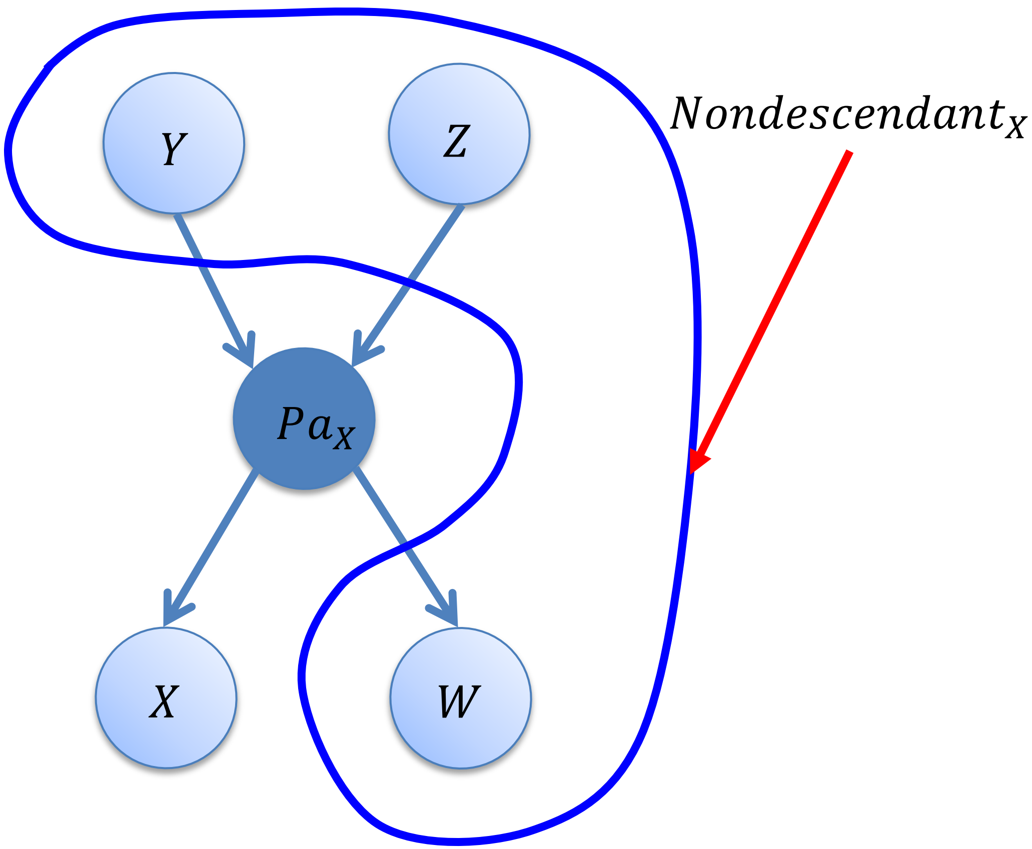
Review

Le Song

Machine Learning
CSE/ISYE 6740, Fall 2019

Factorization in directed GM

- Local Markov Assumption
  - $X \perp Nondescendant_X \mid Pa_X$



$$P(X, Y, Z, W, Pa_X) =$$
$$P(Y)$$
$$P(Z)$$
$$P(Pa_X \mid Y, Z)$$
$$P(X \mid Pa_X)$$
$$P(W \mid Pa_X)$$

In general:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid Pa_{X_i})$$

  - $X \perp Y \mid Pa_X, X \perp Z \mid Pa_X, X \perp W \mid Pa_X$

Factorization in undirected GM

- Given an undirected graph G over variables $\mathcal{X} = \{X_1, \dots, X_n\}$

- A distribution P factorizes over G if there exist
  - subset of variables $D_1 \subseteq \mathcal{X}$, …, $D_m \subseteq \mathcal{X}$ (D_i are maximal cliques in G)
  - non-negative potentials (factors/functions) $\Psi_1(D_1)$,…, $\Psi_m(D_m)$
  - such that

