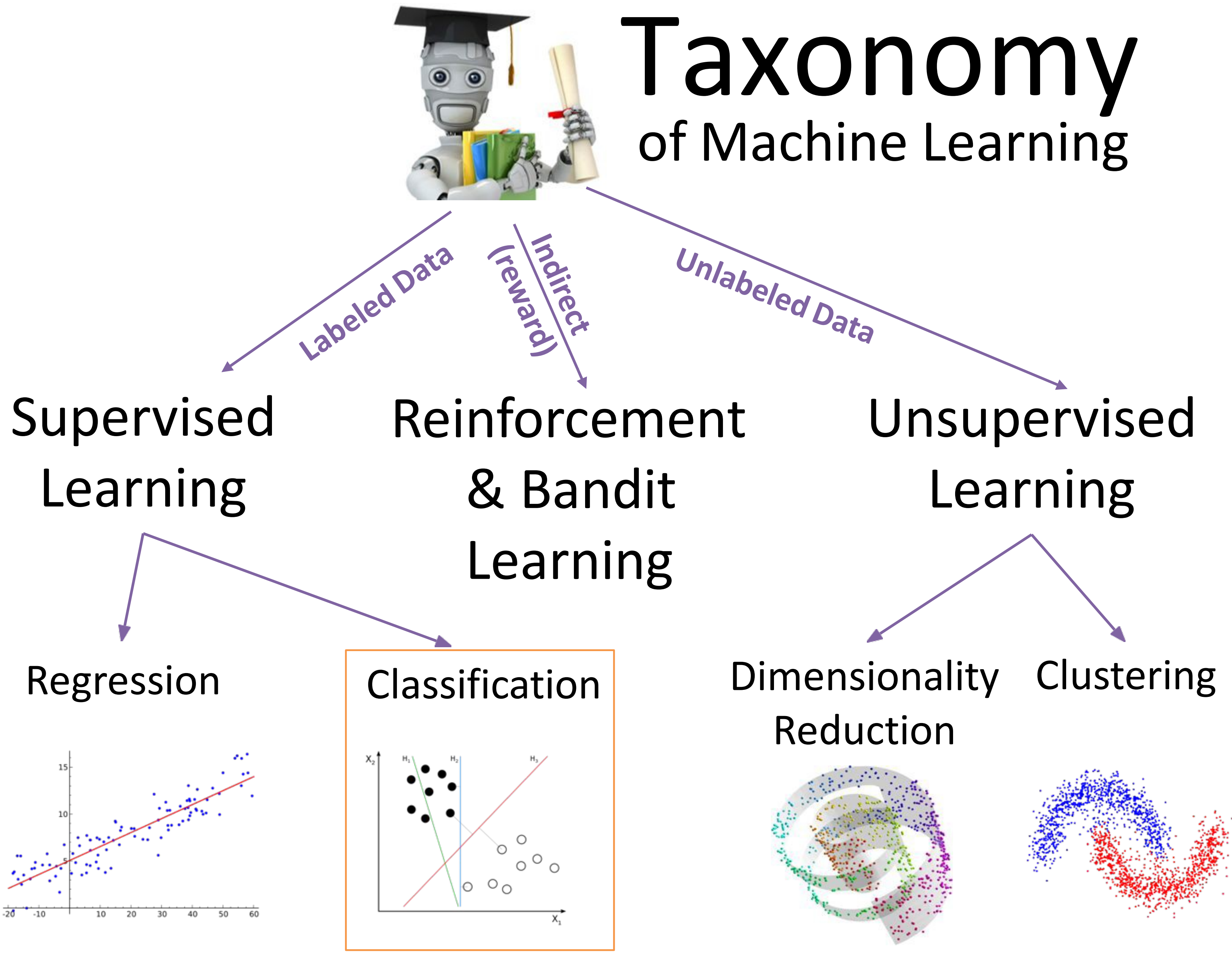
Analytics & Machine Learning in Data Systems (Part 4)

Joseph E. Gonzalez
jegonzal@cs.berkeley.edu

Berkeley cs186


Taxonomy
of Machine Learning
Labeled Data
Indirect
(reward)
Unlabeled Data
Supervised Learning
Reinforcement & Bandit Learning
Unsupervised Learning
Regression
Classification
Dimensionality Reduction
Clustering

Spam **Classification**

- **Goal:** given the **text in an** email **predict** whether it is **spam**
- **Training Data:**

Content	Is Spam
Viagra & Cialas half-off today…	SPAM
Class is Cancelled today	NOT SPAM
Deals on new Autos	SPAM
Receipt from Ritual Coffee …	NOT SPAM

- **First best solution?**
  - What is wrong with this?

