CSE 4850/6850
Introduction to Machine Learning

Instructor: AKM Kamrul Islam

aislam5@cs.gsu.edu

Dept. of Computer Science
Georgia State University

(Materials are highly adapted from different online sources)

Organizational info

- All up-to-date info is on the iCollege GSU
- Instructors

  - A.K.M. Kamrul Islam

- TA: Jinkun Han (email: hjinkun1@student.gsu.edu)
- See Syllabus for contact info, office hours, etc.
- Piazza would be used for questions / comments and likely for class quizzes. Make sure you are subscribed.

About Me

- A.K.M. Kamrul Islam (Russell)
- Graduate Research Assistant(PhD. Candidate)
- Origin: Bangladesh
- Teaching Experience: Assistant Professor (BD)
- Find Me: 1 Park Place, 640
- Office Hours: Friday 3-4:30 pm
- Interests:

  - Computational Genomics, Image Classification and Segmentation
  - Deep learning applications in biomedical images and genomics
  - Time Series Classification

Machine Learning in Everyday Life

Search Engine Rankings

Email Spam Filtering

Social Media Services

online Banking Services

Personal Assistants

Product Recommendations

Georgia State University

Terms to Touch

- Linear Models for Classification and Regression (liner regression,...)
- Directed Graphical Models (Bayesian networks, . . .)
- Mixture Models (EM Algorithm, . . .)
- Latent Linear Models (PCA, ICA, . . .)
- Sparse Linear Models (Lasso, Dictionary Learning, . . .)
- Kernel Methods (SVM, KDE, . . .)
- Neural networks and Deep Learning (MLP, CNN, . . .)

Please see class website for updated weekly schedule

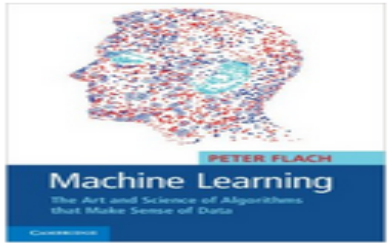
Textbook

Text

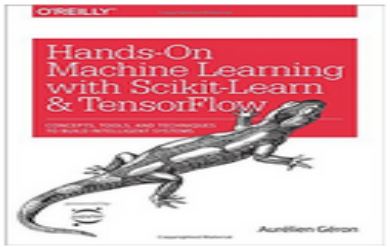
- Machine Learning, Tom Mitchell, McGraw Hill. (http://profsite.um.ac.ir/~monsefi/machine-learning/pdf/Machine-Learning-Tom-Mitchell.pdf)

Reference Book

- Abu-Mostafa, Yaser S. and Magdon-Ismail, Malik and Lin, Hsuan-Tien, Learning From Data, AMLBook.
- The elements of statistical learning. Data mining, inference, and prediction T. Hastie, R. Tibshirani, J. Friedman. (https://web.stanford.edu/~hastie/Papers/ESLII.pdf)
- Christopher Bishop. Pattern Recognition and Machine Learning.
- Richard O. Duda, Peter E. Hart, David G. Stork. Pattern Classification. Wiley.



Machine Learning: The Art and Science of Algorithms That Make Sense of Data by Peter Flach
Cambridge University Press



(OPTIONAL)
Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems by Aurélien Géron
1st Edition, O'Reilly Media, 2017

Resources

- Major journals/conferences: ICML, NIPS, UAI, ECML/PKDD, JMLR, MLJ, etc.
- Machine learning video lectures:

  http://videolectures.net/Top/Computer_Science/Machine_Learning/
- Machine Learning (Theory): http://hunch.net/
- LinkedIn ML groups: \Big Data" Scientist, etc.
- Women in Machine Learning:

  https://groups.google.com/forum/#!forum/women-in-machine-learning
- KDD nuggets http://www.kdnuggets.com/

Links

- http://cs231n.github.io/python-numpy-tutorial/
- https://www.seas.upenn.edu/~cis519/fall2018/assets/resources/python/python_introduction.html
- https://docs.python.org/3/tutorial/
- http://scipy.github.io/old-wiki/pages/Tentative_NumPy_Tutorial
- https://scikit-learn.org/stable/

Terminology

Machine Learning, Data Science, Data Mining, Data Analysis, Statistical Learning, Knowledge Discovery in Databases, Pattern Discovery.

Data Everywhere

1. **Google:** processes 24 peta bytes of data per day.
2. **Facebook:** 10 million photos uploaded every hour.
3. **Youtube:** 1 hour of video uploaded every second.
4. **Twitter:** 400 million tweets per day.
5. **Astronomy:** Satellite data is in hundreds of PB.
6. . . .
7. **"By 2020 the digital universe will reach 44 zettabytes..."**

The Digital Universe of Opportunities: Rich Data and the Increasing Value of the Internet of Things, April 2014.

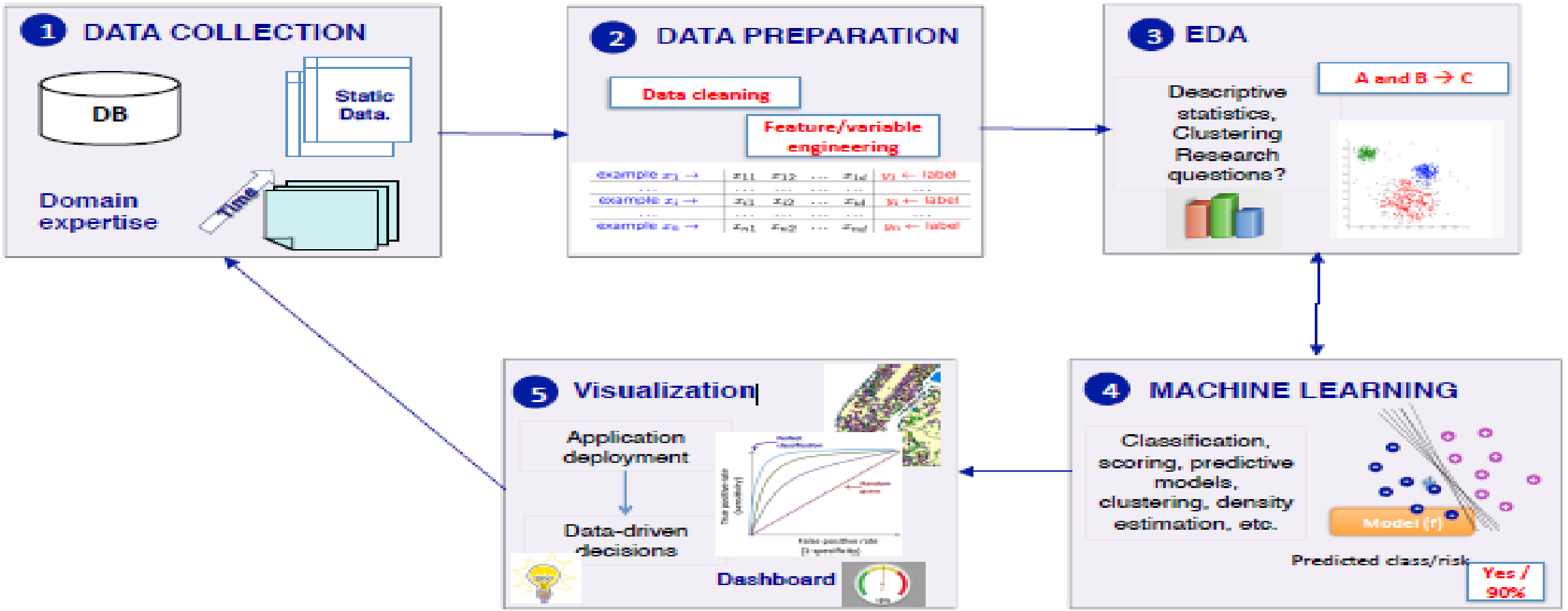
That's 44 trillion gigabytes!

Data Types

Data comes in different sizes and also flavors (types):

- ✓ Texts
- ✓ Numbers
- ✓ Clickstreams
- ✓ Graphs
- ✓ Tables
- ✓ Images
- ✓ Transactions
- ✓ Videos
- ✓ Some or all of the above!

Data Science Process



Interdisciplinary Field

Machine Learning Vs Statistics

Statistics:

- Hypothesis testing
- Experimental design
- Anova
- Linear regression
- Logistic regression
- GLM
- PCA

Machine Learning:

- Decision trees
- Rule induction
- Neural Networks
- SVMs
- Clustering method
- Association rules
- Feature selection
- Visualization
- Graphical models
- Genetic algorithm

http://statweb.stanford.edu/~jhf/ftp/dm-stat.pdf

What is Machine Learning?

"Learning is any process by which a system improves performance from experience."

- Herbert Simon

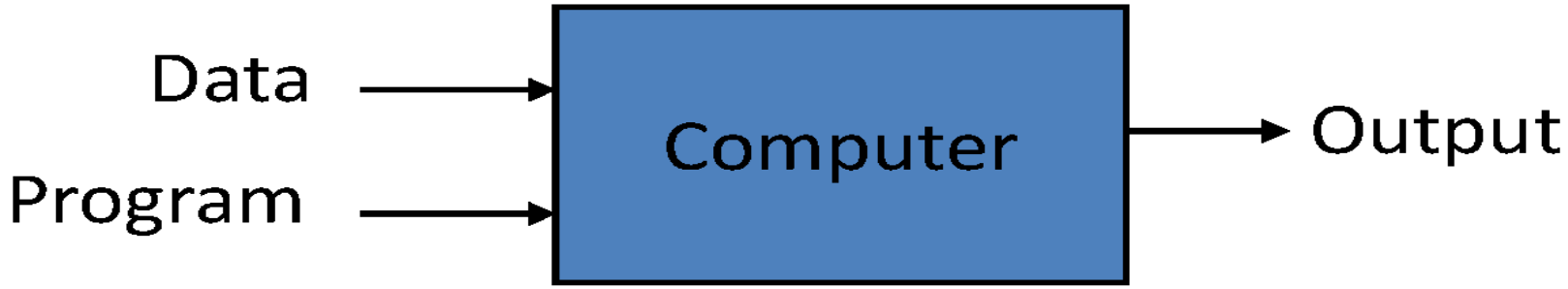
Definition by Tom Mitchell (1998):

Machine Learning is the study of algorithms that

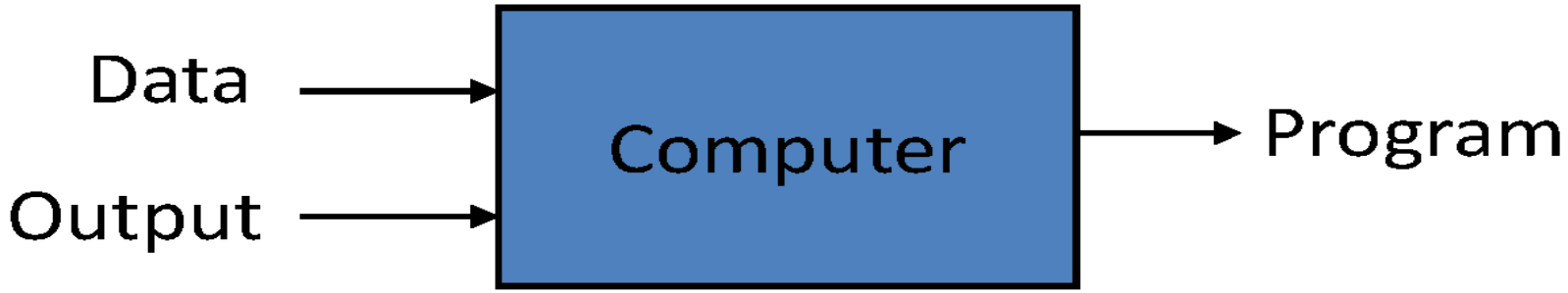
- improve their performance P
- at some task T
- with experience E.

A well-defined learning task is given by <P, T, E>.

Traditional Programming



Machine Learning



Defining the Learning Task

"Machine Learning is the study of algorithms that improve their performance (P) at some task (T) with experience (E)"

Q: Suppose your email program watches which emails you do or do not mark as spam, and based on that learns how to better filter spam. What is P, T, and E in this setting?[1]

- Classifying emails as spam or not spam
- Watching you label emails as spam or not spam
- The fraction of emails correctly classified as spam/not spam

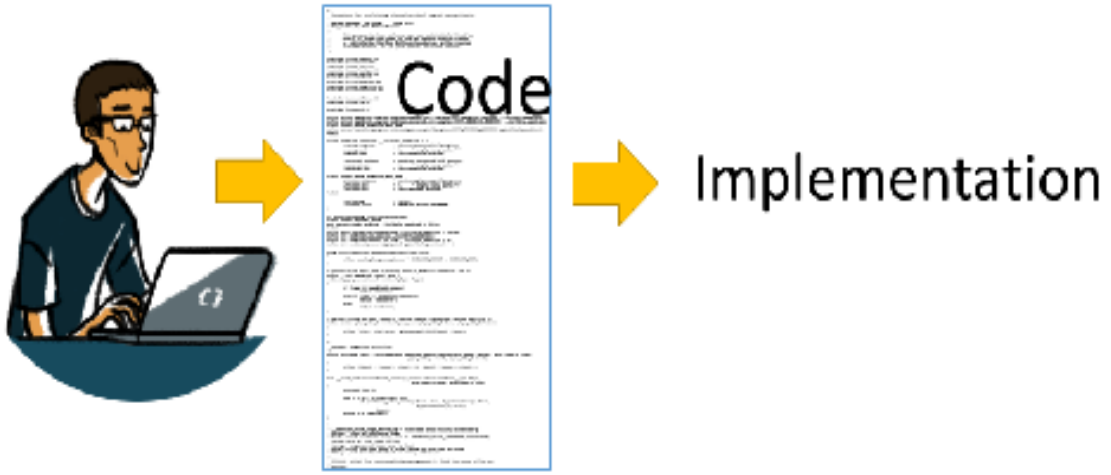
[1] Slide credit: Andrew Ng

Traditional Programming Approach – Simple Problem

How to we develop software?

