum_i t y_i$$

GTx

## Example: Stock Market Investment

Invest to balance return and risk

B = investment budget

n = number of stocks available

$r_i$  = expected return of stock i relative to market

$Q_{ij}$  = covariance of returns of stocks i and j

$m_i$  = minimum dollar amount for each stock

Personal constraints (examples)

Invest in Tesla

Invest in Amazon, Google, or Apple

Invest in both or neither of FedEx, UPS

Invest in exactly one of Coca-Cola and

PepsiCo

Variables

$x_i$  = amount invested in stock i

$y_i = 1$  if invest in stock i, 0 if not

Constraints

$$\sum_i x_i \leq B$$

$x_i \geq 0$  for all stocks i

$x_i \leq B y_i$  for all stocks i

$x_i \geq m_i y_i$  for all stocks i

$y_{\text{Tesla}} = 1$

$y_{\text{Amazon}} + y_{\text{Google}} + y_{\text{Apple}} \geq 1$

$y_{\text{FedEx}} = y_{\text{UPS}}$

$y_{\text{Coca-Cola}} = 1 - y_{\text{PepsiCo}}$

Objective function

$$\text{Maximize } \sum_i r_i x_i - \theta \sum_i \sum_j Q_{ij} x_i x_j - \sum_i t y_i$$

GTx

## Example: Stock Market Investment

Invest to balance return and risk

B = investment budget

n = number of stocks available

$r_i$  = expected return of stock i relative to market

$Q_{ij}$  = covariance of returns of stocks i and j

$m_i$  = minimum dollar amount for each stock

Personal constraints (more examples)

If invest in energy, invest in at least 5

Option 1:  $\sum_{j \in \text{energy}} y_j \geq 5 y_i$  for all energy stocks i

Variables

$x_i$  = amount invested in stock i

$y_i = 1$  if invest in stock i, 0 if not

Constraints

$$\sum_i x_i \leq B$$

$x_i \geq 0$  for all stocks i

$x_i \leq B y_i$  for all stocks i

$x_i \geq m_i y_i$  for all stocks i

$y_{\text{Tesla}} = 1$

$y_{\text{Amazon}} + y_{\text{Google}} + y_{\text{Apple}} \geq 1$

$y_{\text{FedEx}} = y_{\text{UPS}}$

$y_{\text{Coca-Cola}} = 1 - y_{\text{PepsiCo}}$

Objective function

$$\text{Maximize } \sum_i r_i x_i - \theta \sum_i \sum_j Q_{ij} x_i x_j - \sum_i t y_i$$

GTx

## Example: Stock Market Investment

Invest to balance return and risk

$B$  = investment budget

$n$  = number of stocks available

$r_i$  = expected return of stock  $i$  relative to market

$Q_{ij}$  = covariance of returns of stocks  $i$  and  $j$

$m_i$  = minimum dollar amount for each stock

Personal constraints (more examples)  
If invest in energy, invest in at least 5

Option 2:  $z_{\text{Energy}} = 1$  if invest in energy, 0 if not

$$\sum_{j \in \text{energy}} y_j \geq 5z_{\text{Energy}}$$

$$z_{\text{Energy}} \geq y_i \text{ for all energy stocks } i$$

Variables

$x_i$  = amount invested in stock  $i$

$y_i = 1$  if invest in stock  $i$ , 0 if not

Constraints

$$\sum_i x_i \leq B$$

$x_i \geq 0$  for all stocks  $i$

$x_i \leq B y_i$  for all stocks  $i$

$x_i \geq m_i y_i$  for all stocks  $i$

$$y_{\text{Tesla}} = 1$$

$$y_{\text{Amazon}} + y_{\text{Google}} + y_{\text{Apple}} \geq 1$$

$$y_{\text{FedEx}} = y_{\text{UPS}}$$

$$y_{\text{Coca-Cola}} = 1 - y_{\text{PepsiCo}}$$

Objective function

$$\text{Maximize } \sum_i r_i x_i - \theta \sum_i \sum_j Q_{ij} x_i x_j - \sum_i t y_i$$

GTx

## Integer variables in optimization

- Fixed charges in objective function
- Constraints to choose among options
- Constraints requiring same/opposite decisions
- If-then constraints

Optimization modeling is an art!