$$P(X_1, X_2, \dots, X_n) = \frac{1}{Z} \prod_{i=1}^{m} \Psi_i(D_i)$$

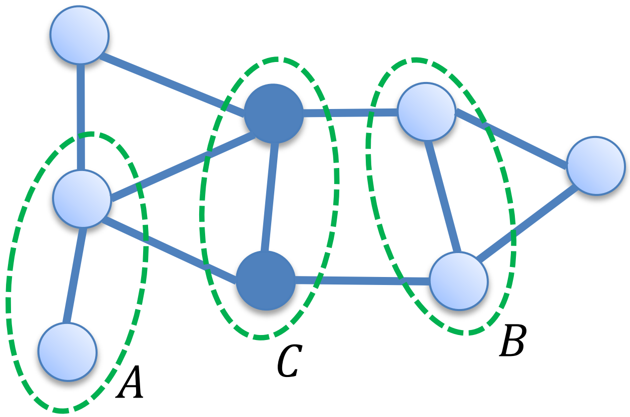
where

$$Z = \sum_{x_1, x_2, \dots, x_n} \prod_{i=1}^{m} \Psi_i(D_i) = \sum_{\boldsymbol{X}} \prod_{i=1}^{m} \Psi_i(D_i)$$

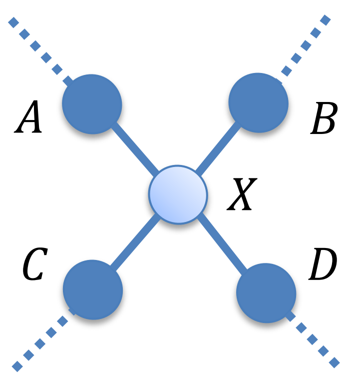
Also know as Gibbs distributions, Markov random Fields, and undirected graphical models

Read conditional independence from UGM

- Global Markov Independence $A \perp B \mid C$
  - Independence based on separation

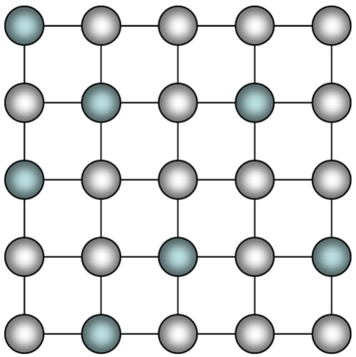


- Local Markov Independence $X \perp TheRest \mid ABCD$
  - ABCD Markov blanket



Pairwise Markov Networks

- All factors over single variables or pairs of variables
  - Node potentials $\Psi_i(X_i) > 0$
  - Edge potentials $\Psi_{ij}(X_i, X_j) > 0$



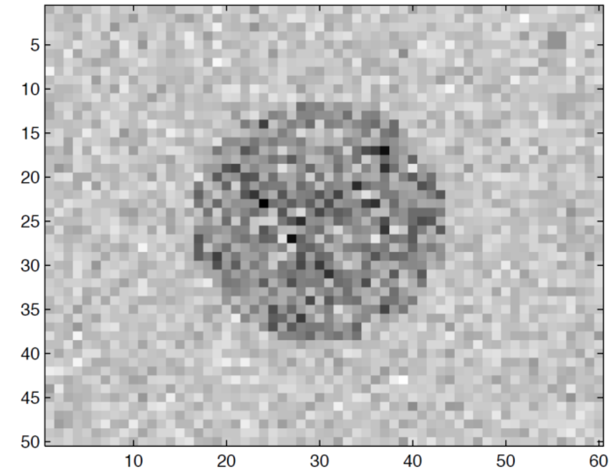
- Factorization
  - $P(X) = \frac{1}{Z} \prod_{i \in V} \Psi_i(X_i) \prod_{(i,j) \in E} \Psi_{ij}(X_i, X_j)$