- Specification:
  function f should return a real value that is three times the input
- Implementation:

```
function y = f(x)
    y = 3.0*x;
end
```



- Tests:

```
f(0.0) == 0.0, f(NaN) == NaN, f(1.0) == 3.0,
abs(f(1.000000000001) - 3.000000000003)<1e-12 ...
```

Machine Learning Approach

1. Assume a template for the function (model)

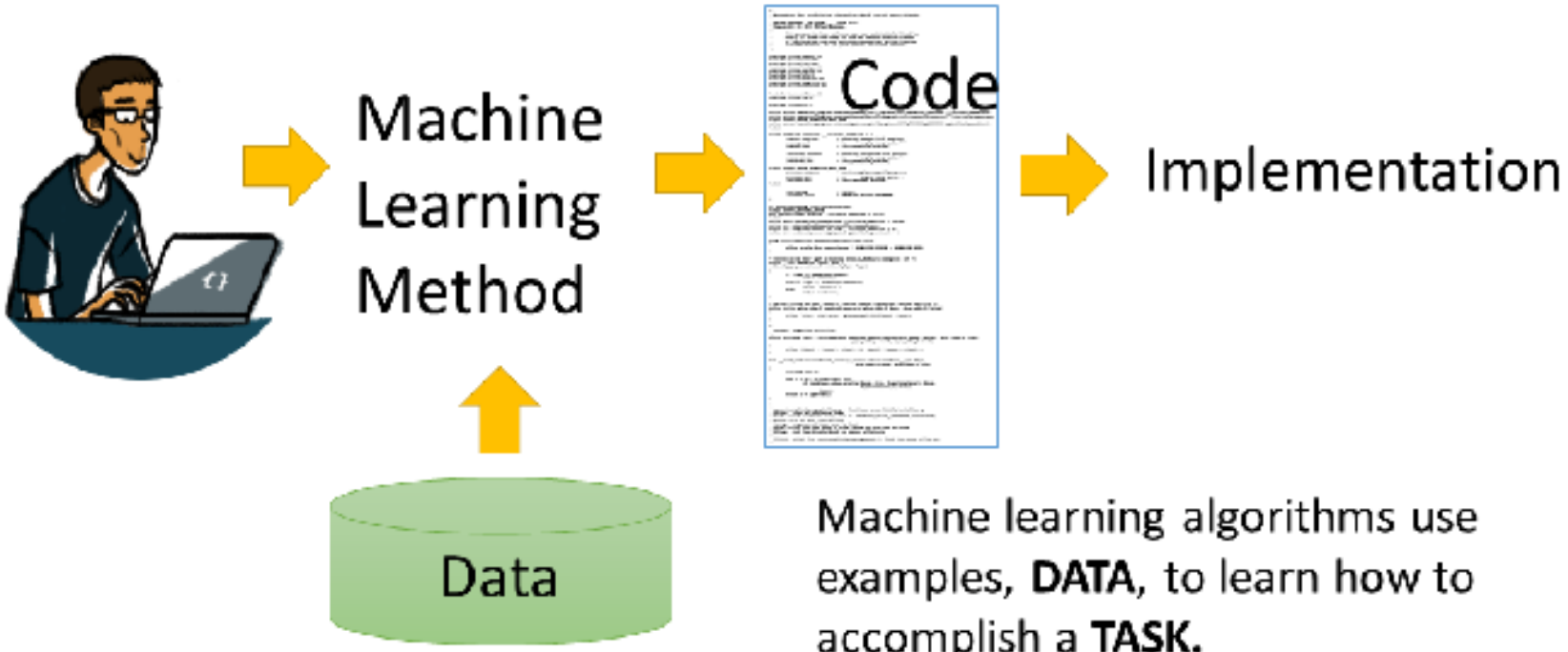
```
function y = f(x,beta)
    y = beta*x;
end
we call beta a parameter
```

2. Specify a cost $C(beta; Data)$ that tells you how well $f(x, beta)$ fits the Data

$$C(beta; Data) = \sum_{(x,y)\in Data} (y - f(x, beta))^2$$

3. Find beta for which cost $C(beta; Data)$ is the smallest, this process is called learning or training

Machine Learning Approach



Machine learning algorithms use examples, **DATA**, to learn how to accomplish a **TASK.**

Machine Learning Approach

- How well does your model perform on new data, data you have not seen during learning? How well your model **"generalize"**?
- If you get a new data instance, for example

  (No history of cancer, Smoker, Male)

  Can you assess how good is your throat cancer predictor?

1. Divide data into training set and testing set
2. Use training set as input to learning procedure to produce a predictor
3. Use testing set to evaluate performance of the resulting predictor

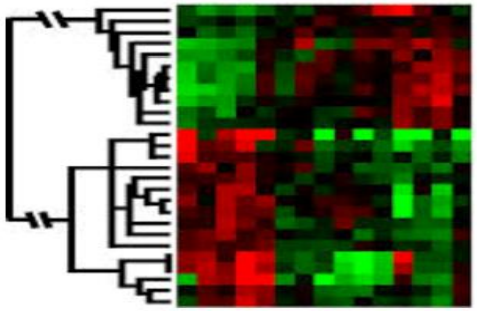
Common mistake to let test data bleed into training data

When Do We Use Machine Learning?

ML is used when:

- Human expertise does not exist (navigating on Mars)
- Humans can't explain their expertise (speech recognition)
- Models must be customized (personalized medicine)
- Models are based on huge amounts of data (genomics)



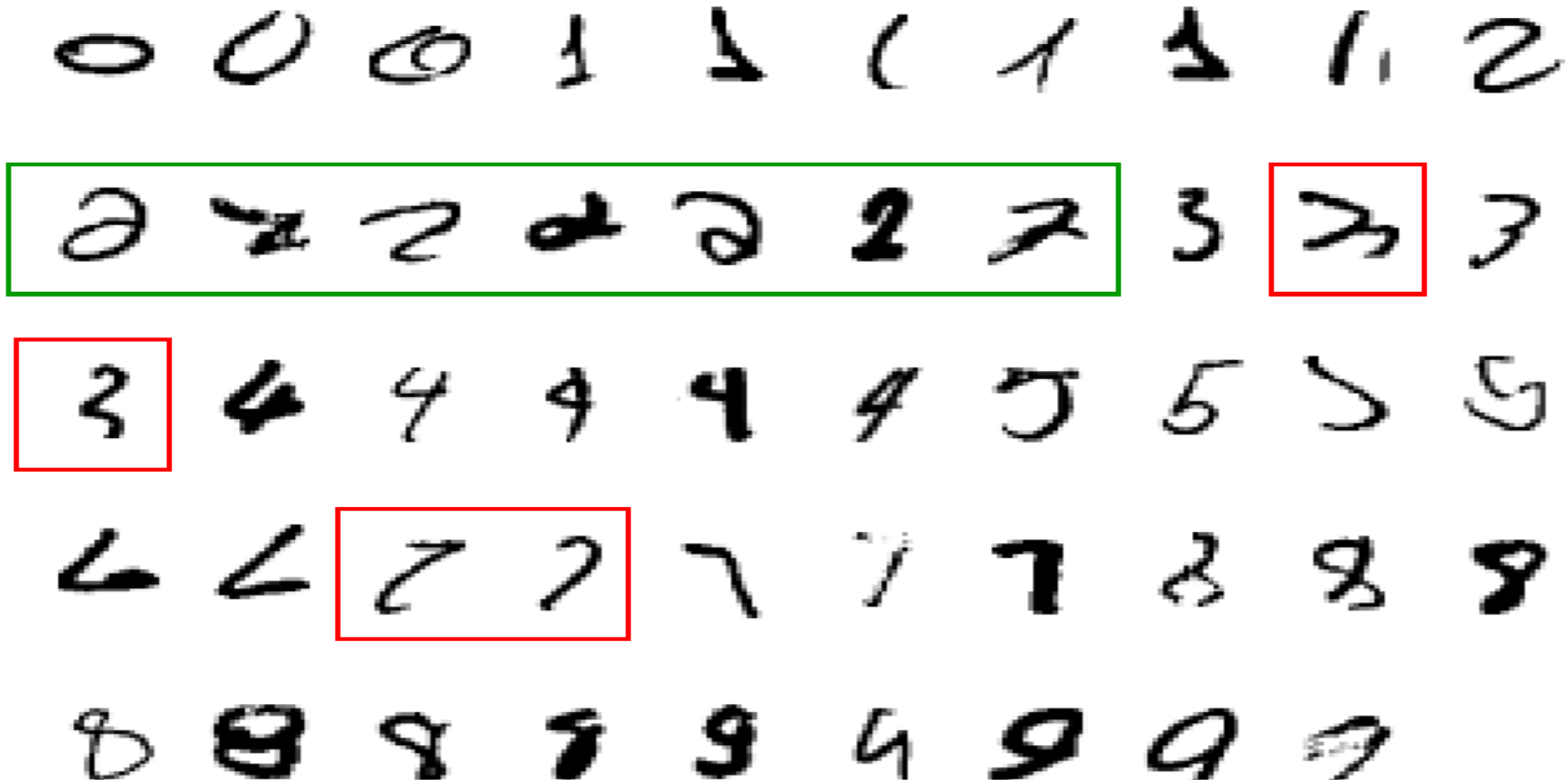



Learning isn't always useful:

- There is no need to "learn" to calculate payroll

A classic example of a task that requires machine learning:

It is very hard to say what makes a 2



Some more examples of tasks that are best solved by using a learning algorithm

- Recognizing patterns:
  - Facial identities or facial expressions
  - Handwritten or spoken words
  - Medical images
- Generating patterns:
  - Generating images or motion sequences
- Recognizing anomalies:
  - Unusual credit card transactions
  - Unusual patterns of sensor readings in a nuclear power plant
- Prediction:
  - Future stock prices or currency exchange rates

Sample Applications

- Web search
- Computational biology
- Finance
- E-commerce
- Space exploration
- Robotics
- Information extraction
- Social networks
- Debugging software
- [Your favorite area]

- Spam filtering
- Credit card fraud detection
- Digit recognition on checks, zip codes
- Detecting faces in images
- MRI image analysis
- Recommendation system
- Search engines
- Handwriting recognition
- Scene classification

Samuel's Checkers-Player

"Machine Learning: Field of study that gives computers the ability to learn without being explicitly programmed." -Arthur Samuel (1959)

Defining the Learning Task

Improve on task T, with respect to
performance metric P, based on experience E

T: Playing checkers
P: Percentage of games won against an arbitrary opponent
E: Playing practice games against itself

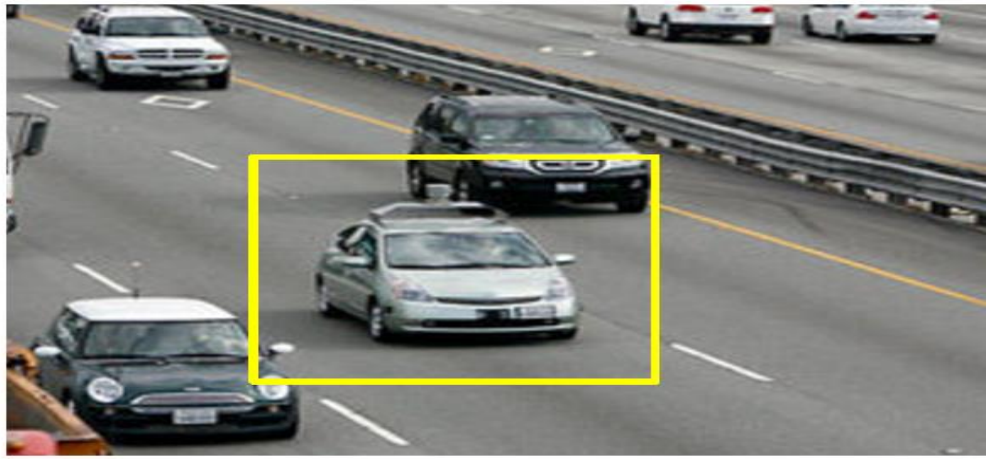
T: Recognizing hand-written words
P: Percentage of words correctly classified
E: Database of human-labeled images of handwritten words

T: Driving on four-lane highways using vision sensors
P: Average distance traveled before a human-judged error
E: A sequence of images and steering commands recorded while observing a human driver.