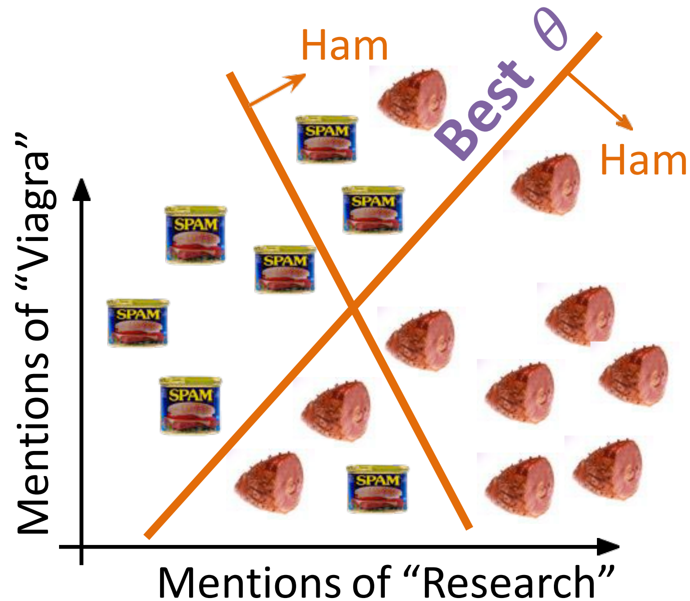
```
def predSpam(doc):
    if “Viagra” in doc:
        return True
    elif “Cialas” in doc:
        return True
    elif “Class” in doc:
        return False
    elif “Deals” in doc:
        return True
    else:
        return False
```

- **Why is Spam Classification Hard?**
  - Easy for humans to **recognize**
  - **Difficult** to formally describe (as an algorithm)
  - **Personal:** different people have different tastes in Spam
  - Good candidate for Machine Learning, the second best solution

Spam **Classification**

- **Goal:** given the **text in an** email **predict** whether it is **spam**
- **Training Data:**

Content	Is Spam
Viagra & Cialas half-off today…	SPAM
Class is Cancelled today	NOT SPAM
Deals on new Autos	SPAM
Receipt from Ritual Coffee …	NOT SPAM



- **Machine Learning:**
  - Learn a function that generalizes the relation:

$$\mathbf{F}(Content; \theta) \rightarrow isSpam$$

  - **F**: is the model type
  - θ: are the model parameters
- Machine learning alg. search for the "best" θ

Basic Classification Models

- Most models predict the probability
  - Why would probability be helpful?
- **Logistic Regression:** *widely used*
  - Similar to least squares regression but for classification
  - **Model form:**

$$\mathbf{P}\left(\text{isSpam} \,|\, \text{Words}\right)$$

- **Naïve Bayes:** *occasionally used*
  - Classic model based on Bayes Rule
  - assumes words are independent given
  - **Model form:**

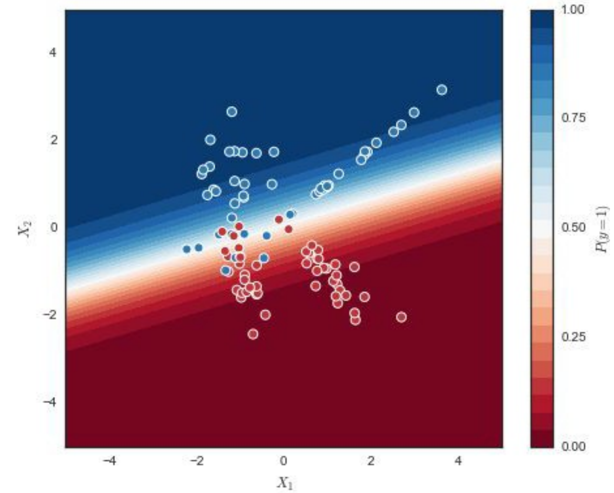
$$\mathbf{P}\left(\text{isSpam} \,|\, \text{Words}\right) \propto \mathbf{P}\left(\text{isSpam}\right) \prod_{\text{Words}} \mathbf{P}\left(\text{Word} \,|\, \text{isSpam}\right)$$

- Which should I use?
  - try both …

Common Classification Models

- □ Most models predict the probability
  - Why would probability be helpful?
- □ **Nearest Neighbor:** *works embarrassingly well*
  - return the label of the nearest training point to the query point
- □ **Logistic Regression:** *widely used and simple*
  - Similar to least squares regression but for classification
- □ **Naïve Bayes:** *occasionally used*
  - Classic model based on Bayes Rule
- □ **Support Vector Machines:** *kernel methods*
  - Capable of automatically growing model size with data
- □ **Deep Learning**: *more on this soon …*

Logistic Regression



X		Y
x_1	x_2	
3.9	3.1	+1
0.2	0.1	-1
4.2	0.9	-1
1.1	2.9	+1

- Basic Model:

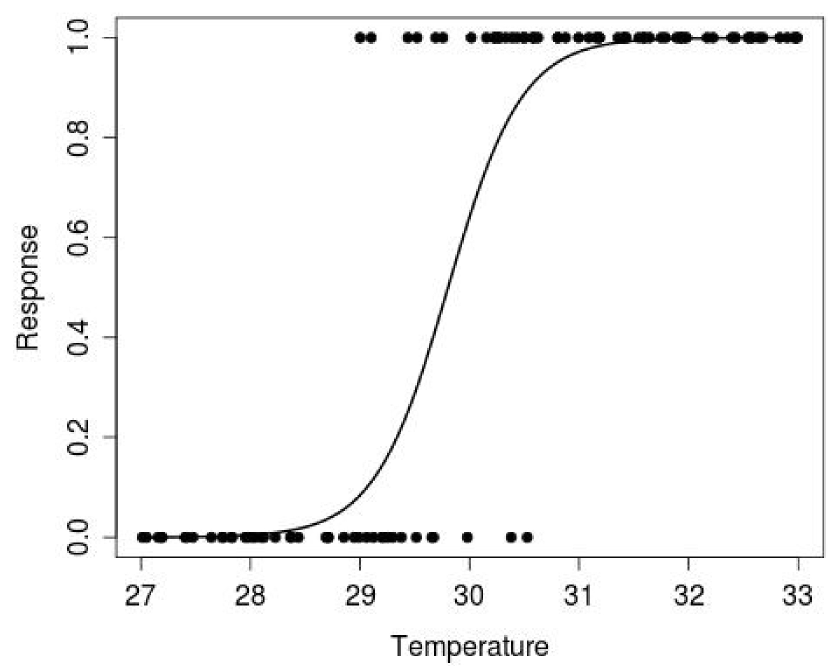
$$\mathbf{P}\left(y|x,\theta\right) = \sigma\left(y(\theta^T x)\right)$$

$$= \frac{1}{1+\exp\left(-y(\theta^T x)\right)}$$

Note that y is either +1 or -1

- Logistic Function:

$$\sigma(z) = \frac{1}{1+\exp(-z)}$$



Learning the Logistic Regression Model

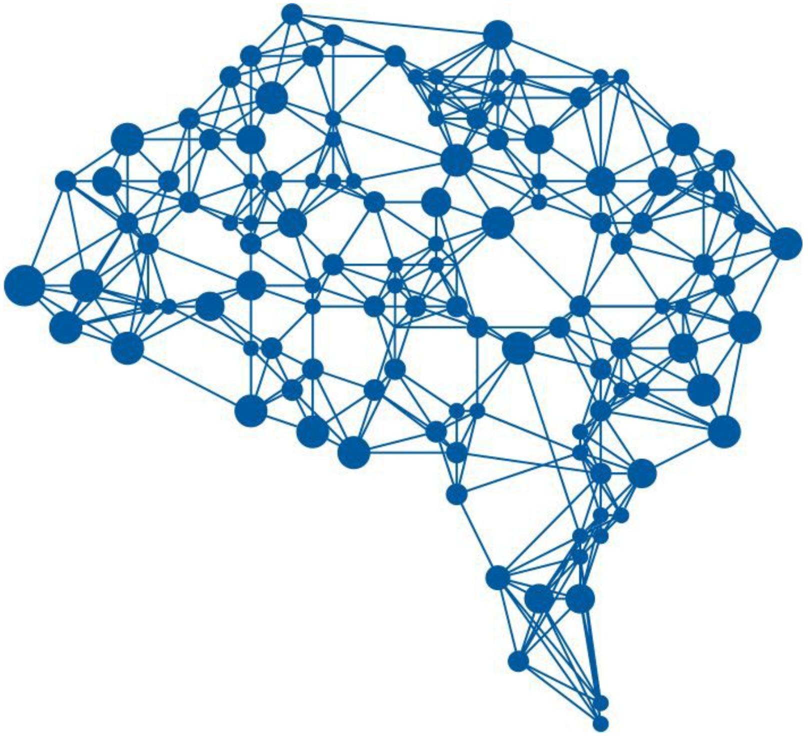
- How do we fit the Logistic Regression model?
  - method of maximum likelihood
- Select the best θ by maximizing prob. of data
  - Solve the following **convex** optimization problem

$$\hat{\theta} = \arg\min_{\theta \in \mathbb{R}^p} \quad \frac{1}{n} \sum_{i=1}^{n} \log\left(1 + \exp\left(-y_i(\theta^T x_i)\right)\right) + \lambda R(\theta)$$

  - Regularized using same techniques as regression

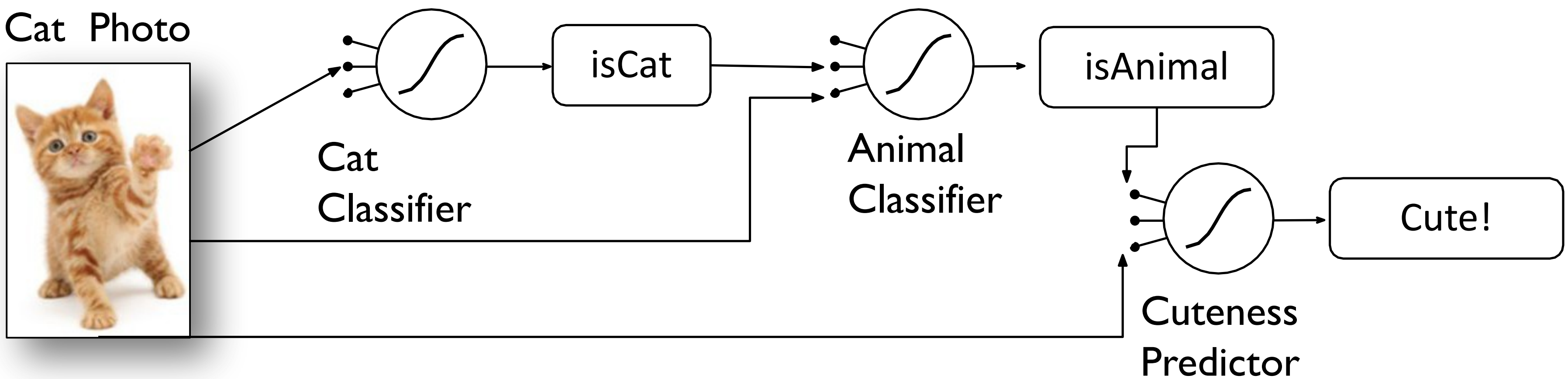
- Optimized using numerical methods
  - **SGD**: Stochastic Gradient Descent