GTx

- 15.5 Optimization for Statistical Models

## Linear Regression Model

Variables

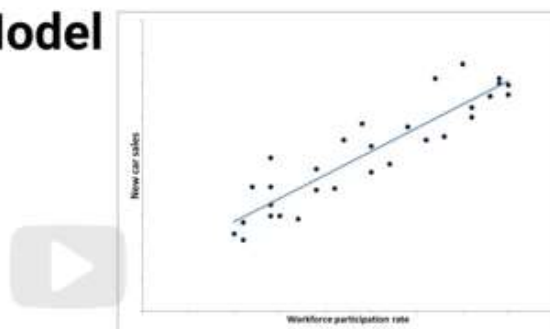
- $a_0, a_1 \dots a_m$

Constraints

- none

Objective function

- Minimize  $\sum_{i=1}^n (y_i - (a_0 + \sum_{j=1}^m a_j x_{ij}))^2$



Given  $n$  data points

- $x_{ij} = j$ th factor for data point  $i$
- $y_i =$  response for data point  $i$

Find coefficients  $a_0, a_1 \dots a_m$  to best fit data

GTx

# Notation confusion: statistics vs. optimization

## Statistical model for regression

$$\hat{y}_i = a_0 + \sum_{j=1}^m a_j x_{ij}$$

$a_j$  are constant coefficients  
 $x_{ij}$  are variables

## Optimization model for regression

Minimize  $\sum_{i=1}^n \left( y_i - \left( a_0 + \sum_{j=1}^m a_j x_{ij} \right) \right)^2$

$a_j$  are variables  
 $x_{ij}$  are constant coefficients

GTx

## Other regression models

Linear regression (no constraints)

Variables  $a_0, a_1 \dots a_m$

Objective Minimize  $\sum_{i=1}^n \left( y_i - \left( a_0 + \sum_{j=1}^m a_j x_{ij} \right) \right)^2$

Lasso regression constraint

$$\sum_{j=1}^m |a_j| \leq T$$

Ridge regression constraint

$$\sum_{j=1}^m (a_j)^2 \leq T$$

Elastic net constraint

$$\lambda \sum_{j=1}^m |a_j| + (1 - \lambda) \sum_{j=1}^m (a_j)^2 \leq T$$



GTx

# Logistic Regression

Variables

- $a_0, a_1 \dots a_m$

Constraints

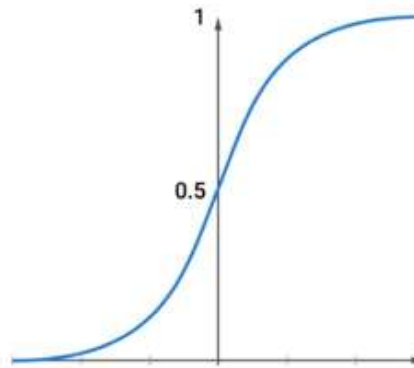
- none

Objective function

- Maximize

$$\prod_{i:y_i=1} p(x_i) \prod_{i:y_i=0} (1 - p(x_i))$$

$$\text{where } p(x_i) = \frac{1}{1 + e^{-(a_0 + \sum_{j=1}^m a_j x_{ij})}}$$



Given  $n$  data points

- $x_{ij}$  =  $j$ th factor for data point  $i$
- $y_i$  = response for data point  $i$

Find coefficients  $a_0, a_1 \dots a_m$  to best fit data

GTx

## Support vector machine models

Hard classification

Variables

- $a_0, a_1 \dots a_m$

Constraints

- $(a_0 + \sum_{j=1}^m a_j x_{ij}) y_i \geq 1$  for each  $i$

Objective function

- Maximize  $\sum_{j=1}^m (a_j)^2$

Soft classification

Variables

- $a_0, a_1 \dots a_m$

Constraints

- none

Objective function

- Minimize  $\sum_{i=1}^n \max\{0, 1 - (a_0 + \sum_{j=1}^m a_j x_{ij}) y_i\} + \lambda \sum_{j=1}^m (a_j)^2$