T: Categorize email messages as spam or legitimate.
P: Percentage of email messages correctly classified.
E: Database of emails, some with human-given labels

State of the Art Applications of Machine Learning

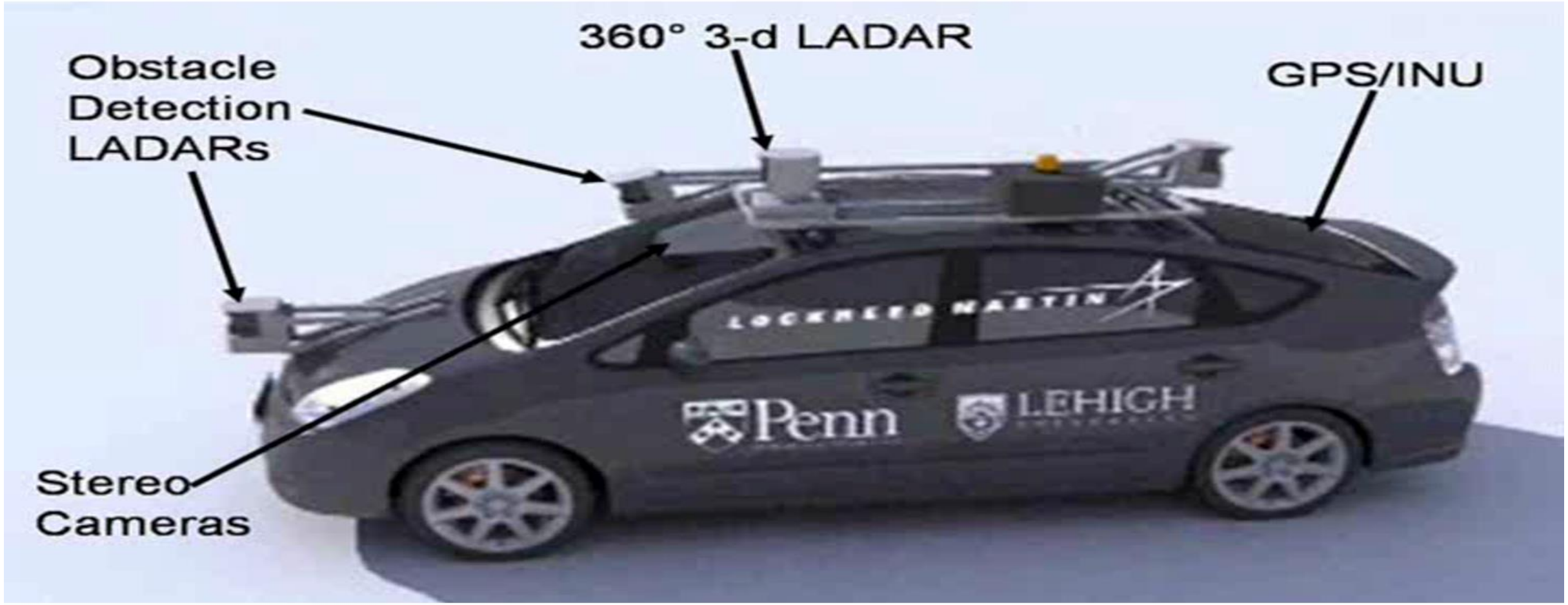
Autonomous Cars



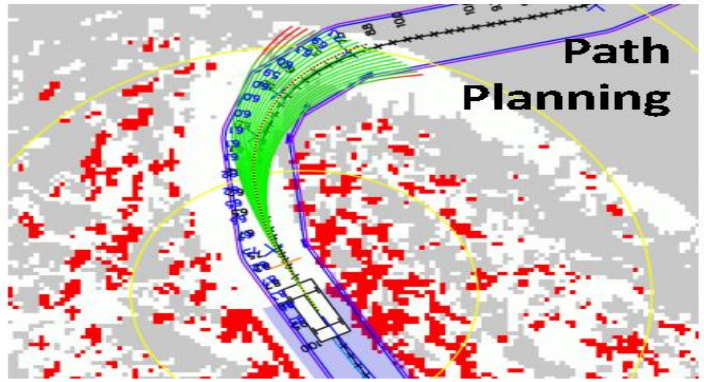
- Nevada made it legal for autonomous cars to drive on roads in June 2011
- As of 2013, four states (Nevada, Florida, California, and Michigan) have legalized autonomous cars

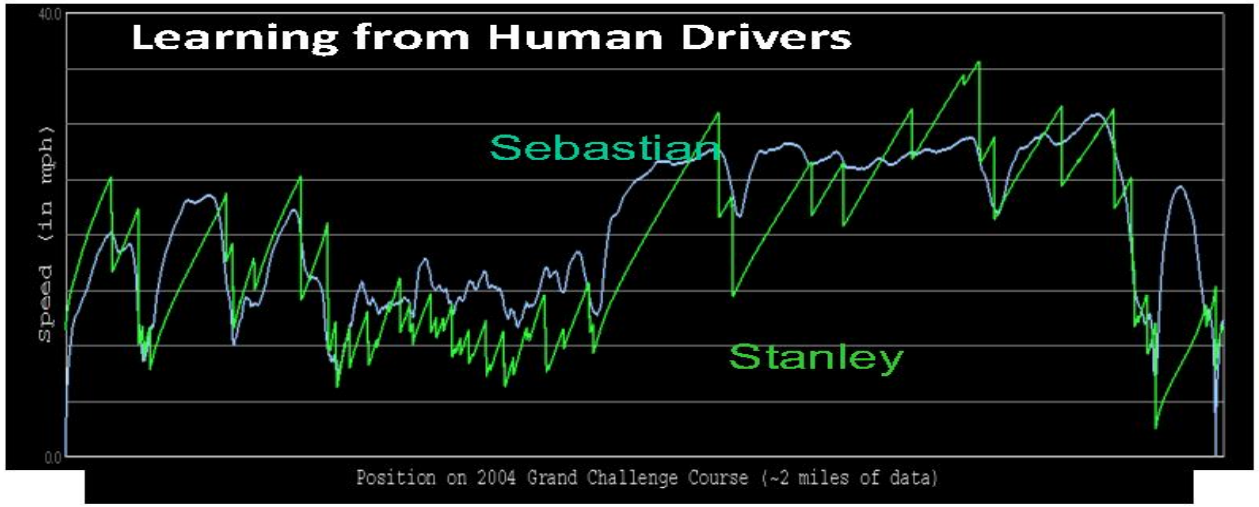
Penn's Autonomous Car →
(Ben Franklin Racing Team)

Autonomous Car Sensors



Autonomous Car Technology





Images and movies taken from Sebastian Thrun's multimedia website.

Deep Learning in the Headlines

BUSINESS NEWS

MIT Technology Review

Is Google Cornering the Market on Deep Learning?

A cutting-edge corner of science is being wooed by Silicon Valley, to the dismay of some academics.

By Antonio Regalado on January 29, 2014

How much are a dozen deep-learning researchers worth? Apparently, more than $400 million.

This week, Google reportedly paid that much to acquire DeepMind Technologies, a startup based in

BloombergBusinessweek
Technology

Acquisitions

The Race to Buy the Human Brains Behind Deep Learning Machines

By Ashlee Vance | January 27, 2014

intelligence projects. "DeepMind is bona fide in terms of its research capabilities and depth," says Peter Lee, who heads Microsoft Research.

According to Lee, Microsoft, Facebook (FB), and Google find themselves in a battle for deep learning talent. Microsoft has gone from four full-time deep learning experts to 70 in the past three years. "We would have more if the talent was there to

WIRED GEAR SCIENCE ENTERTAINMENT BUSINESS SECURITY DESIG

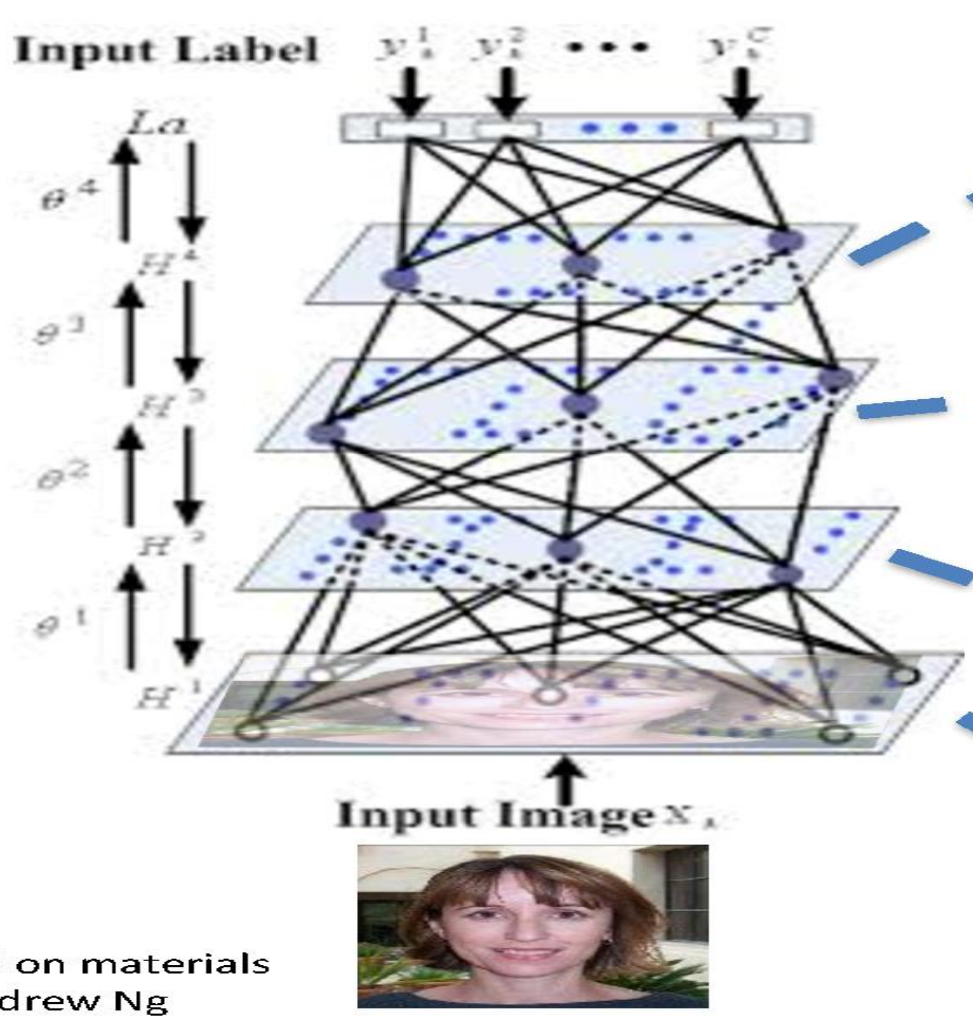
INNOVATION INSIGHTS | community content | featured

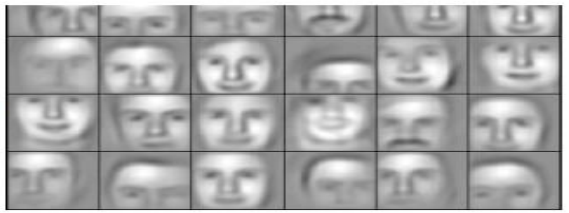
Deep Learning's Role in the Age of Robots

BY JULIAN GREEN, JETPAC 05.02.14 2:56 PM

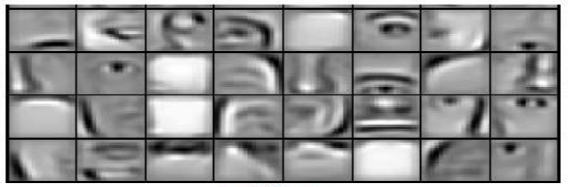
Georgia State University

Deep Belief Net on Face Images

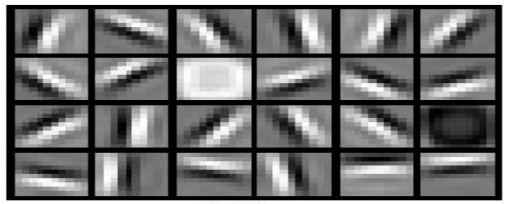




object models



object parts (combination of edges)

