Machine Learning: Just a Bit More Review

Alexander G. Ororbia II

Introduction to Machine Learning

CSCI-736

1/21/2025

Better Form Your Teams!

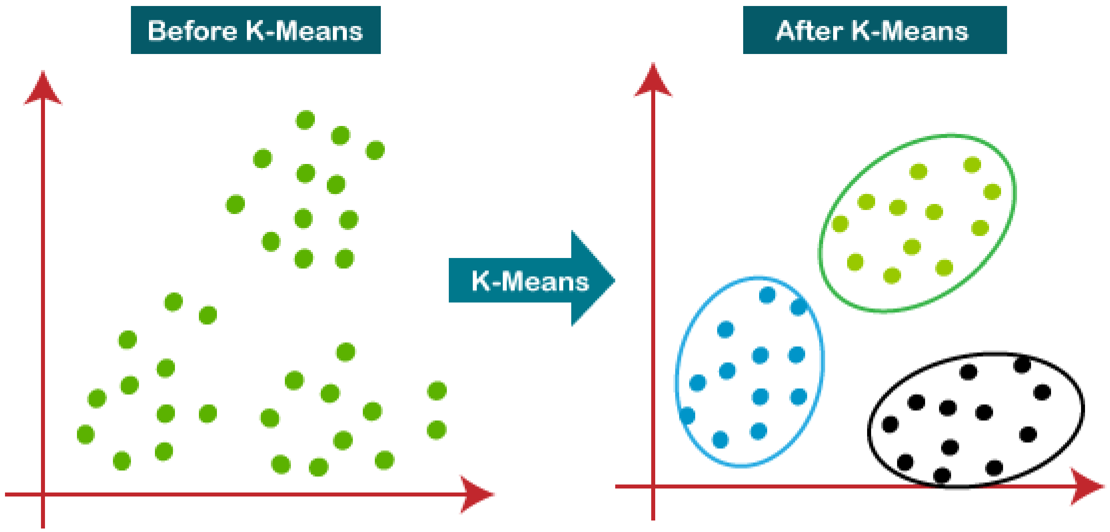
- If you have not already, your teams need to be formed by this Friday, January 24 **by 11:59am (noon!)**
  - Otherwise, you shall be randomly assigned team members/to a team
  - Start figuring out your projects – proposal will be due and you will need to present what your team is doing!
- Starting next week, teams will be presenting on our weekly topic (Thursday)
  - Team will be assigned this Friday/Saturday