Deep Learning



Intuition

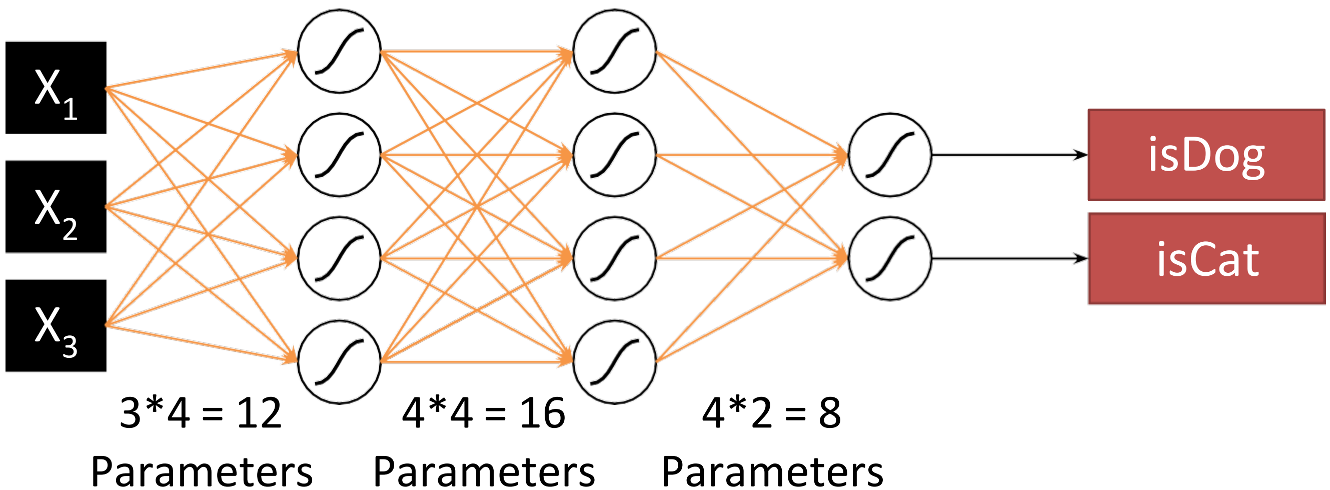
- Model Composition



- Predictions from one model→features for another
- Why not train the entire pipeline of models?

Going a Little Deeper

- **Basic idea**: stacking **logistic regression** models



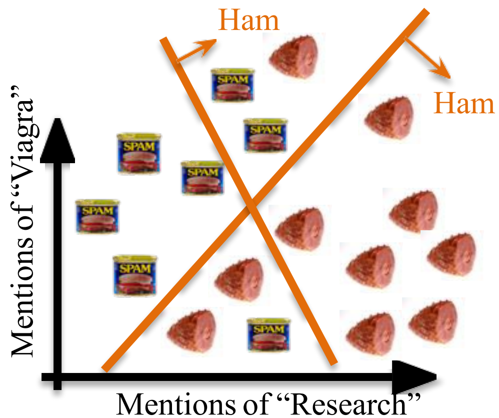
- Many parameters θ (36 in the above model)
  - millions of parameters fits complex functions
  - Requires substantial training data to prevent over-fitting
- Tricky & slow to train
  - Specialized training algorithms and GPU acceleration