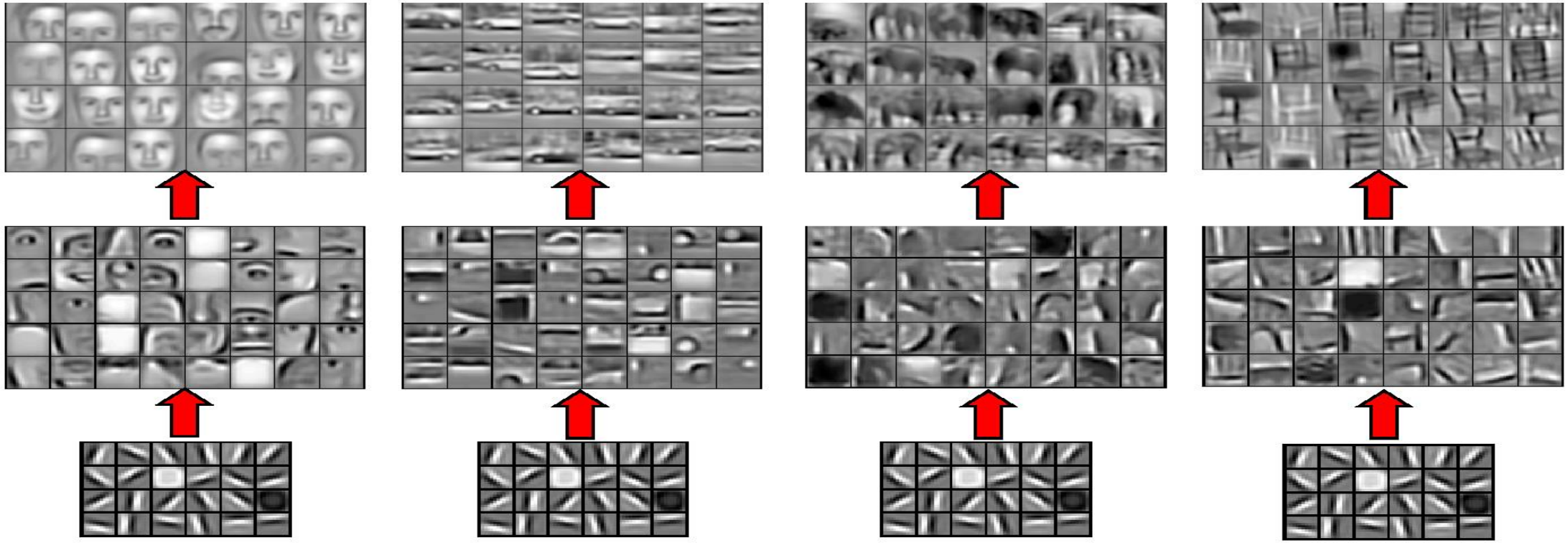


edges

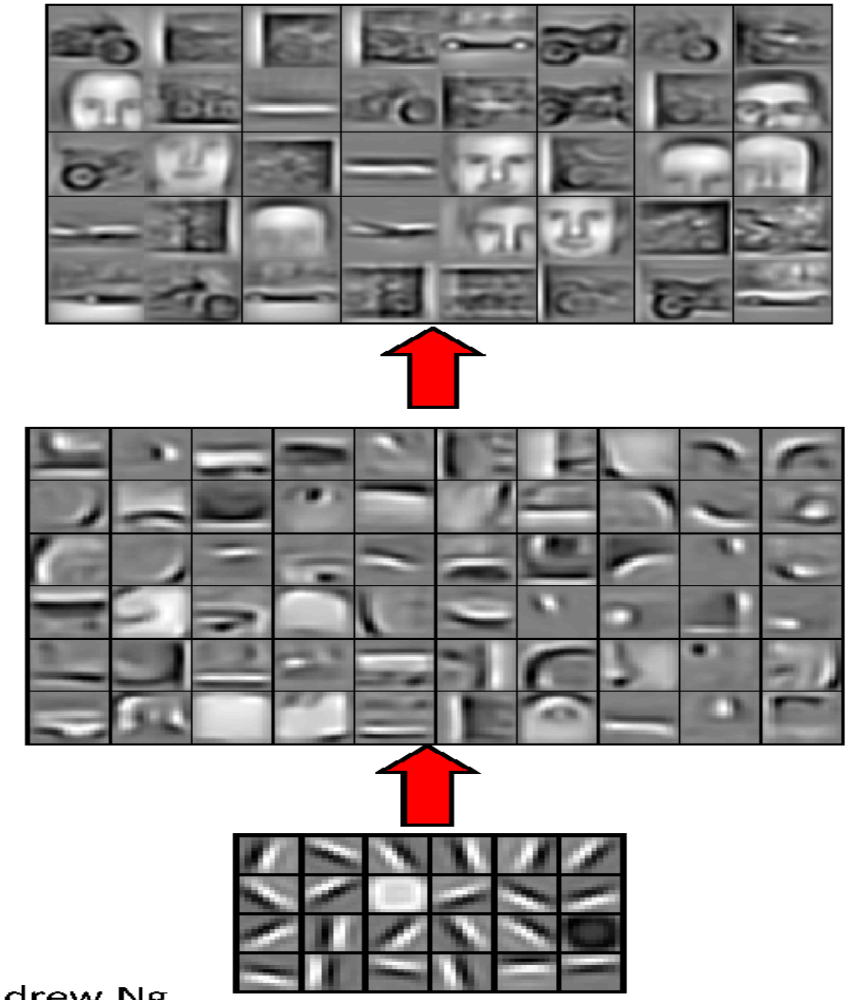
pixels

Based on materials by Andrew Ng

Learning of Object Parts



Training on Multiple Objects



Trained on 4 classes (cars, faces, motorbikes, airplanes).

Second layer: Shared-features and object-specific features.

Third layer: More specific features.

Slide credit: Andrew Ng

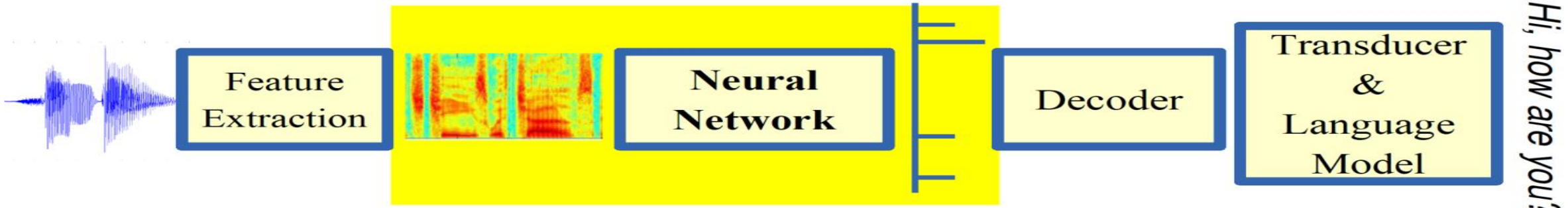
Scene Labeling via Deep Learning

Inference from Deep Learned Models

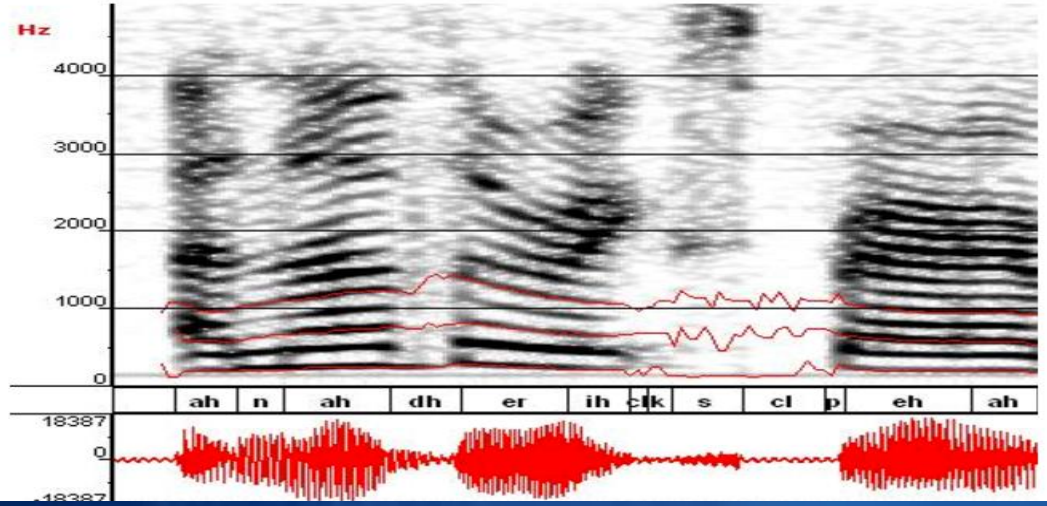
Generating posterior samples from faces by "filling in" experiments
(cf. Lee and Mumford, 2003). Combine bottom-up and top-down inference.

Machine Learning in Automatic Speech Recognition

A Typical Speech Recognition System



ML used to predict of phone states from the sound spectrogram



Deep learning has state-of-the-art results

# Hidden Layers	1	2	4	8	10	12
Word Error Rate %	16.0	12.8	11.4	10.9	11.0	11.1

Baseline GMM performance = 15.4%

[Zeiler et al. "On rectified linear units for speech recognition" ICASSP 2013]

Impact of Deep Learning in Speech Technology

Slide credit: Li Deng, MS Research

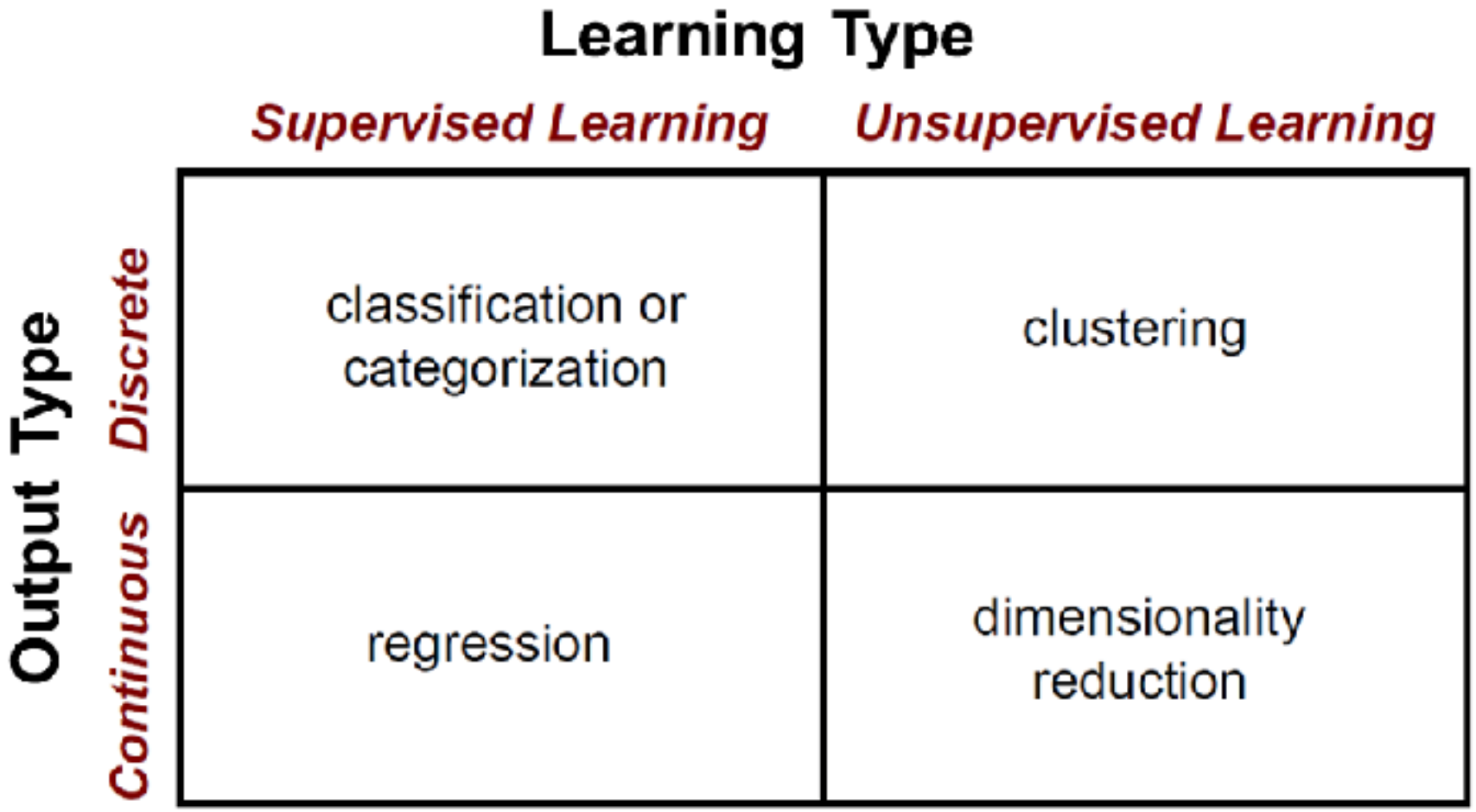
Types of Learning

Types of Learning

- **Supervised (inductive) learning**
  - Given: training data + desired outputs (labels)
- **Unsupervised learning**
  - Given: training data (without desired outputs)
- **Semi-supervised learning**
  - Given: training data + a few desired outputs
- **Reinforcement learning**
  - Rewards from sequence of actions