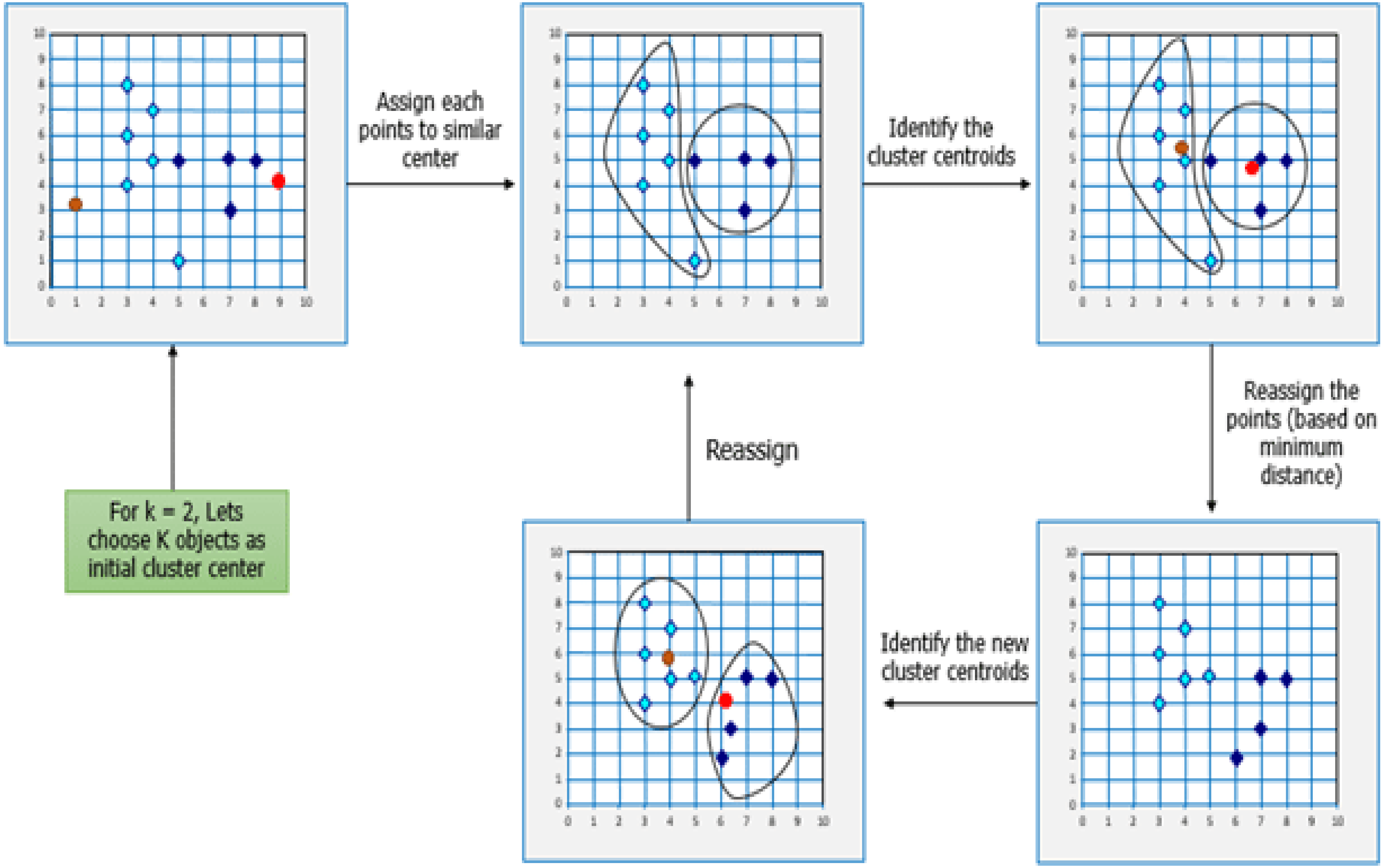
Machine Learning Problems

	Supervised Learning	***Unsupervised Learning***
Discrete	classification or categorization	$\boxed{\text{clustering}}$
Continuous	regression	dimensionality reduction

Clustering Strategies

- K-means
  - Iteratively re-assign points to the nearest cluster center
- Agglomerative clustering
  - Start with each point as its own cluster and iteratively merge the closest clusters
- Many kinds of clustering: mean-shift clustering, spectral clustering, etc.



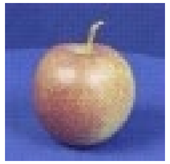

Assign each points to similar center
Identify the cluster centroids
Reassign the points (based on minimum distance)
Identify the new cluster centroids
Reassign
For k = 2, Lets choose K objects as initial cluster center

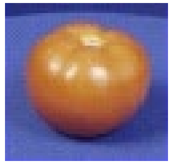
Machine Learning Problems

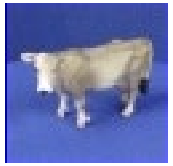
	Supervised Learning	***Unsupervised Learning***
Discrete	$\boxed{\text{classification or categorization}}$	clustering
Continuous	regression	dimensionality reduction

The Machine Learning (ML) Framework

- Apply a prediction function to a feature representation of the image to get the desired output:

f() = “apple”

f() = “tomato”

f() = “cow”