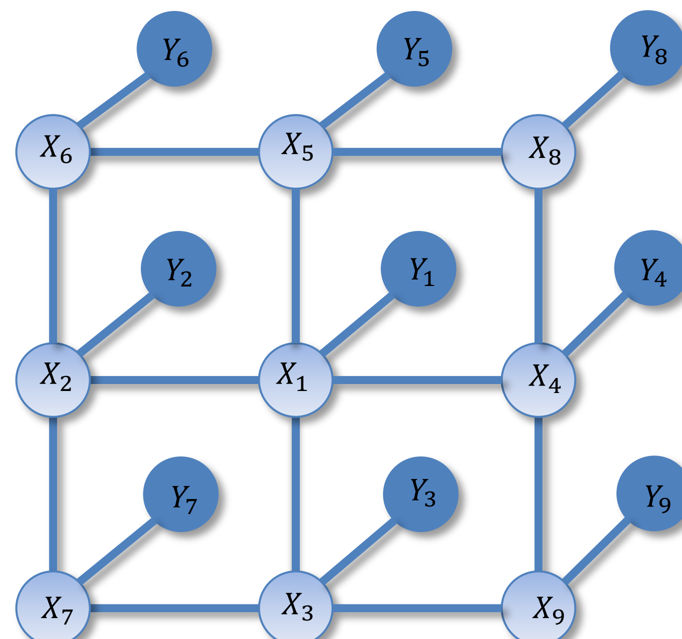
- Eg. Exponential form

$$P(X_1, \ldots, X_n) = \frac{1}{Z} \exp\left(\sum_{(i,j) \in E} \theta_{ij} X_i X_j + \sum_{i \in V} \theta_i X_i + \sum_{i \in V} \alpha_i X_i^2 \right)$$

Image Segmentation

- Noisy grayscale image
- Foreground vs. background pixels
- Model using a pairwise MRF
  - $P(X) = \frac{1}{Z} \prod_i \Psi(X_i) \prod_{ij} \Psi(X_i, X_j)$
  - $\Psi(x_i) = \exp\left(-\frac{\left(y_i - \mu_{x_i}\right)^2}{2\sigma_{x_i}^2}\right)$
  - $\Psi(x_i, x_j) = \exp\left(-\beta\left(x_i - x_j\right)^2\right)$



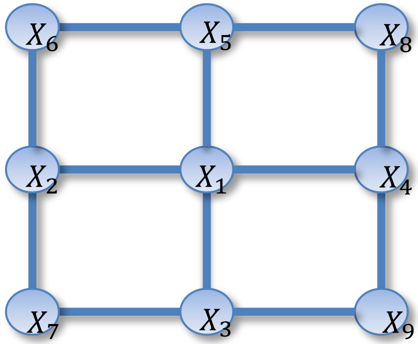


Learning Markov random fields

$$P(X_1, \ldots, X_k|\theta) = \frac{1}{Z(\theta)} \exp\left(\sum_{ij} \theta_{ij} X_i X_j + \sum_i \theta_i X_i\right)$$

$$= \frac{1}{Z(\theta)} \prod_{ij} \exp(\theta_{ij} X_i X_j) \prod_i \exp(\theta_i X_i)$$

$$where \;\; Z(\theta) = \sum_{\boldsymbol{X}} \prod_{ij} \exp(\theta_{ij} X_i X_j) \; \prod_i \exp(\theta_i X_i)$$



Log likelihood

$$l(\theta, D) = \log\left(\prod_{l=1}^{N} \frac{1}{Z(\theta)} \prod_{ij} \exp(\theta_{ij} x_i^l x_j^l) \prod_i \exp(\theta_i x_i^l)\right)$$

$$= \sum_{l}^{N} \left(\sum_{ij} \log(\exp(\theta_{ij} x_i^l x_j^l)) + \sum_i \log(\exp(\theta_i x_i^l)) - \log Z(\theta) \right) = \sum_{l}^{N} \left(\sum_{ij} \theta_{ij} x_i^l x_j^l + \sum_i \theta_i x_i^l - \log Z(\theta) \right)$$

can be other feature function $f(x_i)$

Term $\log Z(\theta)$ *does not decompose!*

Derivatives of log likelihood

$$l(\theta, D) = \frac{1}{N}\sum_{l}^{N} \left(\sum_{ij} \theta_{ij} x_i^l x_j^l + \sum_i \theta_i x_i^l - \log Z(\theta) \right)$$