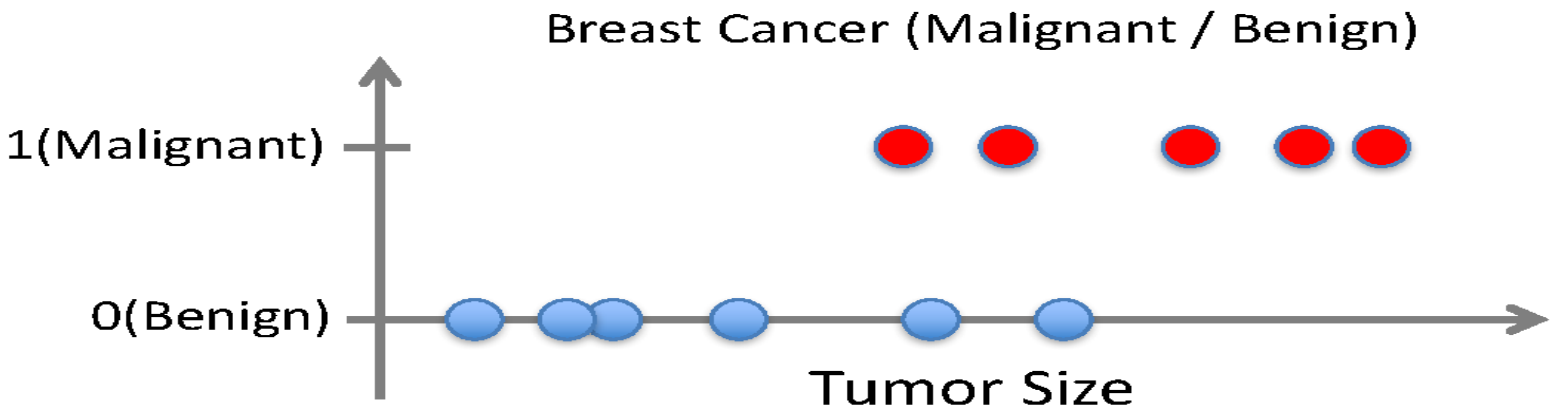
Based on slide by Pedro Domingos

Types of Machine Learning



Supervised Learning: Classification

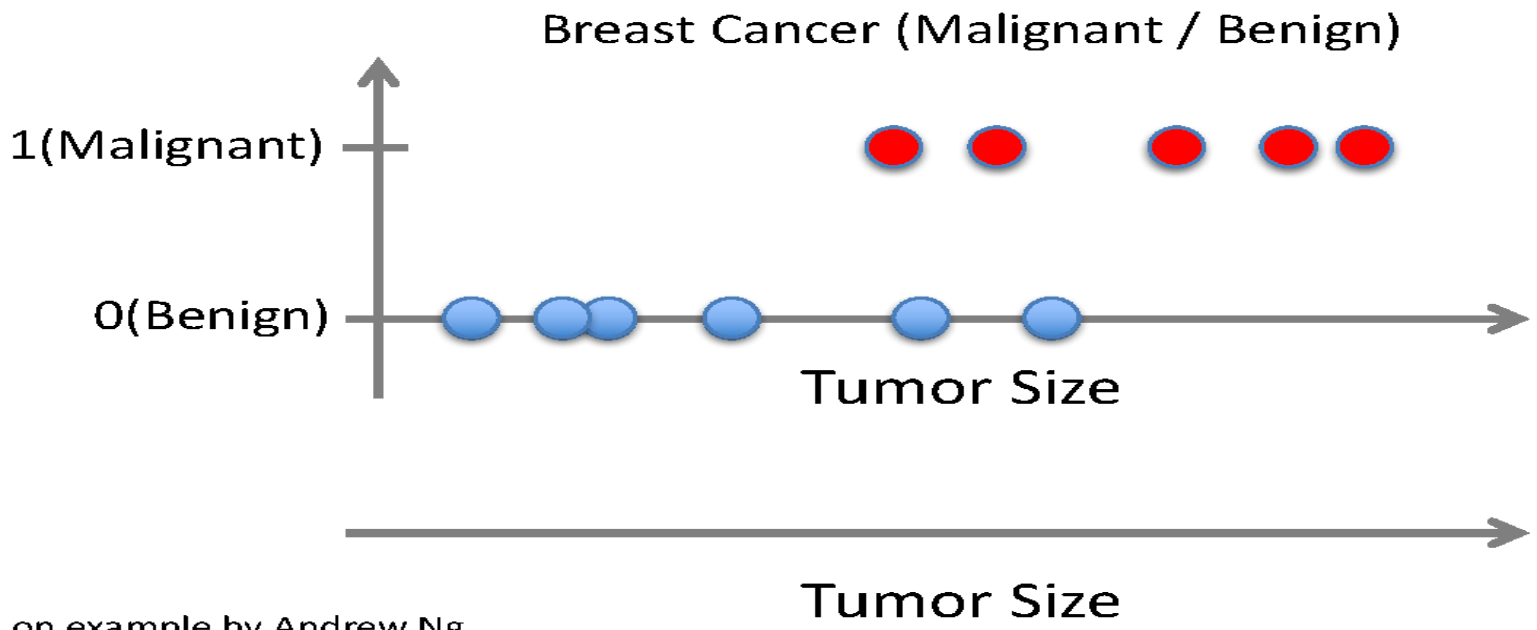
- Given (x_1, y_1), (x_2, y_2), …, (x_n, y_n)
- Learn a function $f(x)$ to predict y given x
  - y is categorical == classification



Based on example by Andrew Ng

Supervised Learning: Classification

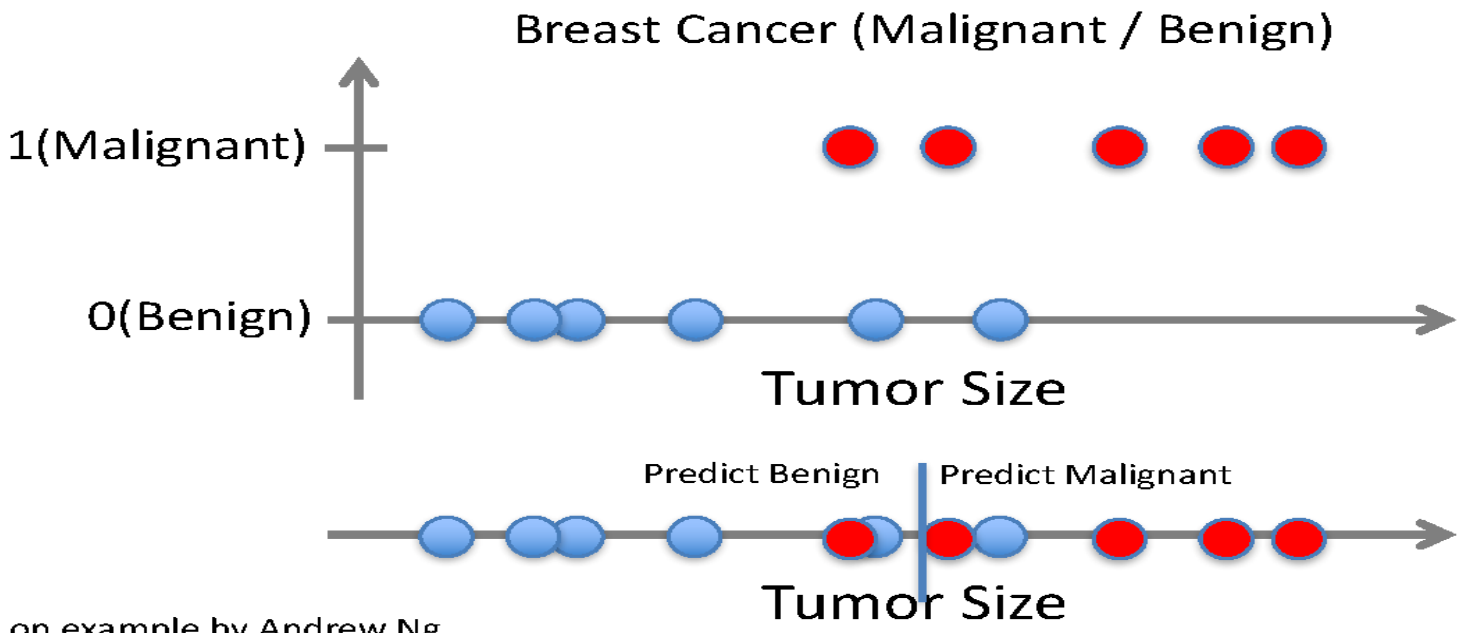
- Given $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$
- Learn a function $f(x)$ to predict y given x
  - y is categorical == classification



Based on example by Andrew Ng

Supervised Learning: Classification

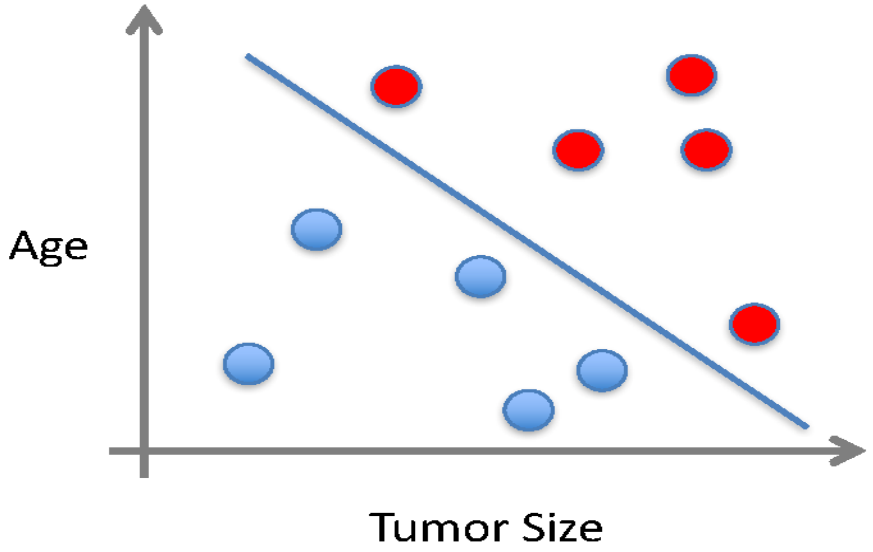
- Given $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$
- Learn a function $f(x)$ to predict y given x
  - y is categorical == classification



Based on example by Andrew Ng

Supervised Learning

- x can be multi-dimensional
  - Each dimension corresponds to an attribute



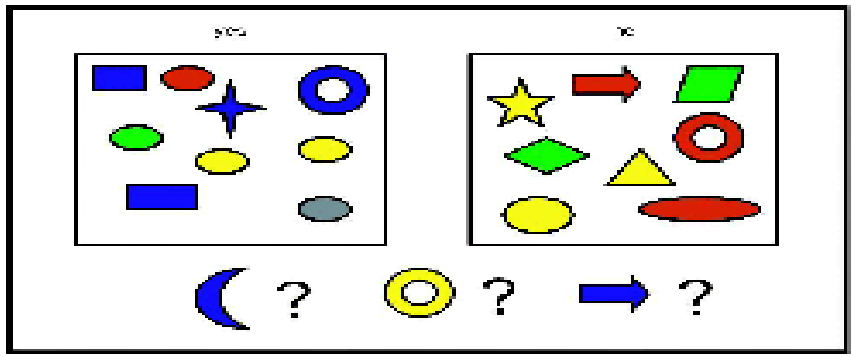
- Clump Thickness
- Uniformity of Cell Size
- Uniformity of Cell Shape

...

Based on example by Andrew Ng

Supervised Learning : Classification

- Given a training set $D = (\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)$, where $y \in \{1, \ldots, C\}$, with C being the number of classes



Some labeled training examples of colored shapes, along with 3 unlabeled test cases.

D features (attributes)

Color	Shape	Size (cm)	Label
Blue	Square	10	1
Red	Ellipse	2.4	1
Red	Ellipse	20.7	0

N cases

Representing the training data as an $N \times D$ design matrix. Row i represents the feature vector x_i. The last column is the label, $y_i \in \{0, 1\}$ ($C = 2$; binary classification).

- Estimate function f given D, then predict using $\hat{y} = \hat{f}(\mathbf{x})$

Q: What label should we assign to the yellow cirlce in the test data?

Machine Learning : Probabilistic View

The need for probabilistic predictions

- To handle ambiguous cases, it is desirable to return a probability
- Denote the probability distribution given $\mathbf{x}$ and and D by $P(y|\mathbf{x}, D)$
- For example, for the yellow circle in last example:

$$P(y = 0|\mathbf{x}, D) = 0.6 \text{ and } P(y = 1|\mathbf{x}, D) = 0.4$$

- Given a probabilistic output, we can always compute our best guess as to the "true label" using:

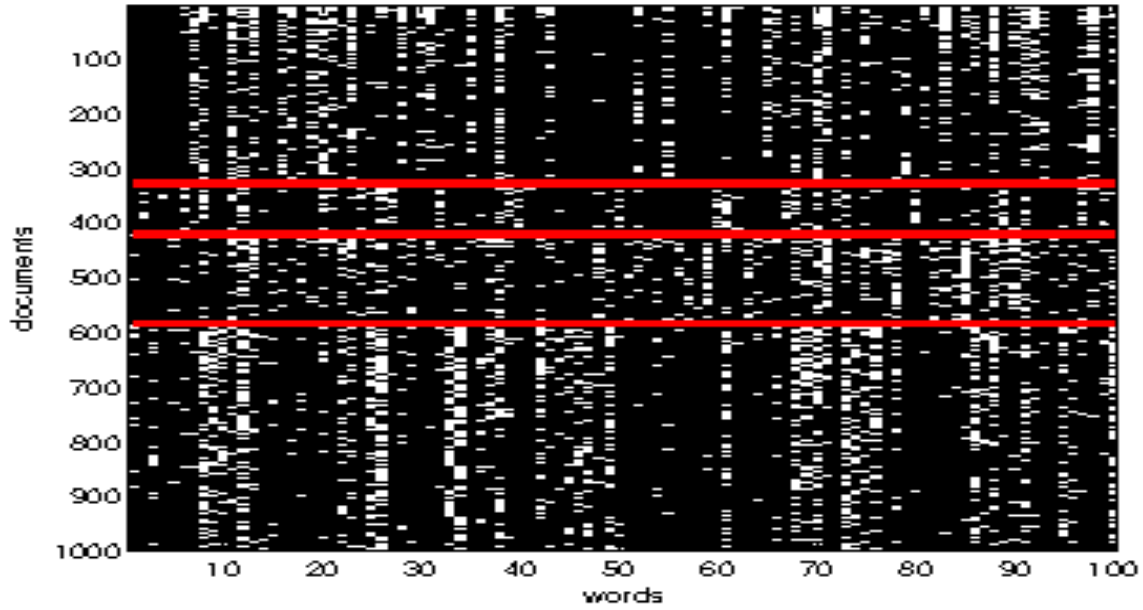
$$\hat{y} = \hat{f}(x) = \underset{c \in C}{\text{argmax}} P(y = c|\mathbf{x}, D)$$

- This is also known as a MAP (maximum a posteriori) estimate
- Continuous predictions will enable feedback on how to adjust them
- This will enable use of optimization and to quantify uncertainty

Supervised Learning : Classification Example

Email spam filtering

- The goal is to classify email message into spam $y = 1$ or $y = 0$
- Bag of words representation for variable-length documents

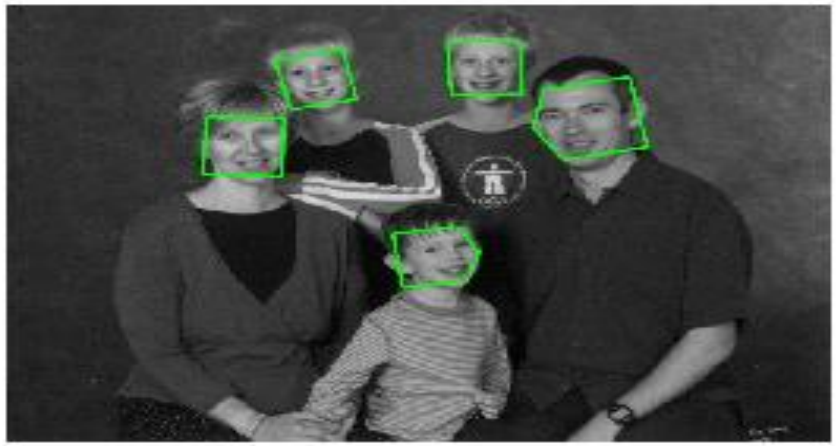


Each row is a document represented as a bag-of-words bit vector. There are subsets of words whose presence or absence is indicative of the class.