GTx

# Time series models

## Exponential smoothing

### Variables

- $\alpha, \beta, \gamma$

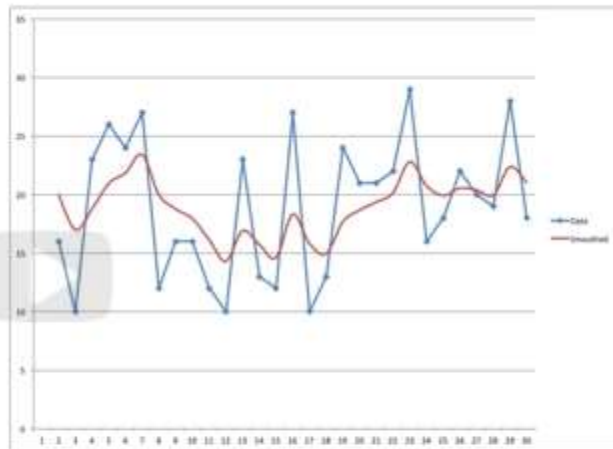
### Constraints

- $0 \leq \alpha \leq 1$
- $0 \leq \beta \leq 1$
- $0 \leq \gamma \leq 1$

### Objective function

- Minimize

$$\sum_{t=1}^n (x_t - \hat{x}_t)^2$$



GT<sub>x</sub>

# Time series models

## ARIMA

### Variables

- $\mu, \varphi_i, \theta_i$

### Constraints

- none

### Objective function

- Minimize

$$\sum_{t=1}^n (x_t - \hat{x}_t)^2$$

where

$$D_{(d)t} = \mu + \sum_{i=1}^p \alpha_i D_{(d)t-i} - \sum_{i=1}^q \theta_i (\hat{x}_{t-i} - x_{t-i})$$

## GARCH

### Variables

- $\omega, \beta_i, \gamma_i$

### Constraints

- none

### Objective function

- Minimize

$$\sum_{t=1}^n (\sigma_t^2 - \hat{\sigma}_t^2)^2$$

where

$$\hat{\sigma}_t^2 = \omega + \sum_{i=1}^q \beta_i \sigma_{t-i}^2 - \sum_{i=1}^p \gamma_i \epsilon_{t-i}^2$$

GT<sub>x</sub>



# k-means for clustering

## Variables

- $z_{jk}$  (coordinate  $j$  of cluster center  $k$ )
- $y_{ik}$  (1 if point  $i$  in cluster  $k$ , 0 if not)

## Constraints

- $\sum_k y_{ik} = 1$  for all data points  $i$   
(each data point assigned to a cluster)

## Objective function

- Minimize  $\sum_i \sum_k y_{ik} \sqrt{\sum_j (x_{ij} - z_{jk})^2}$   
(minimize total distance from data points to their cluster centers)

Given data

- $x_{ij}$  = coordinate  $j$  of data point  $i$   
(value of  $j$ th attribute of data point  $i$ )



GTx

## Summary

- Some optimization models that underlie statistical models

## Future

- Classification of optimization models
- Which solve faster?



GTx

- All of the following have optimization models to find best fit:
  - Linear regression
  - Logistic regression
  - Lasso regression
  - Exponential smoothing
  - k-means clustering
- 15.6 Classification of Optimization Models



# Optimization Models

$x$  = vector of variables

Minimize or maximize objective function

Minimize  $f(x)$  or Maximize  $f(x)$

Variables  $x$  must belong to a set  $X$

Subject to  $x \in X$



GTx

## Linear program

- $f(x)$  is a linear function

Minimize or Maximize  $C + \sum_{i=1}^n c_i x_i$

- Constraint set  $X$  is defined by linear equations and inequalities



$\sum_{i=1}^n a_{ij} x_i \leq b_j$  or  
 $\sum_{i=1}^n a_{ij} x_i = b_j$  or  
 $\sum_{i=1}^n a_{ij} x_i \geq b_j$   
for each constraint  $j$

Easy/fast to solve, even for very large instances

GTx

# Convex quadratic program

- $f(x)$  is a convex quadratic function

Minimize  $f(x)$  or Maximize  $-f(x)$

- Constraint set  $X$  is defined by linear equations and inequalities

Easy/fast to solve, but  
not as quickly as linear programs



# Optimization Model

Convex Optimization Program

- Objective function  $f(x)$  is
  - Concave (if maximizing)
  - Convex (if minimizing)
- Constraint set  $X$  is a convex set
- Easy to solve, but solutions can take a lot longer



# Optimization Model

## Convex Optimization Program

- Objective function  $f(x)$  is
  - Concave (if maximizing)
  - Convex (if minimizing)
- Constraint set  $X$  is a convex set
- Easy to solve, but solutions can take a lot longer

## Integer Program

- Linear program, plus
- Some (or all) variables restricted to take only integer values
  - Variables could be binary
    - Either 0 or 1
- More difficult to solve even with good software packages

## General Non-Convex Program

- Optimization problem is not convex
  - Hard to find optimal solutions

GT<sub>x</sub>

# Optimization Problem Types

From quickest/easiest to slowest/hardest


1. Linear Programs
2. Convex Quadratic Programs
3. Convex Programs
4. Integer Programs
5. General Non-Convex Programs


What if our problem is too hard?