A convex problem

Can find global optimum

- $\frac{\partial l(\theta,D)}{\partial \theta_{ij}} = \frac{1}{N} \sum_l^N \sum_{ij} x_i^l x_j^l - \frac{\partial \log Z(\theta)}{\partial \theta_{ij}}$

- $= \frac{1}{N} \sum_l^N \sum_{ij} x_i^l x_j^l - \frac{1}{Z(\theta)} \frac{\partial\, Z(\theta)}{\partial \theta_{ij}}$

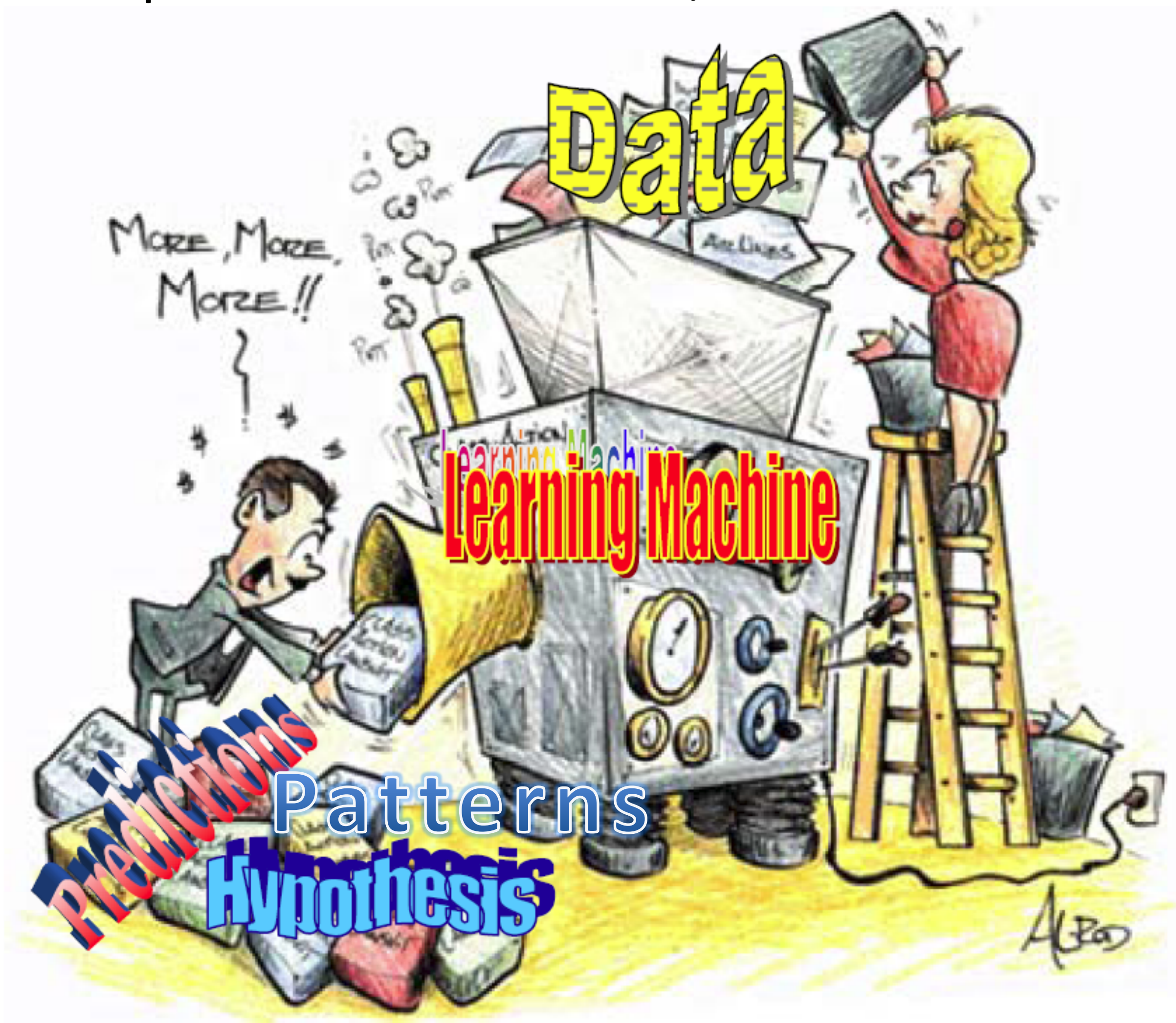
- $= \frac{1}{N} \sum_l^N x_i^l x_j^l - \underbrace{\frac{1}{Z(\theta)} \sum_{\boldsymbol{x}} X_i X_j \prod_{i'j'} \exp(\theta_{i'j'} X_{i'} X_{j'}) \prod_{i'} \exp(\theta_{i'} X_{i'})}_{}$