Supervised Learning : Classification Example

Face detection and recognition

- Find objects (faces) within an image is called object (face) detection
  - Sliding window detector is an example for solving this problem
- Then, face recognition can be used to estimate the person identity
  - Features used are likely to be different than in face detection problem

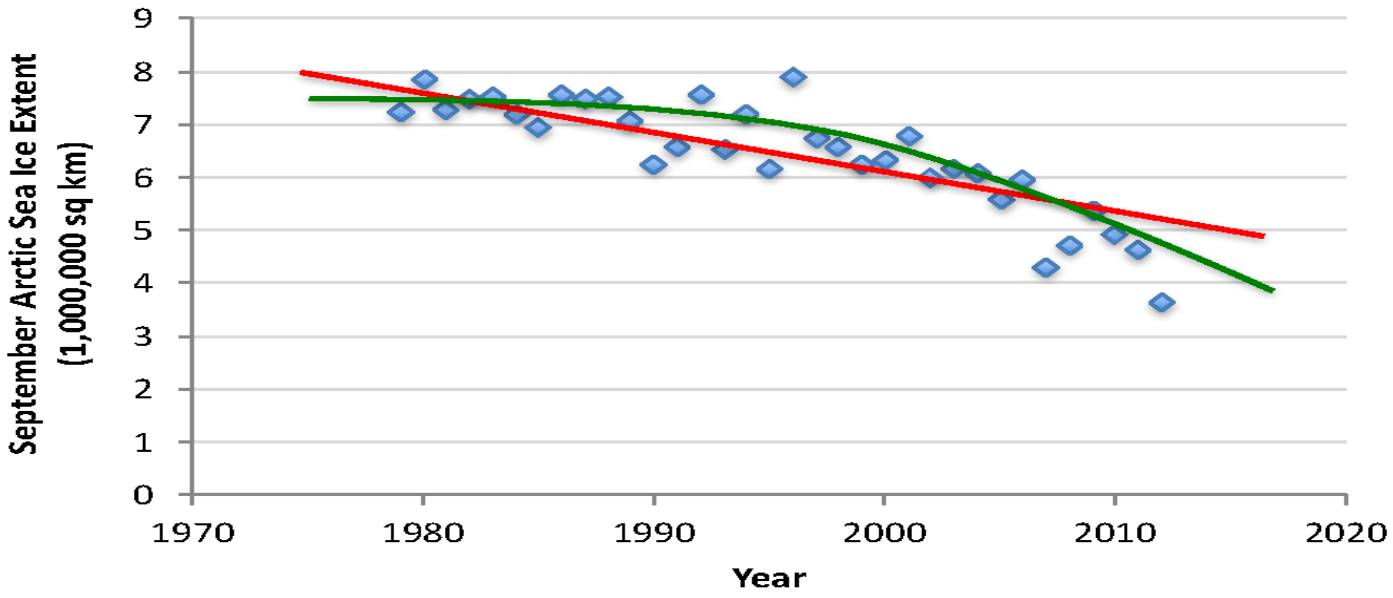


Face detector detected five faces at different poses.

Face database used to train a face recognition classifier.

Supervised Learning: Regression

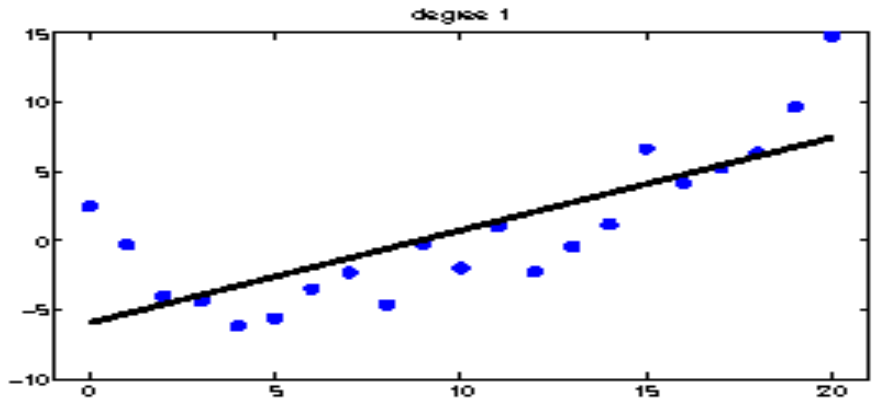
- Given $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$
- Learn a function $f(x)$ to predict y given x
  - y is real-valued == regression



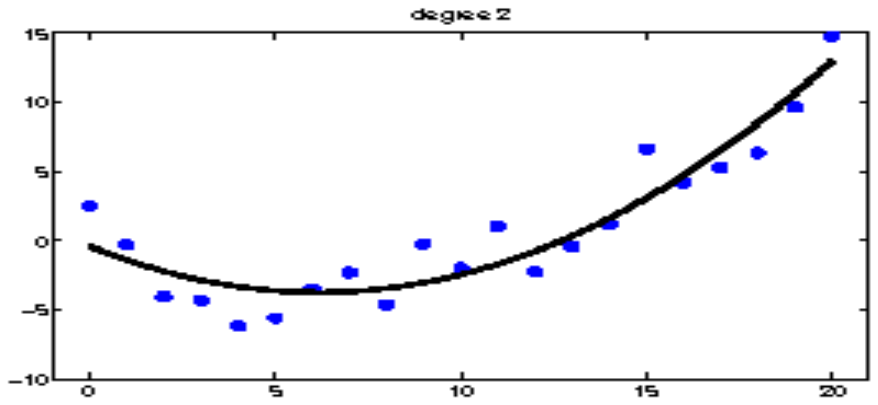
Data from G. Witt. Journal of Statistics Education, Volume 21, Number 1 (2013)

Supervised Learning : Regression

- Regression like classification except the response variable is continuous
- For a single input $x_i \in \mathbb{R}$, and a single response $y_i \in \mathbb{R}$



Linear regression on some 1d data



Same data with polynomial regression

Some examples of real-world regression problems

- Predict tomorrow's stock market price given current market conditions
- Predict the age of a viewer watching a given video on YouTube
- Predict the location of a robot arm given control signals

Supervised Learning : Regression

You're running a company, and you want to develop learning algorithms to address each of two problems[3]

Problem 1: You have a large inventory of identical items. You want to predict how many of these items will sell over the next 3 months

Problem 2: Youd like software to examine individual customer accounts, and for each account decide if it has been hacked/compromised

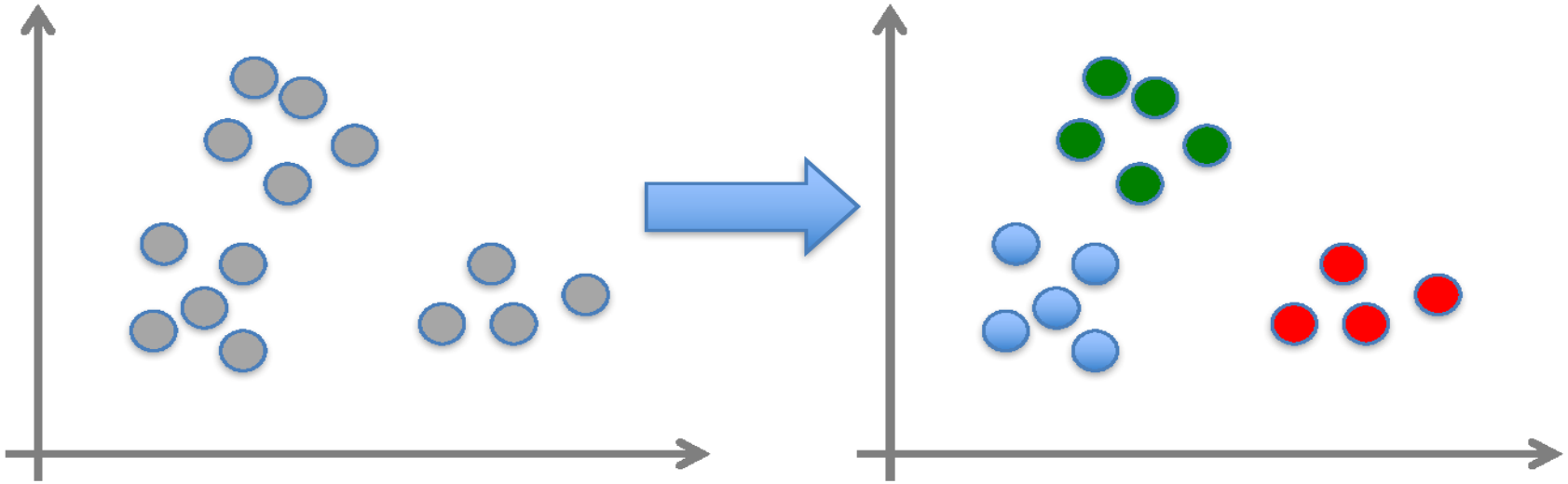
Q: Should we treat these as classification or as regression problems?

1. Both as classification problems
2. Problem 1 as classification problem, problem 2 as regression problem
3. Problem 1 as regression problem, problem 2 as classification problem
4. Both as regression problems

[3]Slide credit: Andrew Ng

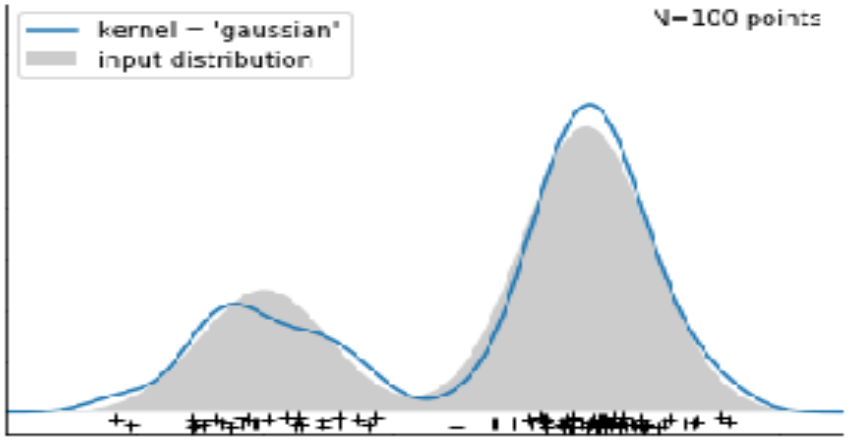
Unsupervised Learning

- Given $x_1, x_2, \ldots, x_n$ (without labels)
- Output hidden structure behind the x's
  - E.g., clustering



Unsupervised Learning : Clustering

- Given data $\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_N}$ (without Labels), and goal to estimate $P(\mathbf{x_i}|\theta)$ instead of $P(y_i|\mathbf{x_i}, \theta)$ in supervised Learning
- In clustering, estimate $P(K|D)$, where K denote is number of clusters

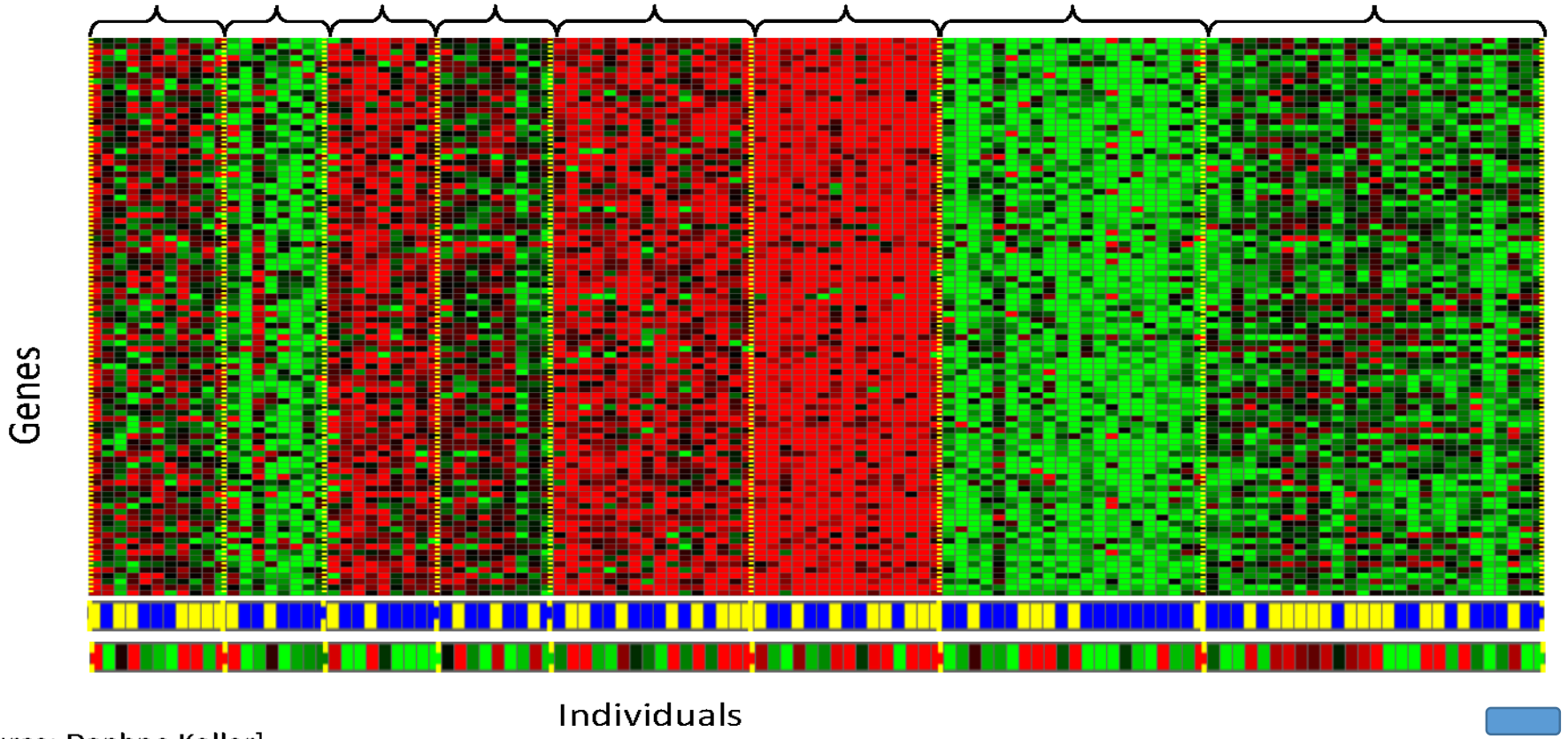


Density estimation using a Gaussian mixture.