Slide credit: L. Lazebnik

The Machine Learning (ML) Framework

$$y = f(\mathbf{x})$$

output

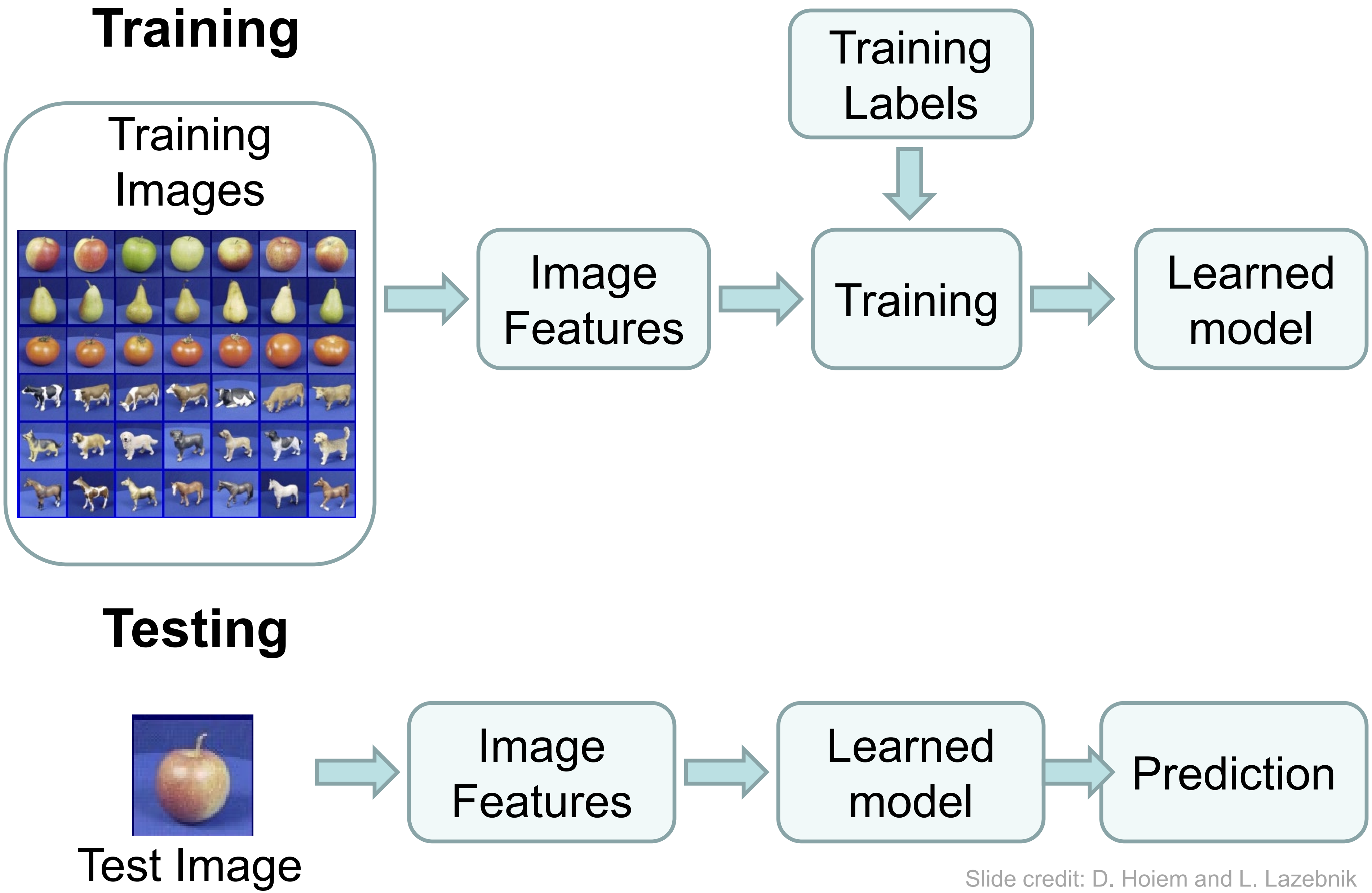
prediction function

Input features

- **Training:** given a *training set* of labeled examples $\{(\mathbf{x}_1,y_1), \ldots, (\mathbf{x}_N,y_N)\}$, estimate the prediction function f by minimizing the prediction error on the training set
- **Testing:** apply f to a never-before-seen *test example* $\mathbf{x}$ and output the predicted value $y = f(\mathbf{x})$

Slide credit: L. Lazebnik

ML Steps



Slide credit: D. Hoiem and L. Lazebnik

Many classifiers to choose from

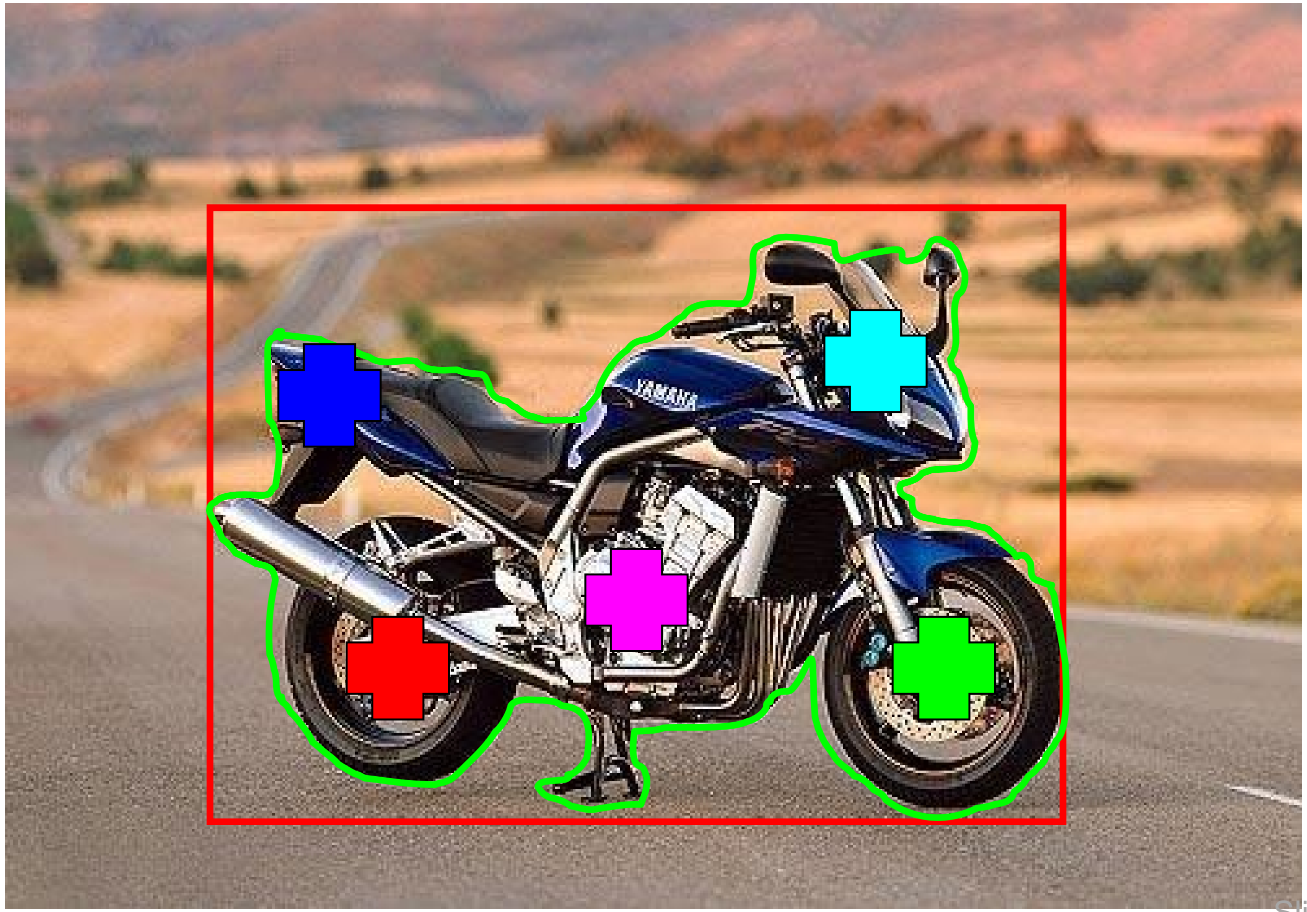
- SVM
- Artificial neural networks
- Naïve Bayes
- Bayesian network
- Logistic regression
- Randomized Forests
- Boosted Decision Trees
- K-nearest neighbor
- RBMs/Harmoniums
- Etc.