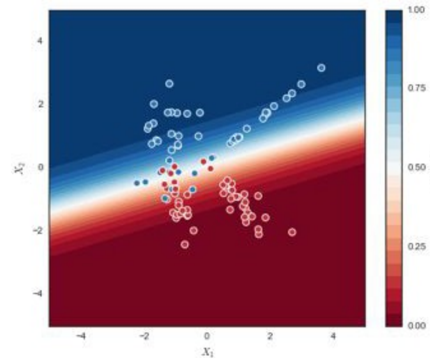
http://playground.tensorflow.org/

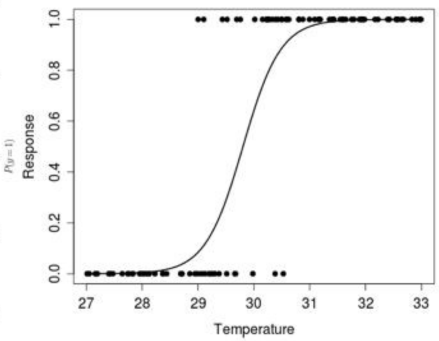
Deep Learning the Big Shift in ML

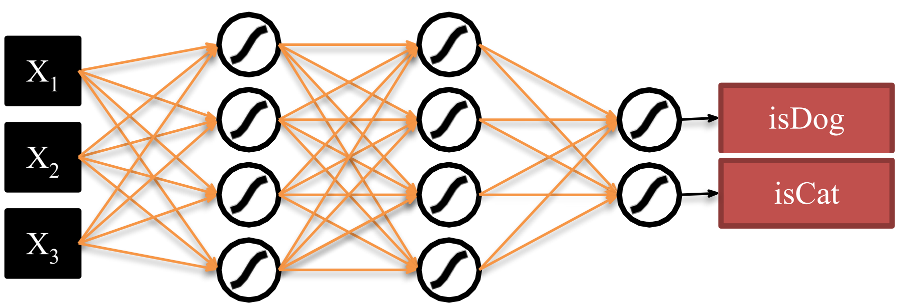
- Recent **big** trend in machine learning
- State of the art results in
  - **Computer Vision**: exceeding human abilities
  - **Speech Recognition**: at the core of all commercial speech recognition systems
  - **AI + Search**: Google's AlphaGo
- Companies investing heavily in Deep Learning:
  - Facebook, Google, Baidu, Nvidia, & Intel have very large Deep Learning groups
  - New software and hardware
- **Hype:** Still requires substantial amounts of data and expertise to train and deploy …
  - Many applications still use other techniques
  - AI winter is coming ...?

Summary of Classification:









Big

Data

Taxonomy
of Machine Learning

Labeled Data

Indirect (reward)

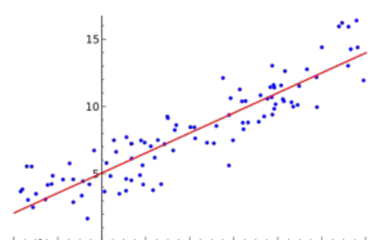
Unlabeled Data

Supervised Learning

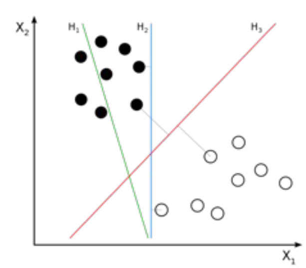
Reinforcement & Bandit Learning

Unsupervised Learning

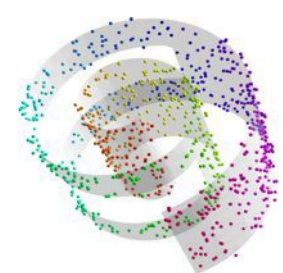
Regression



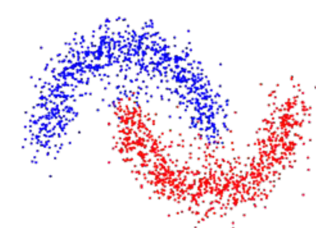
Classification



Dimensionality Reduction



Clustering



Taxonomy
of Machine Learning

Labeled Data

Indirect (reward)