- Heuristic
  - Rule-of-thumb process
  - Usually gives good solutions

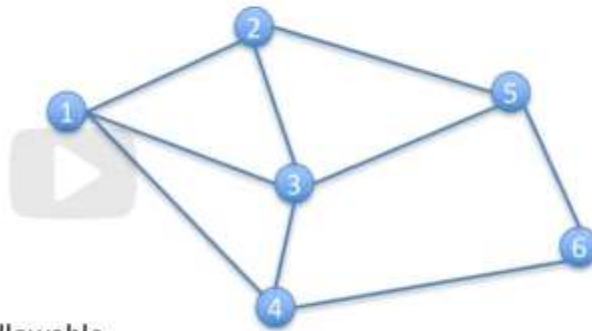


# Network Models (type of linear program)

 Location (node/vertex)

 Connection (arc/edge)

$x_{ij}$  : variable for arc from  $i$  to  $j$   
(how much flow there is)



## Constraints

- Flow into node = flow out of node
- Flow on arc between min and max allowable

## Objective function

- Linear function of variables

If all data is integer, then  
all optimal variable values will  
automatically be integer too!

GT<sub>x</sub>

## Common network models

### Shortest path model

- Find quickest/shortest route from one place to another
- E.g., Google Maps, GPS

### Assignment model

- E.g., which worker gets which job to maximize workforce efficiency?

### Maximum flow model

- E.g., how much oil can flow through complex network of pipes?

GT<sub>x</sub>

- Adding integer variables moves the model from a linear program, which usually solves very quickly, to an integer program, which sometimes takes a long time to solve.
- 15.7 Stochastic Optimization



# Optimization

What if data or parameter isn't known exactly?  
What if forecast values aren't known exactly?

## Model conservatively

Ex: Call center worker constraint

$$x_{\text{Fri}} + x_{\text{Sat}} + x_{\text{Sun}} + x_{\text{Mon}} + x_{\text{Tue}} \geq d_{\text{Tue}} + \theta$$

$x_i$  = number of workers starting 5-day shift on day  $i$

$d_i$  = expected demand on day  $i$

$\theta$  = extra workers just in case



GT<sub>x</sub>

# Optimization

What if data or parameter isn't known exactly?  
What if forecast values aren't known exactly?

## Scenario modeling

**Scenario 1** : two small recurring bugs

**Scenario 2** : one major bug 3 days after launch

**Scenario 3** : two major, immediate bugs

**Scenario 4** : catastrophic scenario after  
10,000 signups

Etc.

$x_i$  = number of workers starting 5-day shift on day  $i$

$d_{is}$  = expected demand on day  $i$  in scenario  $s$



GT<sub>x</sub>

# Optimization

What if data or parameter isn't known exactly?  
What if forecast values aren't known exactly?

## Scenario modeling

- Scenario 1** : two small recurring bugs
- Scenario 2** : one major bug 3 days after launch
- Scenario 3** : two major, immediate bugs
- Scenario 4** : catastrophic scenario after 10,000 signups



## Robust solution

$$\left. \begin{aligned} x_{\text{Fri}} + x_{\text{Sat}} + x_{\text{Sun}} + x_{\text{Mon}} + x_{\text{Tue}} &\geq d_{\text{Tue},1} \\ x_{\text{Fri}} + x_{\text{Sat}} + x_{\text{Sun}} + x_{\text{Mon}} + x_{\text{Tue}} &\geq d_{\text{Tue},2} \\ x_{\text{Fri}} + x_{\text{Sat}} + x_{\text{Sun}} + x_{\text{Mon}} + x_{\text{Tue}} &\geq d_{\text{Tue},3} \\ x_{\text{Fri}} + x_{\text{Sat}} + x_{\text{Sun}} + x_{\text{Mon}} + x_{\text{Tue}} &\geq d_{\text{Tue},4} \end{aligned} \right\} \text{Satisfy all scenario demands}$$

$x_i$  = number of workers starting 5-day shift on day  $i$   
 $d_{is}$  = expected demand on day  $i$  in scenario  $s$

GTx

# Optimization

What if data or parameter isn't known exactly?  
What if forecast values aren't known exactly?