need to do inference: (loopy) belief propagation

Summary

What is Machine Learning (ML)

- Study of algorithms that can discover patterns from uncertain data, make prediction into future, and react to the environment.



Keys topics

- Unsupervised learning techniques
  - Dimensionality reduction
    - PCA
    - Graph based methods
  - Clustering
    - Kmeans
    - Graph based methods (spectral algorithms)
  - Density estimation
    - Parametric models
    - Histogram
    - Kernel density estimator
    - Mixture of Gaussian

Keys topics

- Supervised learning techniques
  - Feature selection
    - Mutual information
  - Bayes decision rule
    - Naïve Bayes
  - Linear classifier
    - Logistic regression
    - Support vector machine
  - Nonlinear classifier
    - K-nearest neighbors

Keys topics AFTER midterm

- Supervised learning techniques
  - Neural networks
    - Single neuron ≈ logistic regression
    - Deep neural networks
  - Regression
    - Linear regression
    - Polynomial regression
    - Ridge regression

Keys topics

- Advanced topics
  - Generalization ability
    - Overfitting
    - Bias-variance trade-off
    - Cross-validation
  - Kernel methods
    - Kernel functions
    - Feature spaces
    - Kernel tricks
  - Graphical models
    - Directed graphical models (HMM)
    - Undirected applications (MRF)
- Applications
  - Computational Biology, ML system, NLP

The process of designing ML algorithms

- What is the objective?
  - Extract group? Visualization? Reduce computation/memory? Compress data? Find useful features? Classification?
- Formulate the objective
  - Understand your data, and make assumptions: Independent? variance enough? Linear? Gaussian? Euclidean distance?
  - Parameterization: parametric? Nonparametric? Prior? Constraint?
- Looking for algorithms
  - Convex? Nonconvex? Computational and memory complexity? Iterative or one-shot? Global best? Guarantee to improve or stop?
- Interpretation:
  - Results make sense? What groups? What principal component? Selected feature meaningful? What errors made by classifier? Improvements?
- Think deeper
  - Would the learned model perform well in future?
  - Nonlinear models? (Neural networks, Kernel methods) Many variables? (graphical models)

Key mathematical tools

- Linear algebra, vector spaces and functional analysis
  - Vector, projection, linear combination
  - inner product, distance
  - Eigen-decomposition: $A = U\Sigma U^\top, or\ Av = \lambda v$
  - Singular value decomposition: $A = USV^\top, or\ Av = \sigma u$
  - Kernel functions, matrices, Hilbert spaces

- Probability and Statistics
  - Mean, variance, density, distribution, parametric models
  - Sum rule, product rule, Bayes rule, conditional independence
  - Maximum likelihood estimation
    - Fully observed case (often convex)
    - With hidden variables (expectation-maximization algorithm)

Key mathematical tools (cont.)