Q: What is K here? How would you perform group assignments?

Unsupervised Learning

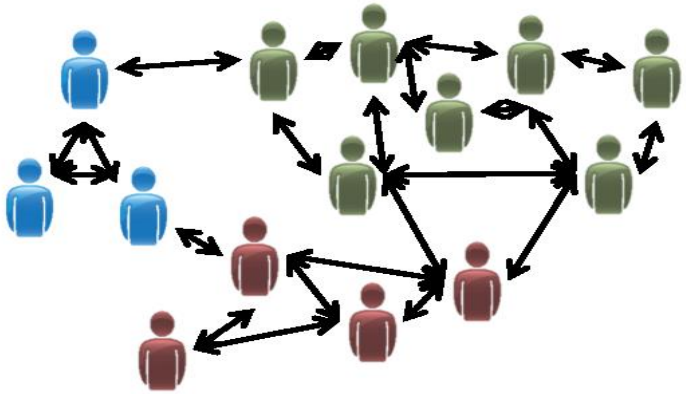
Genomics application: group individuals by genetic similarity



[Source: Daphne Koller]

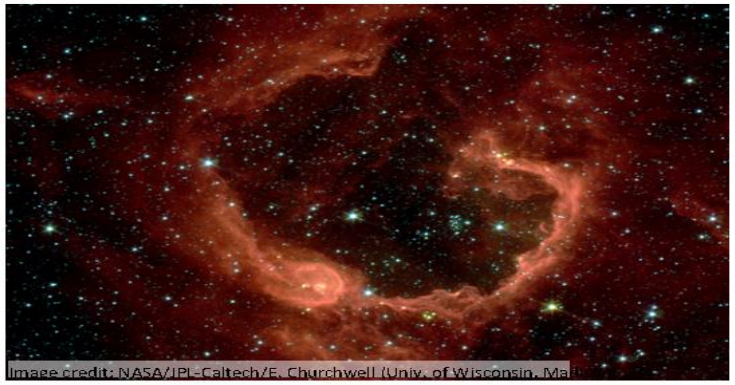
Unsupervised Learning

Organize computing clusters



Social network analysis

Market segmentation



Astronomical data analysis

Slide credit: Andrew Ng

Unsupervised Learning

- Independent component analysis – separate a combined signal into its original sources

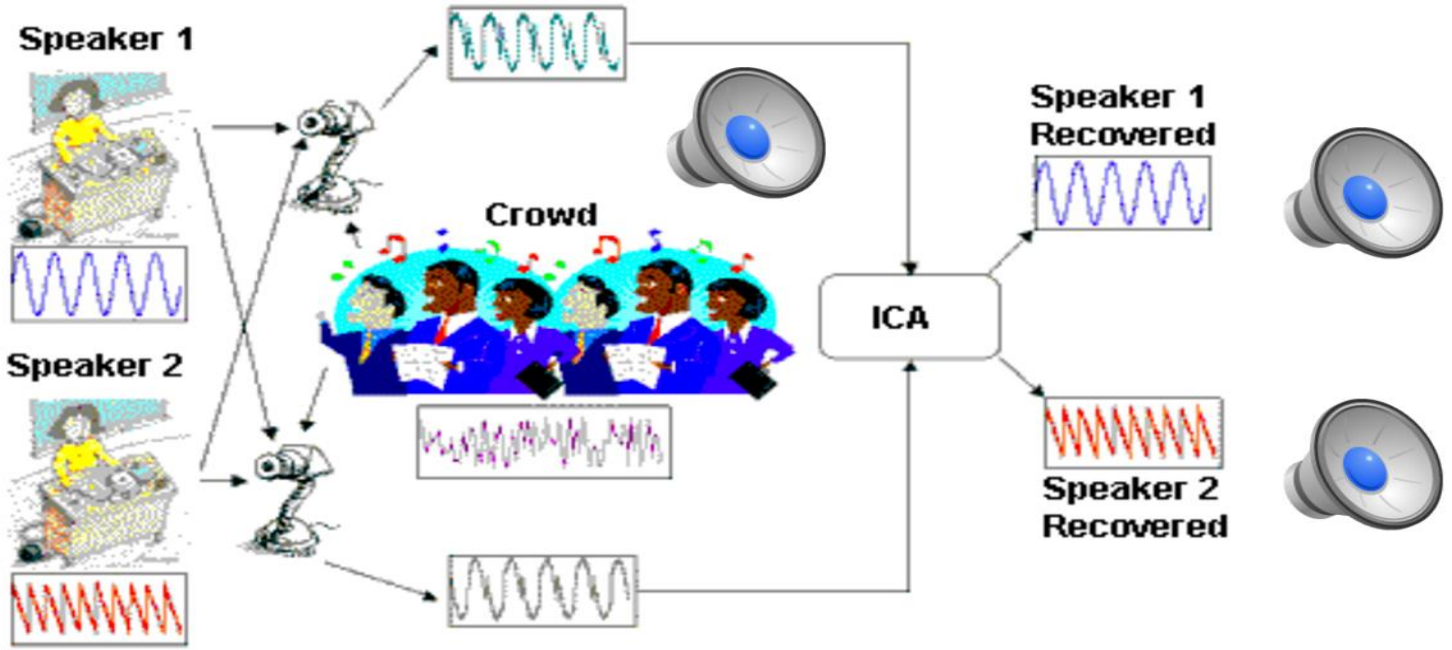


Image credit: statsoft.com Audio from http://www.ism.ac.jp/~shiro/research/blindsep.html

Unsupervised Learning

- Independent component analysis – separate a combined signal into its original sources

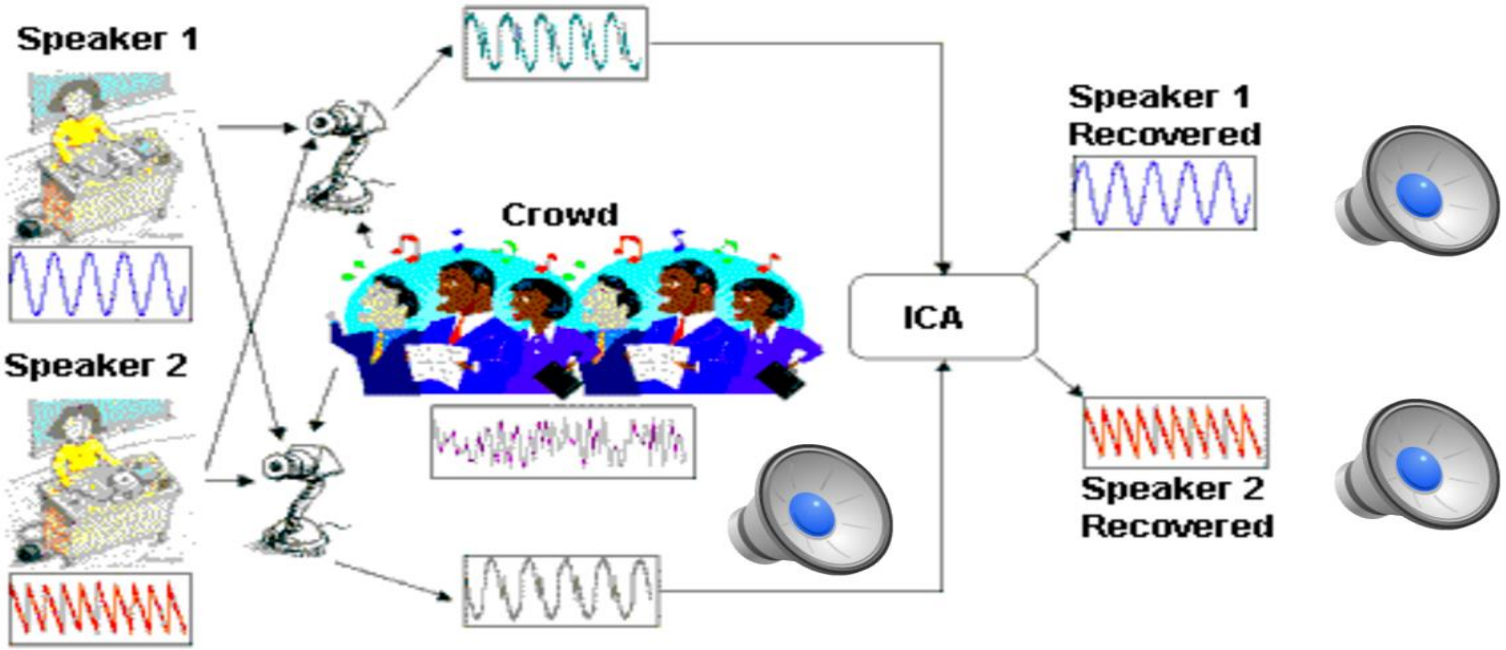
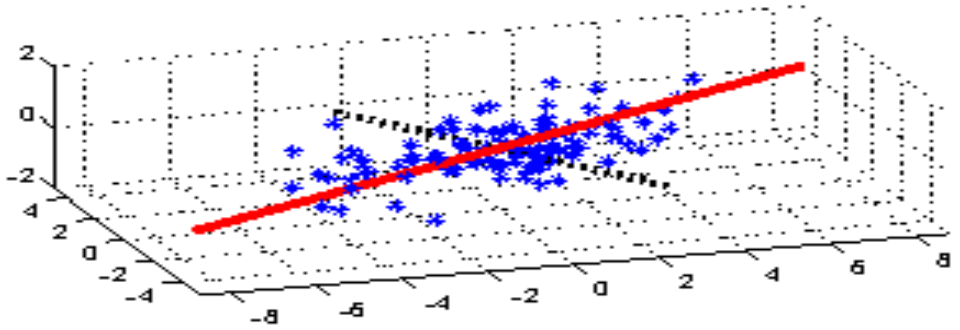


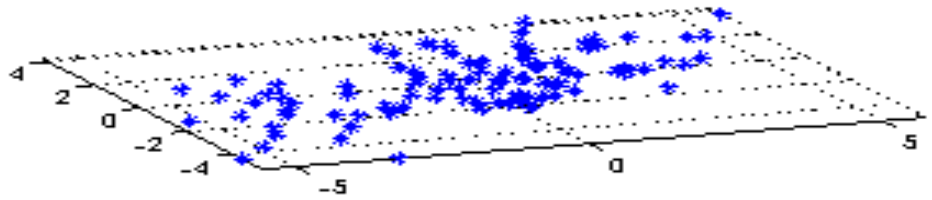
Image credit: statsoft.com Audio from http://www.ism.ac.jp/~shiro/research/blindsep.html

Unsupervised Learning : Dimensionality Reduction

- Data may appear high dimensional, there may only be a small number of degrees of variability, corresponding to **latent factors**
- When feature reduction is performed by selection a subset of the original features, this is called **feature selection**
- In **feature extraction**, dimensionality is reduced by projecting data to a lower dimensional subspace that captures the "essence" of the data
  - Principal components analysis (PCA) is a commonly used approach



A set of points that live on a 2d linear subspace embedded in 3d.



2D representation of the data.

Unsupervised Learning : Dimensionality Reduction Example

EignFaces: Modeling the appearance of face images

- Only few underlying latent factors can describe most of the variability
  - e.g., lighting, pose, identity, etc

25 randomly chosen pixel images from the face database.

The mean and the first three principal component basis vectors (eigenfaces).