## Scenario modeling

- Scenario 1** : two small recurring bugs
- Scenario 2** : one major bug 3 days after launch
- Scenario 3** : two major, immediate bugs
- Scenario 4** : catastrophic scenario after 10,000 signups



## Optimize expected cost

minimize  $5(x_{\text{Sun}} + x_{\text{Mon}} + \dots + x_{\text{Sat}}) + \sum_s c(p_{\text{Sun},s}y_{\text{Sun},s} + p_{\text{Mon},s}y_{\text{Mon},s} + \dots + p_{\text{Sat},s}y_{\text{Sat},s})$   
 Example constraints:  $x_{\text{Fri}} + x_{\text{Sat}} + x_{\text{Sun}} + x_{\text{Mon}} + x_{\text{Tue}} + y_{\text{Tue},1} \geq d_{\text{Tue},1}$  and  $y_{\text{Tue},1} \geq 0$

- $x_i$  = number of workers starting 5-day shift on day  $i$
- $d_{is}$  = expected demand on day  $i$  in scenario  $s$
- $c$  = cost for each worker below demand level
- $y_{is}$  = expected worker shortfall on day  $i$  in scenario  $s$
- $p_{is}$  = probability of scenario  $s$  occurring

GTx



# Mathematical Programming Models

Variables, Constraints, objective function

*Other models have different structure*

## Dynamic program

- States (the exact situations, and their values)
- Decisions (choices of next state)
- Bellman's equation: determine optimal decisions

## Stochastic dynamic program

- Dynamic program, but decisions have **probabilities** of next state

## Markov decision process

- Stochastic dynamic program with Discrete states and decisions
- Probabilities depend only on current state/decision

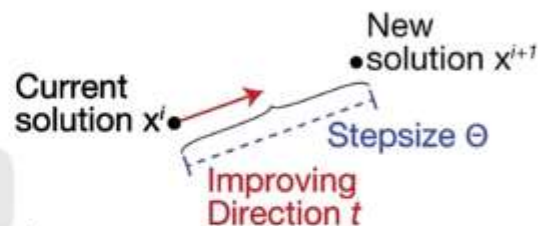


- Optimization models treat all of the data as known exactly.
- 15.8 Basic Optimization Algorithms

## Solving an Optimization Model

Two main steps

- Create a first solution
  - Can be simple/bad/infeasible
- Repeat
  - Find an **improving direction  $t$**
  - Using a **step size  $\Theta$**  to move along it
  - **New solution = old solution +  $\Theta t$**
- Stop when solution doesn't change much or time runs out



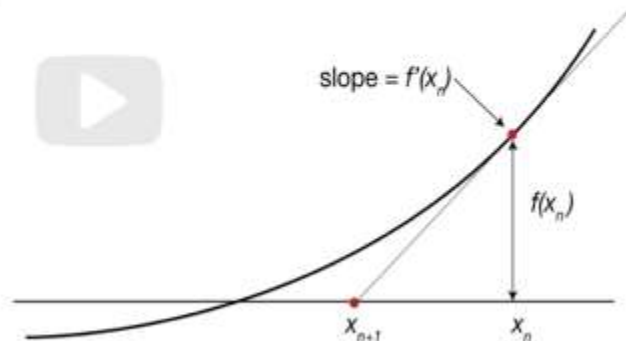
GTx

# Algorithm from Calculus

- Newton's method: finding root of  $f(x)$
- Current solution  $x_n$  at step  $n$

$$x_{n+1} = x_n - 1 \frac{f(x_n)}{f'(x_n)}$$

Step size      Improving direction



GTx

## Convex Optimization Problem

### Convex optimization problem

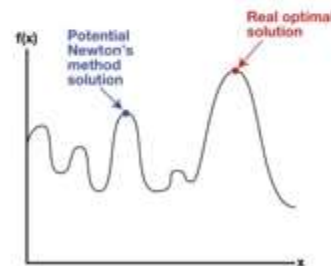
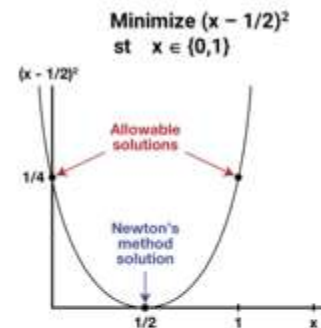
- Guaranteed to find optimal solution