- Convex Optimization
  - Convex set, convex function
  - Derivative of function (and with respect to vectors, matrices)
  - Lagrangian function, dual problems
  - Optimality conditions
- Computer Science
  - Complexity: computation and memory, trade-off
  - Data structures: image and graph representation, hashing
  - Local search heuristic (greedy algorithms)
  - Sophisticated algorithm: shortest path, nearest neighbor search
  - Programming: loop vs. vectorized, underflow

Example I

We learned about bias-variance decomposition and model selection in the class. It basically deals with the problem of choosing a model complexity. In each sub-questions, we show a pair of two candidate models for various machine learning problems. Mark B on the one with less bias, and mark V on the other. You are not required to explain why. For example,

- Model 1: A model with less bias – (B)
- Model 2: A model with less variance – (V)

(a) [3 pts]

- Model 1: A flexible model with many parameters – (B)
- Model 2: A rigid model with a few parameters – (V)

(b) [3 pts]

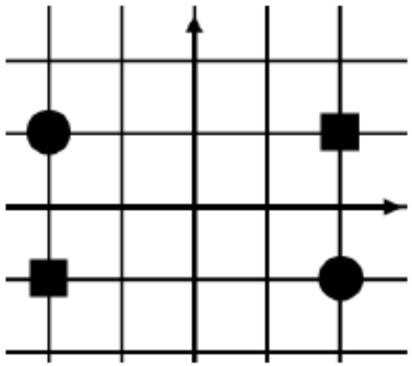
- Model 1: Ridge regression with large regularization coefficient λ – (V)
- Model 2: Unregularized linear regression – (B)

(c) [3 pts]

- Model 1: Regression with higher degree – (B)
- Model 2: Regression with lower degree – (V)

Example II

Consider a supervised learning problem in which the training examples are points in 2-dimensional space. The coordinates (x_1, x_2) and corresponding label Y are shown below



x_1	x_2	Y
2	-1	+1
-2	1	+1
2	1	-1
-2	-1	-1

(a) Are the points linearly separable? [2 pts]

(b) Suppose we use degree-2 polynomial kernel $K(u, v) = (u^T v)^2$. After training the kernel SVM in dual form, we get the Lagrangian multipliers α_i in the table below. Please use this to classify the test point $[-0.5, 2]$. Can you get zero training error now? [6 pts]

x_1	x_2	Y	α
2	-1	+1	0
-2	1	+1	0.0625
2	1	-1	0
-2	-1	-1	0.0625

Example III

The Restricted Boltzmann Machine (RBM) is an undirected graphical model over binary vectors. It has "visible" variables v and "hidden" variables h. The jointly distribution is

$$p(v, h) \propto e^{-E(v,h)} \tag{1}$$

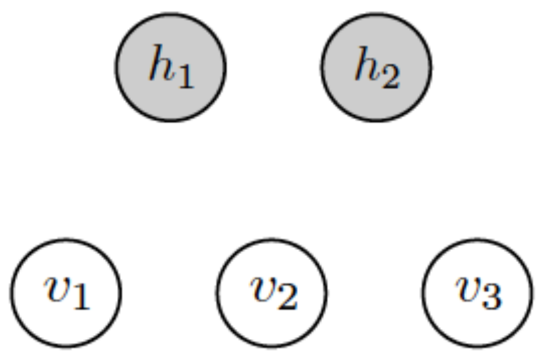
Note that you need to normalize $e^{-E(v,h)}$ to get the probability.

The energy function $E(v, h)$ is defined as

$$E(v, h) = -\sum_i a_i v_i - \sum_j b_j h_j - \sum_{i,j} v_i h_j w_{ij}$$

where v_i and h_j are the binary states of the visible variable i and hidden variable j, respectively. a_i and b_i are their biases, and $w_{i,j}$ is the weight between them.