Which is the best one?

Recognition Task and Supervision

- Images in the training set must be annotated with the "correct answer" that the model is expected to produce

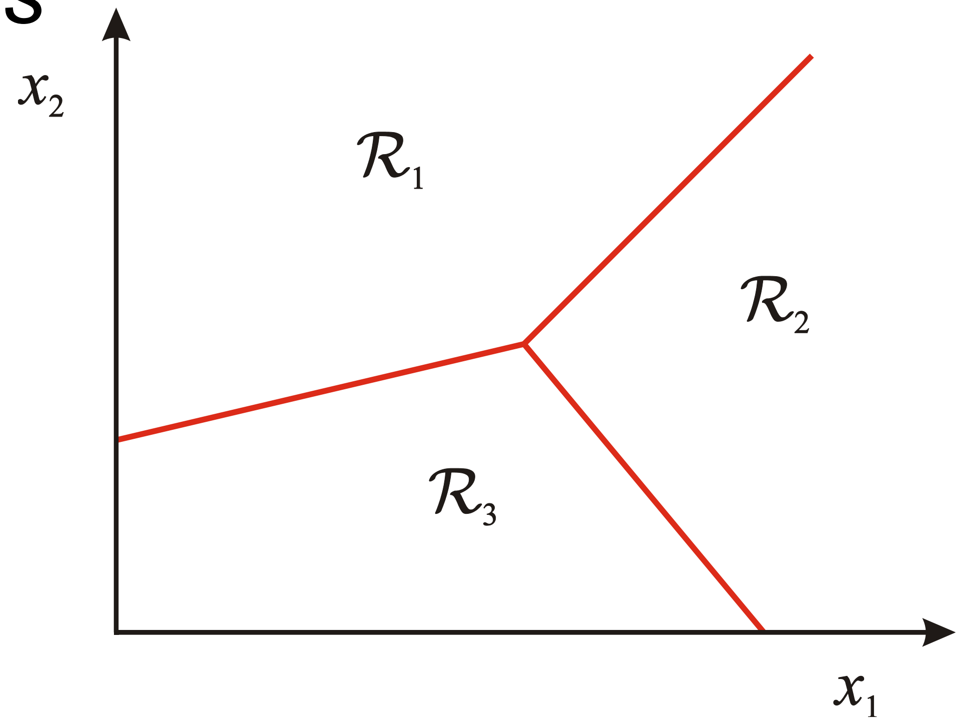
Contains a motorbike



Slide credit: L. Lazebnik

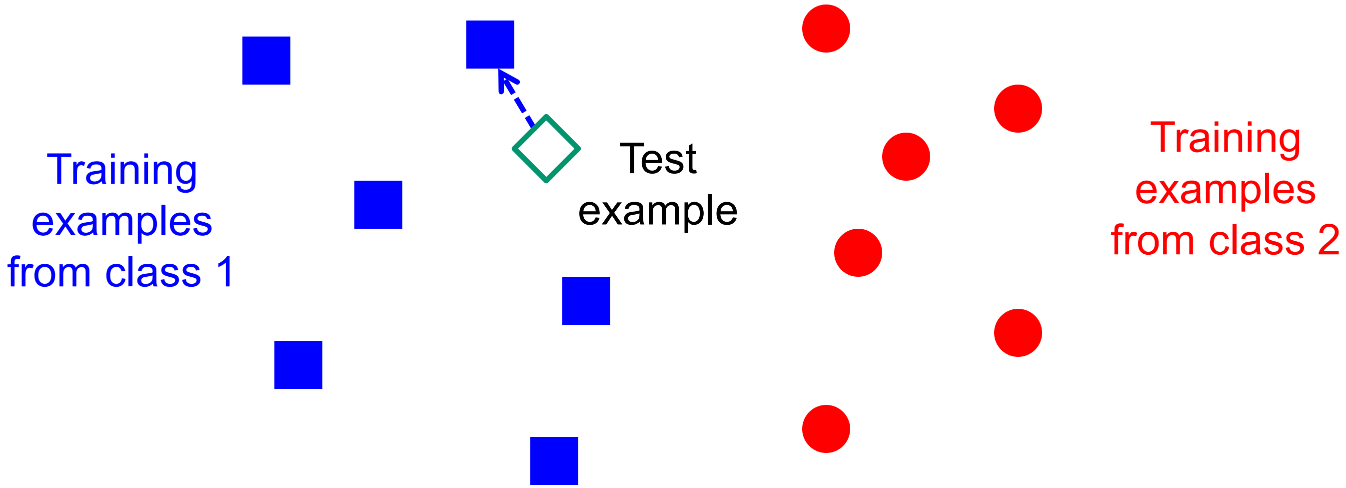
Classification

- Assign input vector to one of two or more classes
- Any decision rule divides input space into *decision regions* separated by *decision boundaries*



Slide credit: L. Lazebnik

Classifiers: Nearest neighbor



f(**x**) = label of the training example nearest to **x**

- All we need is a distance function for our inputs
- No training required!