### Non-convex optimization problem

- Not guaranteed to find optimal solution
  - Ex: converge to infeasible solution
  - Ex: converge to local optimum

### Running time

- Integer programs: can be long
- Linear programs: often fast



GTx

- The two main steps of most optimization algorithms are:
  - Find a good direction to move from the current solution
  - And determine how far to go in that direction.
  - Many optimization algorithm follow the pattern of finding an improving direction and a step size, make the move, and repeat.
- 16.1 Non-Parametric Methods

## McNemar's Test – Example

- Two competing treatments for a virus
  - A: successful on 61/100
  - B: successful on 68/100



### Scenario 1

- 32 cases: neither worked
  - 61 cases: both A and B worked
  - 7 cases: B worked, A did not
  - 0 cases: A worked, B did not
  - Conclude that B is better
- $p=0.02$  (accept)



### Scenario 2

- 12 cases: neither worked
  - 41 cases: both A and B worked
  - 27 cases: B worked, A did not
  - 20 cases: A worked, B did not
  - Conclude that B is *not* better
- $p=0.38$  (reject)

McNemar's (binomial) test: only consider where A and B are different

GT<sub>x</sub>

## Wilcoxon Signed Rank Test for Medians

- Assumption:
  - Distribution is continuous and symmetric
- Question
  - Is the median of the distribution different from  $m$ ?



GT<sub>x</sub>

## Wilcoxon Signed Rank Test

### Is the median different from $m$ ?

Given responses  $y_1, \dots, y_n$

1. Rank  $|y_1 - m|, \dots, |y_n - m|$  from smallest to largest
2.  $W = \sum_{y_i > m} \text{rank}(y_i - m)$  = sum of all ranks where  $y_i > m$
3. p-value test for  $W$

### Do two sets of paired samples have the same median?

Given pairs  $(y_1, z_1), \dots, (y_n, z_n)$  from observations  $y$  and  $z$

- Use  $|y_1 - z_1|, \dots, |y_n - z_n|$  for rank test

### Comparing paired samples

- Numeric data: use Wilcoxon
- Yes/no data: use McNemar

GT<sub>x</sub>

## Mann-Whitney test

Two data sets, but not paired samples

- Given independent observations  $y_1, \dots, y_n$ , and  $z_1, \dots, z_m$ 
  1. Rank all observations together:  $y_1, \dots, y_n, z_1, \dots, z_m$
  2.  $U$  = smaller of two adjusted rank sums:
$$U = \min\{U_y, U_z\}$$
$$U_y = \sum_{i=1}^n \text{rank}(y_i) - \frac{n(n+1)}{2}$$
$$U_z = \sum_{j=1}^m \text{rank}(z_j) - \frac{m(m+1)}{2}$$
  3. Find significance of  $U$  (need software or a table)

GT<sub>x</sub>

## Summary

### Nonparametric tests

- Use even when nothing is known about underlying distribution
- Two data sets
  - McNemar's test (paired yes/no)
  - Wilcoxon signed rank test (paired numeric data)
  - Mann-Whitney (unpaired)
- One data set
  - Wilcoxon signed rank test (compare possible median)
- Other nonparametric tests too



- Because we don't have a good distribution to fit parameters to, a nonparametric test is useful. Many nonparametric tests focus on the median, and they can be used even when we do not know the form of the underlying distribution.}}
- This one is a little tricky! By focusing on the median, nonparametric tests make it less important whether a small data set includes the right distribution and range of results. All a nonparametric test needs is enough data to figure out approximately where the middle value is.}}
- 16.2 Bayesian Modeling

## Bayesian Models

- Conditional probability -- Bayes' rule or Bayes' theorem
  - $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$
- Example: Medical test for a disease
  - True positives: 98%
  - False positives: 8%
  - 1% of population really has disease
    - 8.9% of people test positive
  - If someone tests positive, what is the probability they have the disease?

GTx



## Bayesian Models

- $A$ : has the disease
- $B$ : tested positive
- $P(A|B) = \frac{P(B|A)P(A)}{P(B)} = \frac{98\% \times 1\%}{8.9\%} = 11\%$
- After testing positive
  - a person has only an **11%** chance of having the disease

Why?