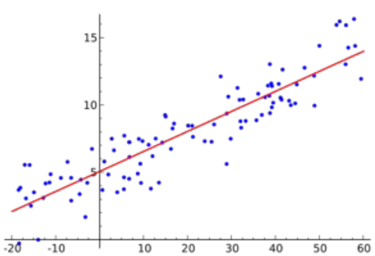
Unlabeled Data

Supervised Learning

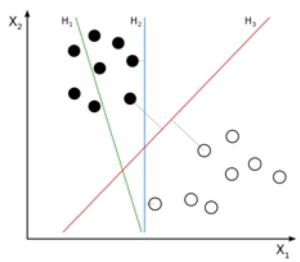
Reinforcement & Bandit Learning

Unsupervised Learning

Regression

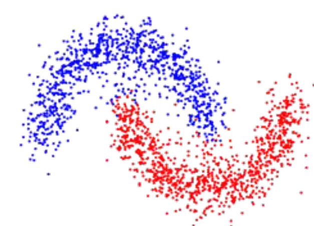


Classification



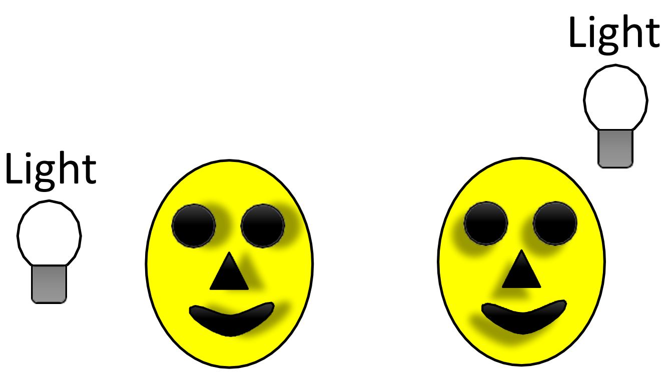
Dimensionality Reduction

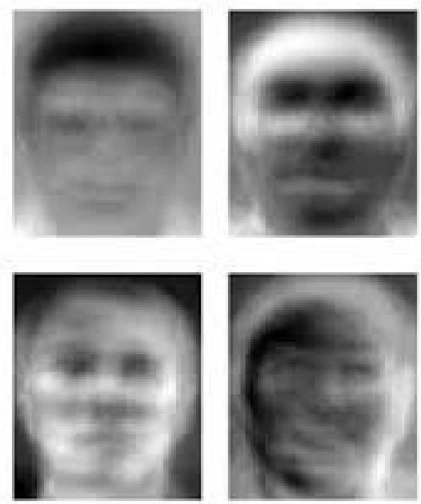
Clustering

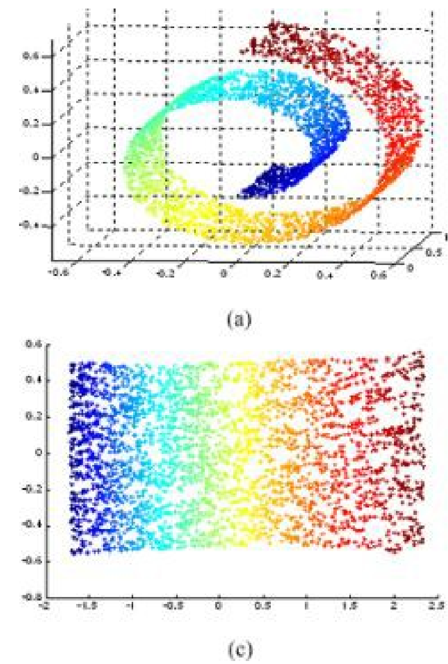


Dimensionality Reduction: Eigen Faces

Given images under different lighting construct images under any light lighting







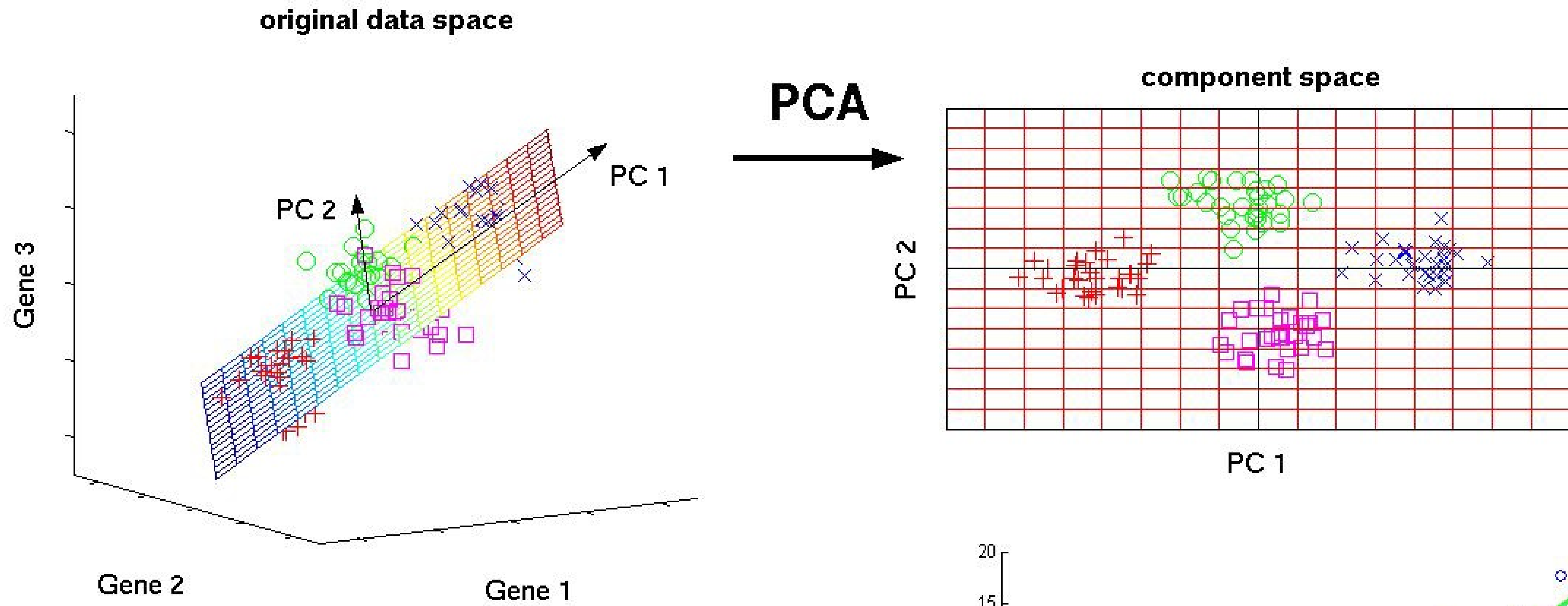
- □ **Machine Learning Approach:**

$$\textbf{Embedding}(\mathit{Image};\ \theta) \mapsto \{x_1, x_2, x_3, x_4\}$$

$$\textbf{Recovery}(\{x_1, x_2, x_3, x_4\};\ \theta) \mapsto \mathit{Reconstructed\ Image}$$

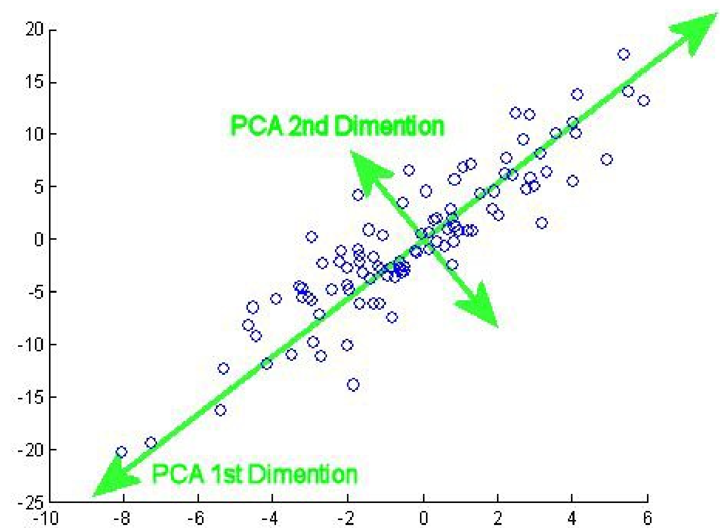
- □ Use common structure in data to identify embedding