Linear Regression

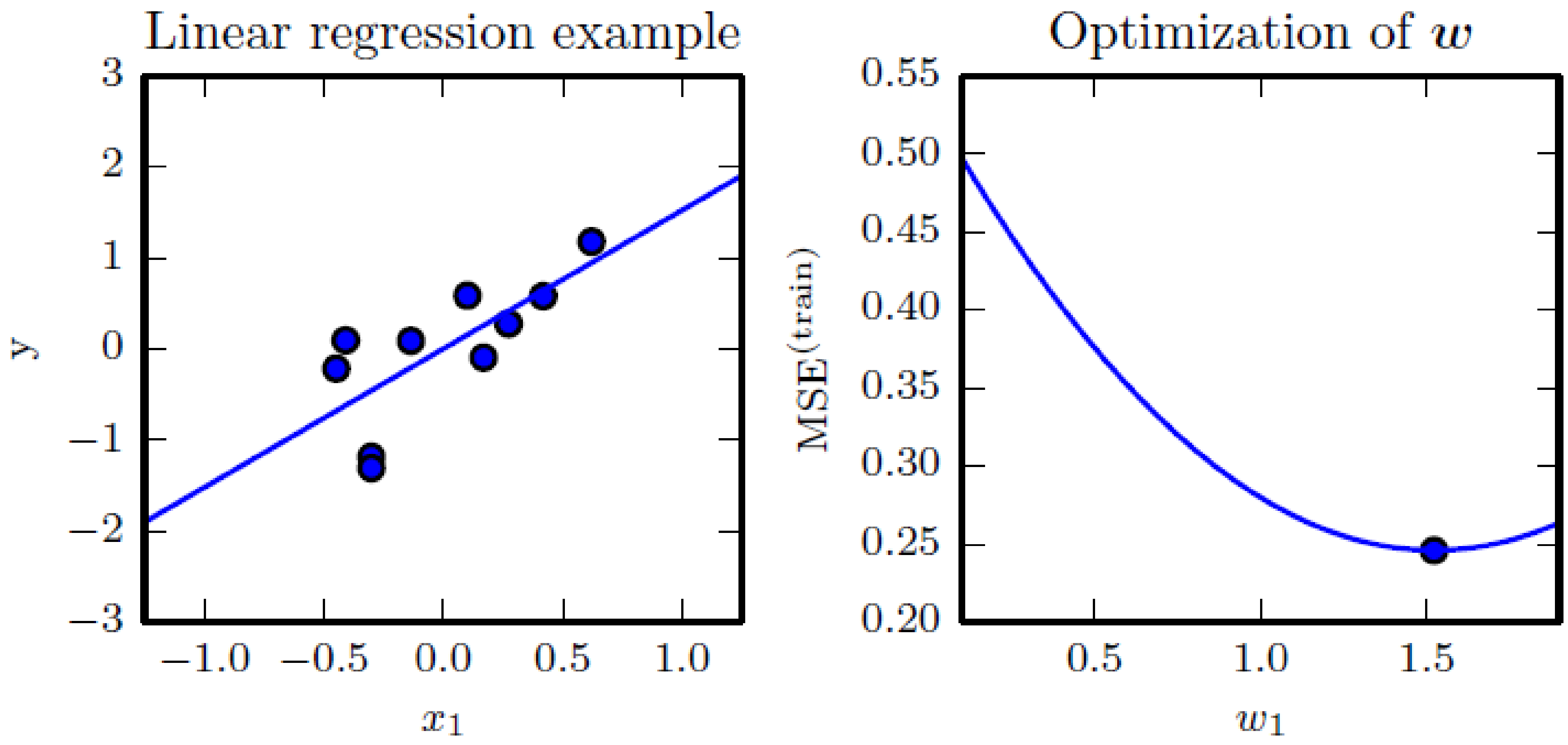


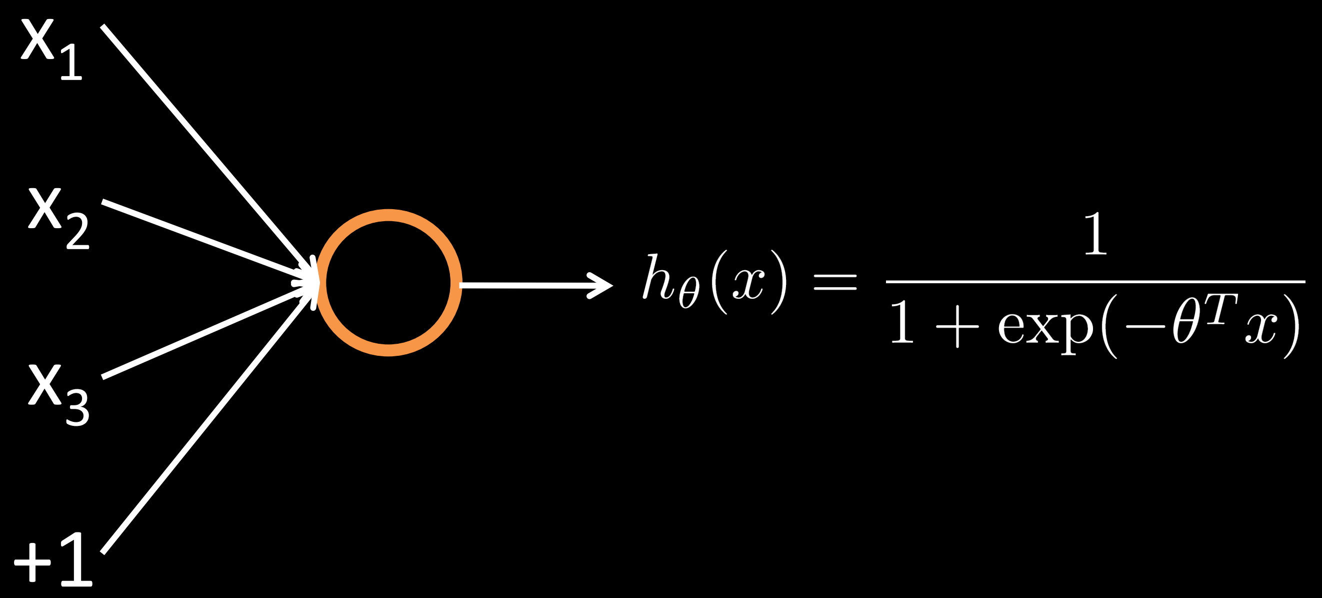
Figure 5.1

Logistic regression has a learned parameter vector θ.
On input x, it outputs:

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

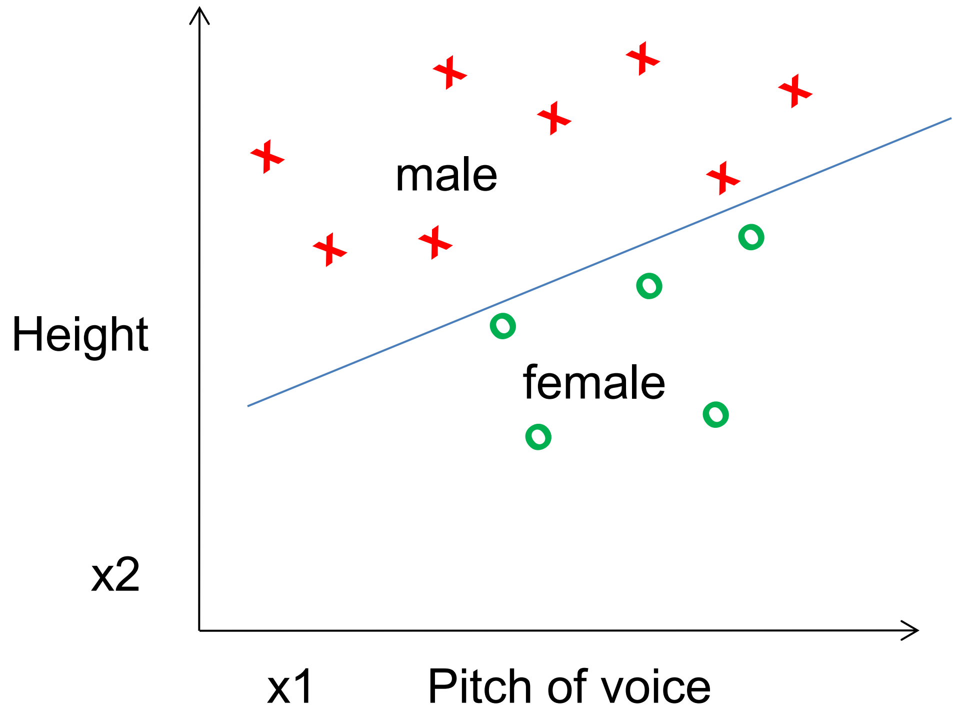
where $\sigma(z) = 1/(1+\exp(-z))$

Draw a logistic regression unit as:



Classifiers: Logistic Regression

Maximize likelihood of label given data, assuming a log-linear model



Comparison

assuming x in {0 1}

	Learning Objective	Training	Inference
Naïve Bayes	$\text{maximize} \sum_i \left[\sum_j \log P(x_{ij} \mid y_i; \theta_j) + \log P(y_i; \theta_0) \right]$	$\theta_{kj} = \frac{\sum_i \delta(x_{ij} = 1 \wedge y_i = k) + r}{\sum_i \delta(y_i = k) + Kr}$	$\boldsymbol{\theta}_1^T \mathbf{x} + \boldsymbol{\theta}_0^T (1 - \mathbf{x}) > 0$ where $\theta_{1j} = \log \frac{P(x_j = 1 \mid y = 1)}{P(x_j = 1 \mid y = 0)}$, $\theta_{0j} = \log \frac{P(x_j = 0 \mid y = 1)}{P(x_j = 0 \mid y = 0)}$
Logistic Regression	$\text{maximize} \sum_i \log(P(y_i \mid \mathbf{x}, \boldsymbol{\theta})) + \lambda \lVert \boldsymbol{\theta} \rVert$ where $P(y_i \mid \mathbf{x}, \boldsymbol{\theta}) = 1/(1 + \exp(-y_i \boldsymbol{\theta}^T \mathbf{x}))$	Gradient ascent	$\boldsymbol{\theta}^T \mathbf{x} > 0$
Linear SVM	$\text{minimize } \lambda \sum_i \xi_i + \frac{1}{2} \lVert \boldsymbol{\theta} \rVert$ such that $y_i \boldsymbol{\theta}^T \mathbf{x} \geq 1 - \xi_i \quad \forall i$	Linear programming	$\boldsymbol{\theta}^T \mathbf{x} > 0$
Kernelized SVM	complicated to write	Quadratic programming	$\sum_i y_i \alpha_i K(\hat{\mathbf{x}}_i, \mathbf{x}) > 0$
Nearest Neighbor	most similar features → same label	Record data	y_i where $i = \underset{i}{\text{argmin}}\ K(\hat{\mathbf{x}}_i, \mathbf{x})$

Slide credit: D. Hoiem

What to Remember about Classifiers

- No free lunch: machine learning algorithms are tools, not dogmas
- Try simple classifiers first
- Better to have smart features and simple classifiers than simple features and smart classifiers
- Use increasingly powerful classifiers with more training data (bias-variance tradeoff)

Slide credit: D. Hoiem

Machine Learning Problems

	Supervised Learning	*Unsupervised Learning*
Discrete	classification or categorization	clustering
Continuous	regression	$\boxed{\text{dimensionality reduction}}$