(a) Consider the RBM with three visible variables and two hidden variables. Complete the graphical model by drawing the edges. [3 pts]



How to do well in final exam?

- The final exam will cover materials relevant to all lectures. Not enough to use just lecture slides
- Some materials will come from textbooks, which you should have read when completing assignments
- The final exam will last 3 hours (2:50—5:40pm). The score will be multiplied by 0.2 and add to overall grade.
- Difficulty similar to midterm II. Lots of bonus points, answer as many questions as you can.

Conclusion

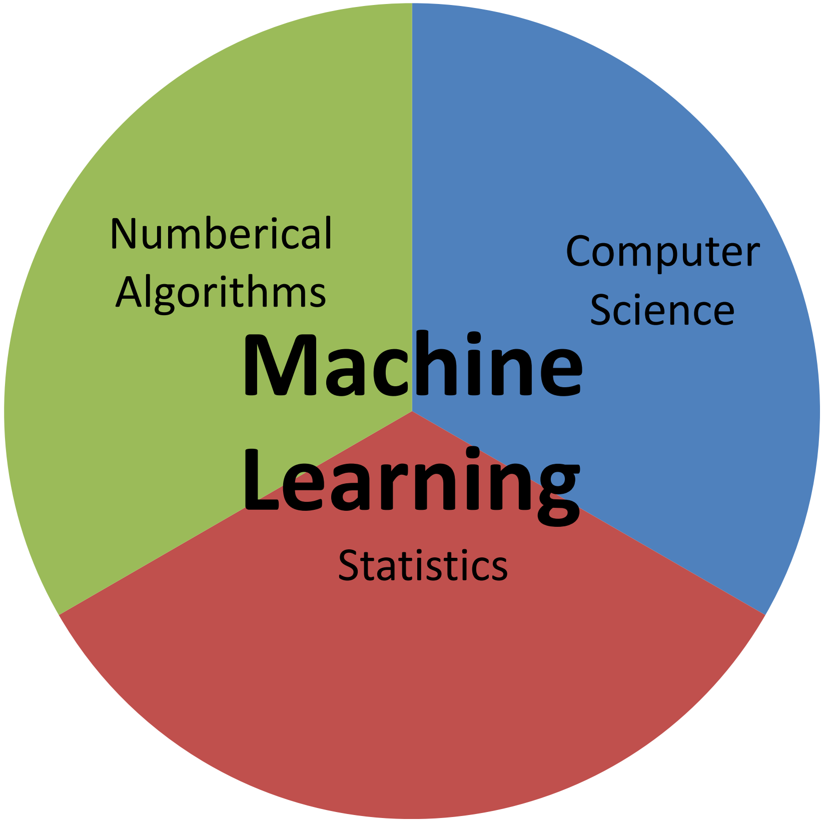
The need for machine learning

Georgia Tech

- Machine Learning is used to facilitate research in many disciplines, such as Computer Vision, Robotics, Planning, Natural Language Processing, HCI, Finance, Business, Computational Biology, Sustainability
- Machine Learning graduates are highly demanded by many high tech companies, such as Google, Microsoft, IBM, Amazon, eBay, Yahoo!, GE, Bloomberg, Walmart, Pandora, …
- Machine Learning has the largest number of Master and PhD applicants in College of Computing
- 300 students across campus want to take advanced machine learning introductory courses (CSE/ISYE 6740/CS7641)

Core machine learning techniques

- Machine Learning is an interdisciplinary field and has strong ties to Computer Science, Statistics and Numerical Algorithms which deliver both methods and theory to the field.



Road to machine learning expert

- Core classes
  - Intermediate Statistics
  - Advanced Introduction to Machine Learning
  - Theoretical Foundation of Machine Learning
  - Probabilistic Graphical Models
  - Convex Optimization
  - Functional analysis

- Others
  - Markov Chain Monte Carlo
  - Reinforcement Learning
  - Numerical Linear Algebra
  - Computer Vision
  - NLP
  - Deep Learning