So many more people don't have the disease...

...so there are many more false positives than true positives.

GT<sub>x</sub>

- Bayesian Models
  - $P(B|A)$  = probability of testing positive given  $A = 98\%$
  - $P(A)$  = probability of having the disease =  $1\%$
  - $P(B)$  = probability of testing positive =  $8.9\%$
- Empirical Bayes Modeling

## Summary

### Bayesian approach

- Even a single observation
  - Combined with broader set of observations
- } Make a deduction or prediction
- Bayesian models work especially in the absence of lots of data



- Expert opinion can be used to define the initial distribution of  $P(A)$ , and observed data about  $B$  can be used with Bayes' theorem to obtain a revised opinion  $P(A|B)$ .
- The initial distribution assumed for  $P(A)$  is called the 'prior distribution' and the revised distribution  $P(A|B)$  is called the 'posterior distribution'
- 16.3 Communities in Graphs



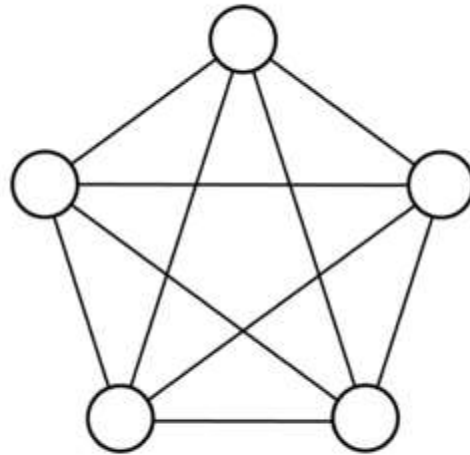
# Communities

## Community

- a set of circles that's highly connected within itself

## Graph

- Circles = nodes/vertices
- Lines = arcs/edges
- **Clique** = a set of nodes that all have edges between each other



Louvain algorithm: decomposing a graph into communities

GT<sub>x</sub>

## Louvain Algorithm

Maximize the modularity of a graph

- $a_{ij}$ : weight on the arc between nodes  $i$  and  $j$
- $w_i$ : total weight of arcs connected to  $i$
- $W$ : total weight of all the arcs
- Modularity =  $\frac{1}{2W} \sum_{i,j \text{ in same community}} (a_{ij} - \frac{w_i w_j}{2W})$

GT<sub>x</sub>

# Louvain Algorithm

## Step 0

Each node is its own community

## Step 1

Repeat...

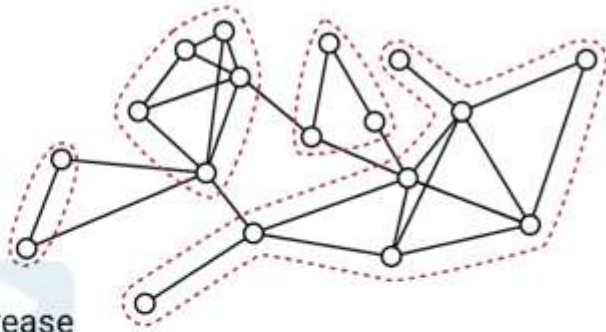
Make biggest modularity increase  
by moving a node from its  
community to an adjacent  
node's community

...until no move increases modularity

## Step 2

Each community is a super-node

Repeat Step 1 using super-nodes



GTx

- Modularity is a measure of how well the graph is separated into communities or modules that are connected a lot internally, but not connected much between each other.
- Louvain is a heuristic (not guaranteed to find the absolute best partition of the graph into communities, it often gives very good solutions though very quickly).
  - Used somewhat often when we want to find communities inside a large network, especially social media networks, and networks of people, computers, etc.
- [Quick jargon break: just like a set of nodes with edges between each pair is called a 'clique', a set of nodes without any edges between them is called an 'independent set'.]
- 16.4 Neural Networks and Deep Learning
  - Basic way to adjust weights within the model is to use gradient descent using the slope of a function