Principal Component Analysis



Big Ideas

- □ Identify dimensions of maximum variance
- □ Project data onto those dimensions



Scaling Principal Component Analysis

□ PCA Algorithm

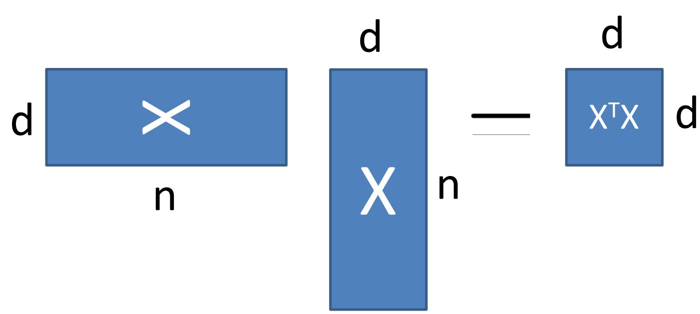
- Computes eigenvectors of of covariance matrix

$$\mathbf{Cov}(X) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})^T = \frac{1}{n}X^T X - \bar{x}\bar{x}^T$$

- The covariance matrix *d×d* is generally smaller than X (*n×d)*
  - For high dimensional data consider dist. Lacnzos …

□ We therefore only need to compute:

$X^T X =$



$$= \sum_{i=1}^{n} x_i x_i^T$$

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i =$$

- In summation form
- Only one pass required!

PCA for Anomaly Detection

- Run PCA and get top k eigenvectors: $V_{(k)}$

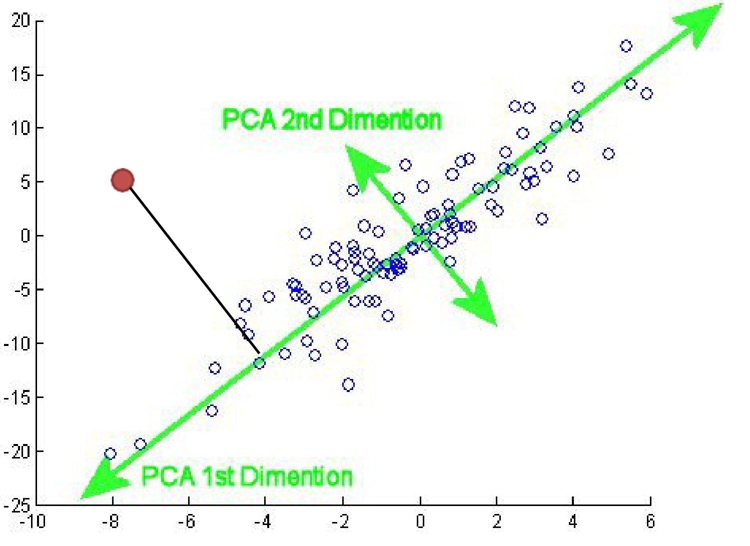
$$\mathbf{Proj}(x) = V_{(k)}^T (x - \bar{x})$$

$$\mathbf{Recv}(q) = V_{(k)} q + \bar{x}$$

- Compute the error in approximate recovery:

$$\mathbf{Error}(x) = \| x - \mathbf{Recv}\left(\mathbf{Proj}\left(x\right)\right) \|_2^2$$

- Outliers are those points far from their embedding



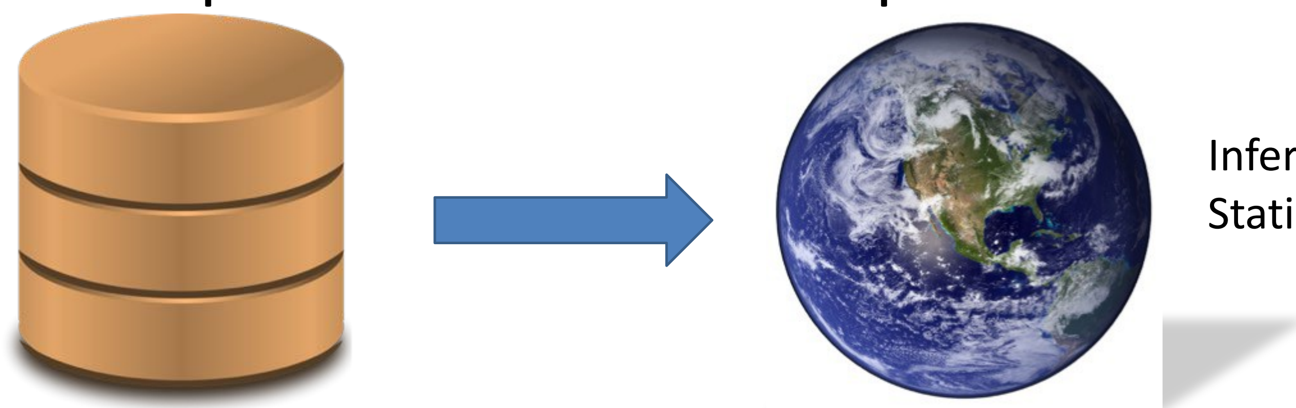
Knowledge Discovery in Databases (KDD)