Dimensionality Reduction

Encoder or ***Projector*** operator

$$R^p \rightarrow R^m,\ m << p$$

Data space

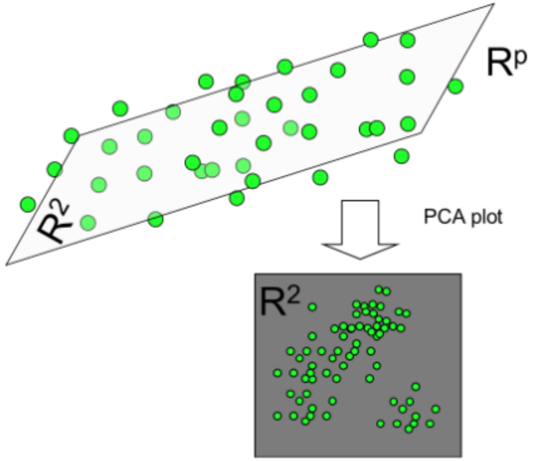
Some space*,**,***

*can have complex geometry and topology

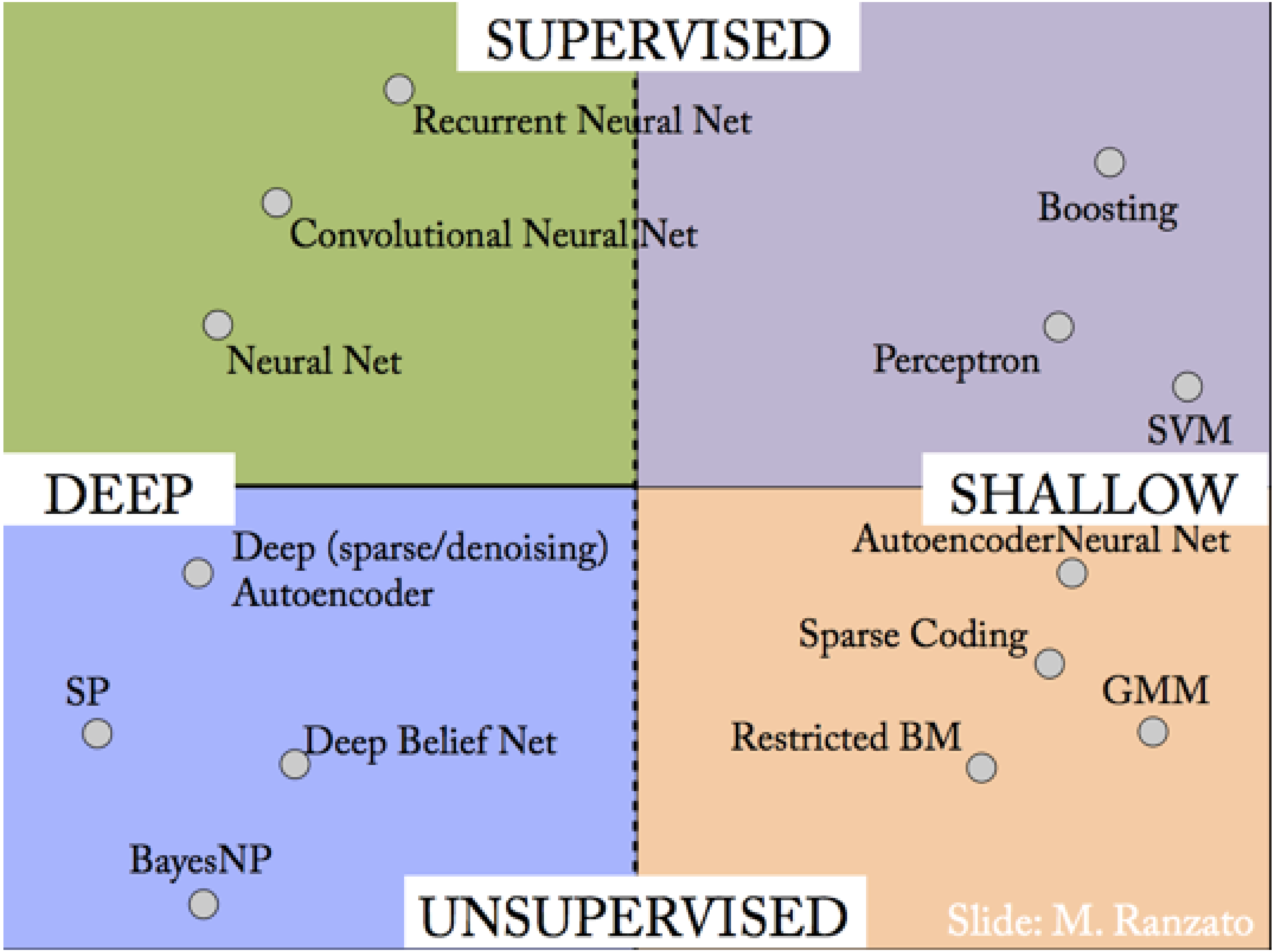
**does not have to be part of R^p

***simplest case: m-dimensional linear subspace in R^p

$$R^p \rightarrow R^m,\ m = 1,2,3$$



- Dimensionality reduction examples:
  - PCA, ICA,
  - Isomap,
  - Locally linear embedding (LLE)
  - Multidimensional scaling
  - t-distributed stochastic nearest neighbor embedding (t-SNE)
  - Neural approaches


SUPERVISED
Recurrent Neural Net
Convolutional Neural Net
Neural Net
Boosting
Perceptron
SVM
DEEP
SHALLOW
Deep (sparse/denoising) Autoencoder
SP
Deep Belief Net
BayesNP
AutoencoderNeural Net
Sparse Coding
GMM
Restricted BM
UNSUPERVISED
Slide: M. Ranzato

Some Areas We Will Study Together

- Representation learning
- Generative models
- Variational inference
- Reinforcement learning
- Recurrence and temporal learning
- A small dose of brain-inspired computing
- Diffusion models
- Language models, transformers

Questions?

References

- Slides/content were adapted from:
  - "Deep Learning" (Machine Learning Basics, Chapter 5, Goodfellow et al., 2016)
  - "An Overview of Machine Learning" (Yi-Fan Chang, 2011)
  - "CSE 446 Machine Learning" (intro) (Pedro Domingos)
  - "Feature learning for image classification" (Kai Yu and Andrew Ng)
  - "Machine Learning" (Computer Vision) James Hays, Brown
- Andrew Ng's Machine Learning course/lectures:
  - http://openclassroom.stanford.edu/MainFolder/CoursePage.php?course=MachineLearning
- Data Mining textbook : "Data Mining: Concepts and Techniques, Third Edition (The Morgan Kaufmann Series in Data Management Systems)" Han et al. 2011

Some Machine Learning References

- General
  - Tom Mitchell, *Machine Learning*, McGraw Hill, 1997
  - Christopher Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995
- Adaboost (to learn about Boosting)
  - Friedman, Hastie, and Tibshirani, "Additive logistic regression: a statistical view of boosting", Annals of Statistics, 2000
- SVMs
  - http://www.support-vector.net/icml-tutorial.pdf

Slide credit: D. Hoiem