# Neural Networks

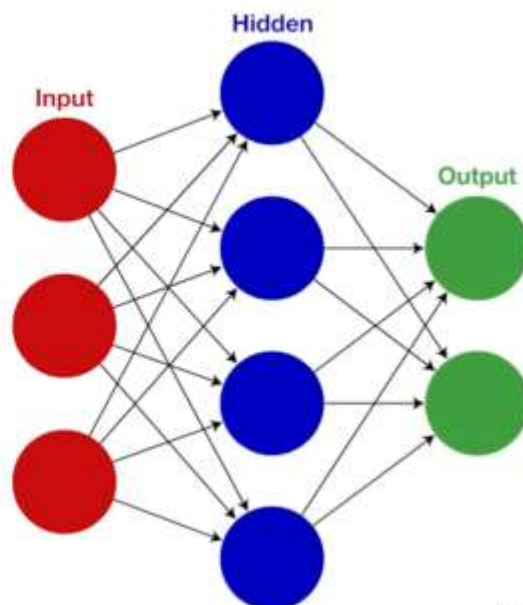
## 3 levels of neurons:

- Input level
- Hidden level
- Output level

## Each neuron:

- Gets inputs from previous layer
- Calculates function of weighted inputs
- Gives its output to next layer

Weights/functions updated based  
on correctness of results



GTx

## Deep Learning

- Similar to neural networks
- Many layers (“deep”)
- Relative success in
  - Natural language processing
  - Speech recognition
  - Image recognition



GTx

- Deep learning is currently one of the best approaches for recognizing images, speech, writing, and language.
- 16.5 Competitive Models

## Us-Against-The-Data

- Descriptive models
  - Get an understanding of reality
- Predictive models
  - Find hidden relationships
  - Predict the future
- Prescriptive models
  - Find the best thing to do
- Assumes the system does not react
- What if the system reacts intelligently?
  - Use analytics to consider all sides of the system



GTx

# Zero Sum and Non-Zero-Sum Games

- **Zero-sum**
  - Whatever one side gets, the other side loses
  - Example: Rock/paper/scissors
    - Overall outcome is 1 win + 1 loss
- **Non-zero-sum games**
  - Total benefit might be higher or lower
  - Example: Economics

GTx

- Competitive decision making (game theoretic model) appropriate
  - A company wants to optimize its production levels, based on production cost, price, and demand. The company already has estimated a function to give predicted selling price and demand as a function of the number of units produced, and the number of units its competitor produces.
    - A company wants to optimize its production levels, based on production cost, price, and demand. The company already has estimated a function to give predicted selling price and demand as a function of the number of units produced, and the number of units its competitor produces.

## Game Theory Example (Shell's point of view)

Cost = \$1.00/gallon

	BP \$2.50	BP \$2.00
Shell \$2.50	Profit (both) = $\frac{d}{2}(\$1.50) = \$0.75d$	Profit (Shell) = \$0
Shell \$2.00	Profit (Shell) = $d(\$1.00) = \$1.00d$	Profit (Shell) = $\frac{d}{2}(\$1.00) = \$0.50d$

Lower price is better

Lower price is better

Because of symmetry, lower price is better for BP too.

GTx



## Same Example with Smaller Profit Margin

Cost = \$1.75/gallon

	BP \$2.50	BP \$2.00	
Shell \$2.50	Profit (both) = $\frac{d}{2}(\$0.75)$ = $\$0.375d$	Profit (Shell) = \$0	Higher price is better
Shell \$2.00	Profit (Shell) = $d(\$0.25)$ = $\$0.25d$	Profit (Shell) = $\frac{d}{2}(\$0.25)$ = $\$0.125d$	

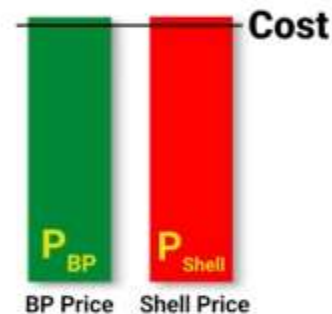
Higher price is better

Now, both are better off charging higher price.

GT<sub>x</sub>

## Extension of Example

- Choose any price they want
- BP price =  $p_{BP}$ 
  - Shell: If  $p_{Shell} > p_{BP}$ , then profit = \$0 (sell nothing)
  - If  $p_{Shell} = p_{BP}$ , then profit =  $\frac{d}{2}(p_{Shell} - \text{cost})$
  - If  $p_{Shell} < p_{BP}$ , then profit =  $d(p_{Shell} - \text{cost})$
  - So, Shell might price slightly lower price than BP
    - Then, BP might price slightly lower than Shell
      - Then Shell might price slightly lower than BP
        - Etc.
- Both keep lowering prices until price is about equal to the cost



GT<sub>x</sub>