- Process of extracting ***knowledge*** from a ***data***

Sample

Population

Descriptive Statistic



Inferential Statistics

- **Descriptive Statistics:** *describe* the sample data
    - Can be **measured directly** from the database
- **Inferential Statistics:** *estimate* the population
    - May be **estimated** using descriptive statistics

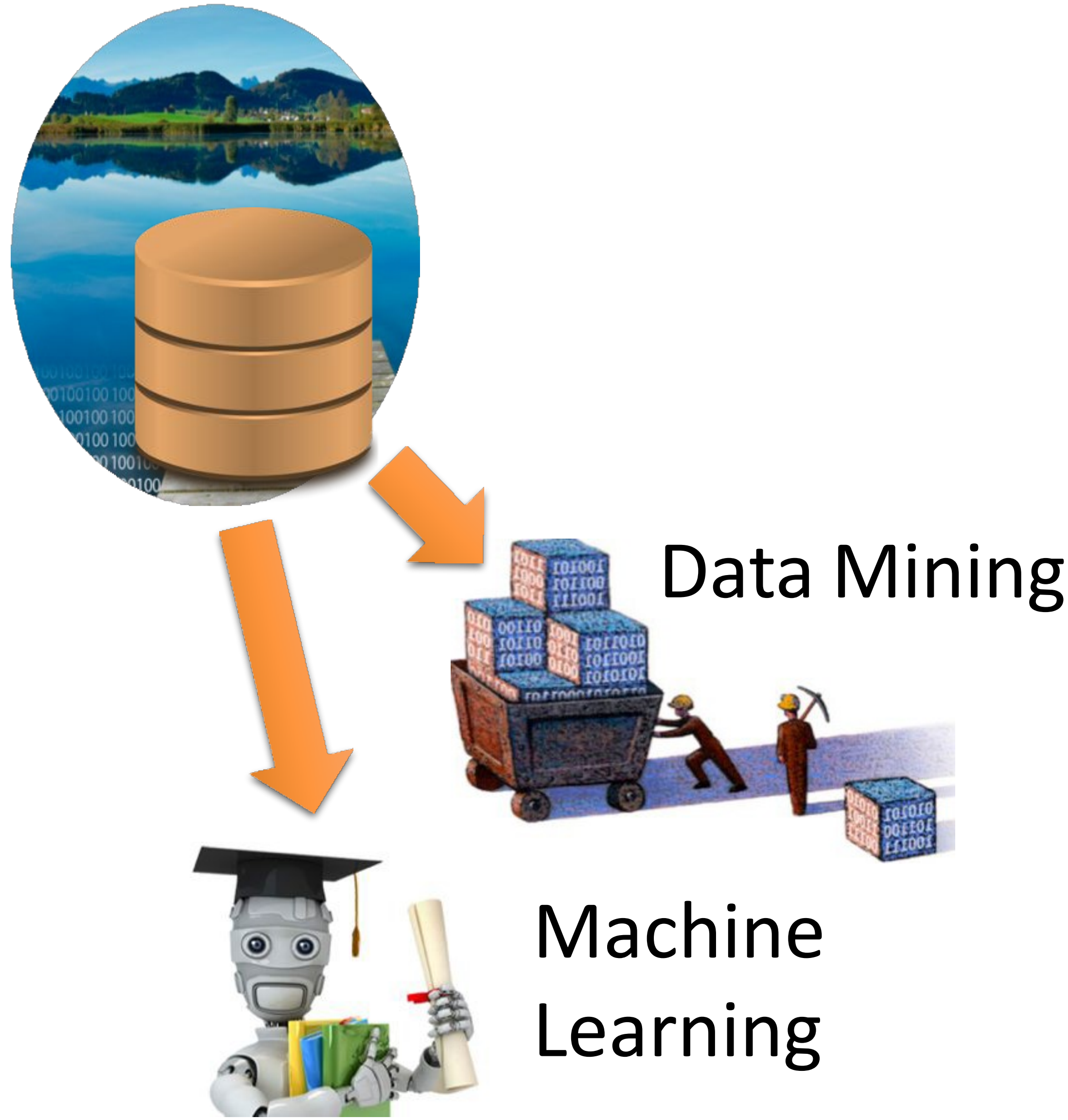
The Knowledge Discovery Process

- □ **Data Selection:** *What data do I need for a given task?*
  - If data was already collected, how was the data collected?

- □ **Data Cleaning:** *Preparing the data for a given task*
  - Typically most challenging (time consuming) part.
  - Why might ETL not be enough?

- □ **Data Mining & ML:** *Running algorithms to infer patterns*
  - The fun part! Many tools, many options, complex tradeoffs.

- □ **Evaluation:** *Verifying that patterns are significant*
  - Algorithms will typically find patterns especially when none exist.


Data Mining
Machine Learning