Unsupervised Learning

Q: Of the following examples, which would you address using an unsupervised learning algorithm? (Choose all that apply.)[6]

1. Given email labeled as spam/not spam, learn a spam filter
2. Given a set of news articles found on the web, group them into set of articles about the same story
3. Given a database of customer data, automatically discover market segments and group customers into different market segments
4. Given a dataset of patients diagnosed as either having diabetes or not, learn to classify new patients as having diabetes or not

[6] Slide credit: Andrew Ng

Supervised Vs Unsupervised

Given: Training data: $(x_1, y_1), \ldots, (x_n, y_n)$ / $x_i \in \mathbb{R}^d$ and y_i is the label.

example $x_1 \rightarrow$	x_{11}	x_{12}	$\ldots$	x_{1d}	$y_1 \leftarrow$ label
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
example $x_i \rightarrow$	x_{i1}	x_{i2}	$\ldots$	x_{id}	$y_i \leftarrow$ label
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
example $x_n \rightarrow$	x_{n1}	x_{n2}	$\ldots$	x_{nd}	$y_n \leftarrow$ label

fruit	length	width	weight	label
fruit 1	165	38	172	Banana
fruit 2	218	39	230	Banana
fruit 3	76	80	145	Orange
fruit 4	145	35	150	Banana
fruit 5	90	88	160	Orange
...				
fruit n	...	...	...	...

Supervised Vs Unsupervised

fruit	length	width	weight	label
fruit 1	165	38	172	Banana
fruit 2	218	39	230	Banana
fruit 3	76	80	145	Orange
fruit 4	145	35	150	Banana
fruit 5	90	88	160	Orange
…				
fruit n	…	…	…	…

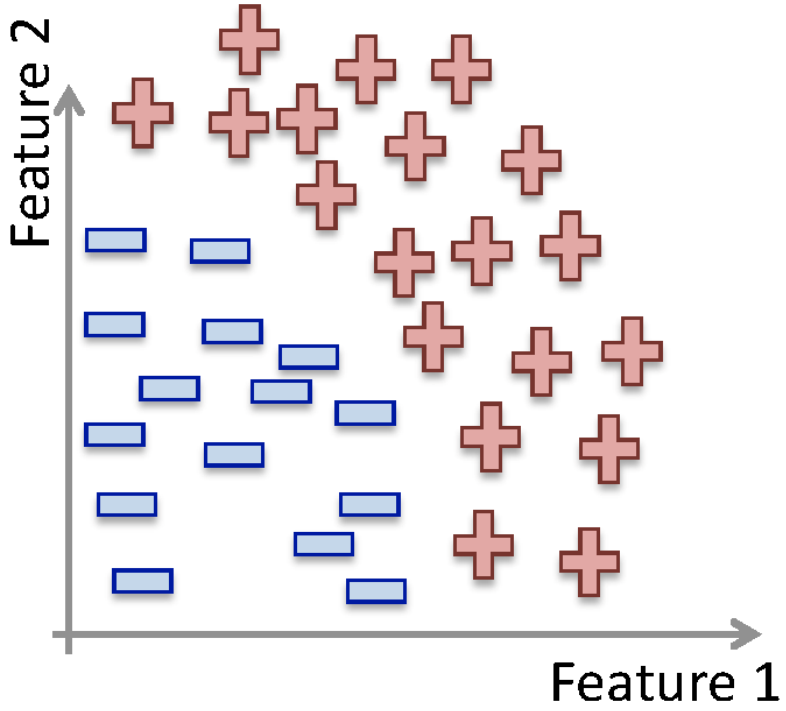
Unsupervised learning:

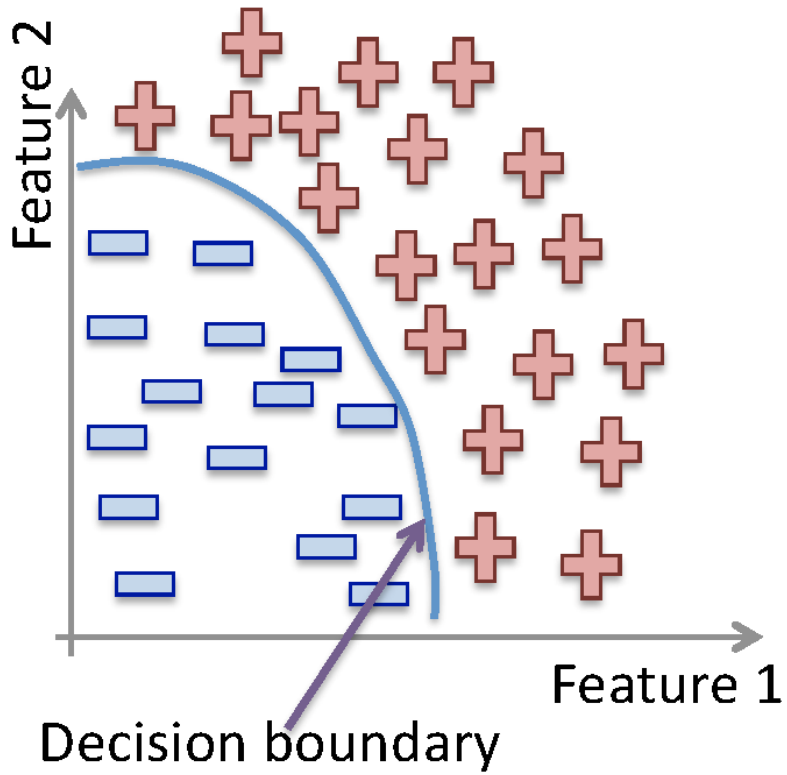
Learning a model from **unlabeled** data.

Supervised learning:

Learning a model from **labeled** data.

Supervised Learning

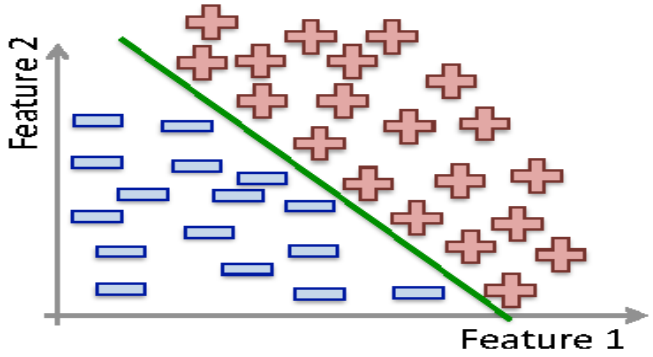


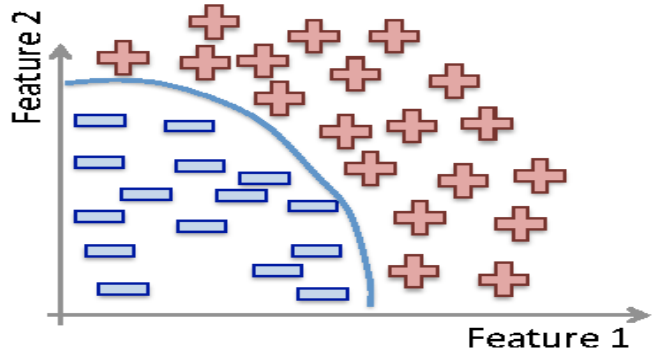


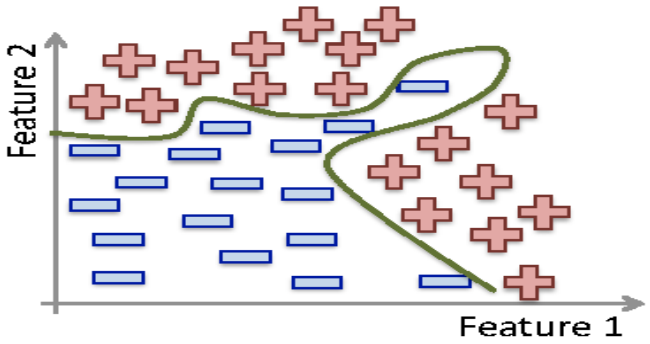
Methods: Support Vector Machines, neural networks, decision trees, K-nearest neighbors, naive Bayes, etc.

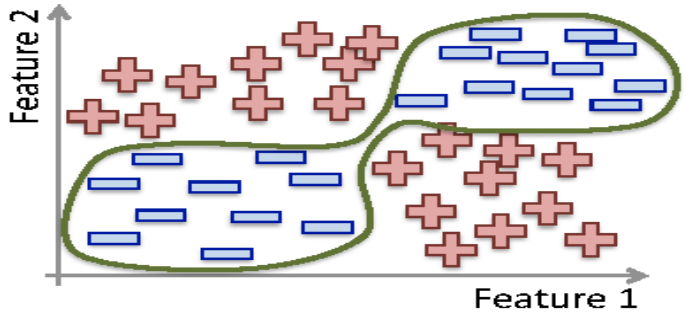
Supervised Learning

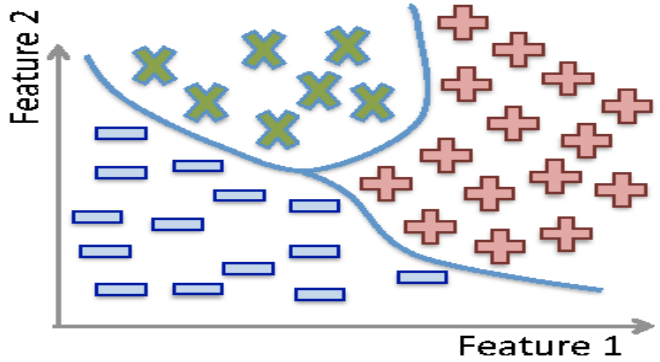
Classification:



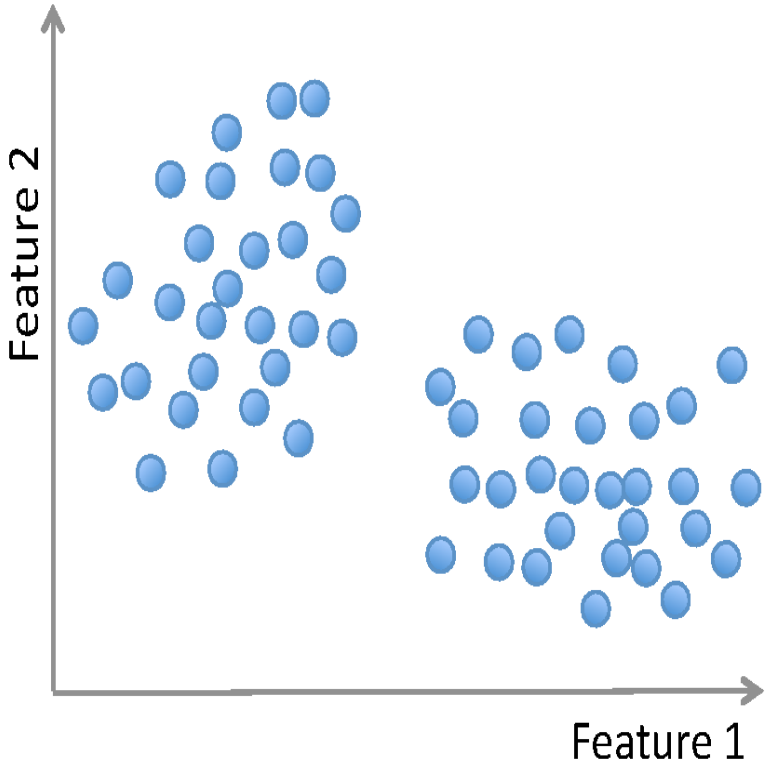


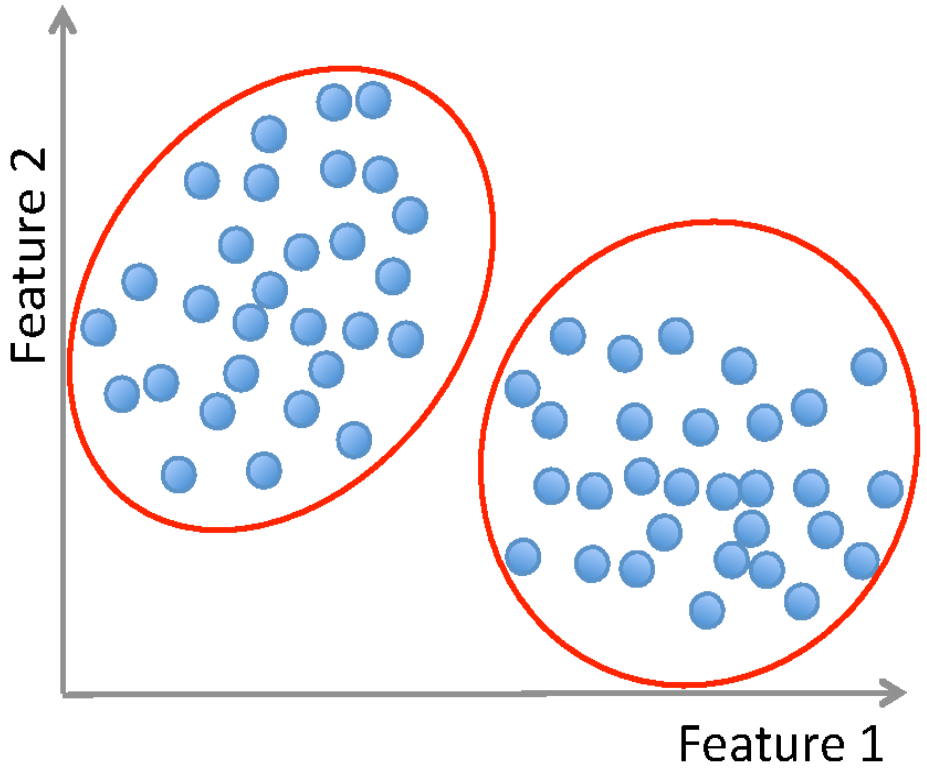






Unsupervised Learning



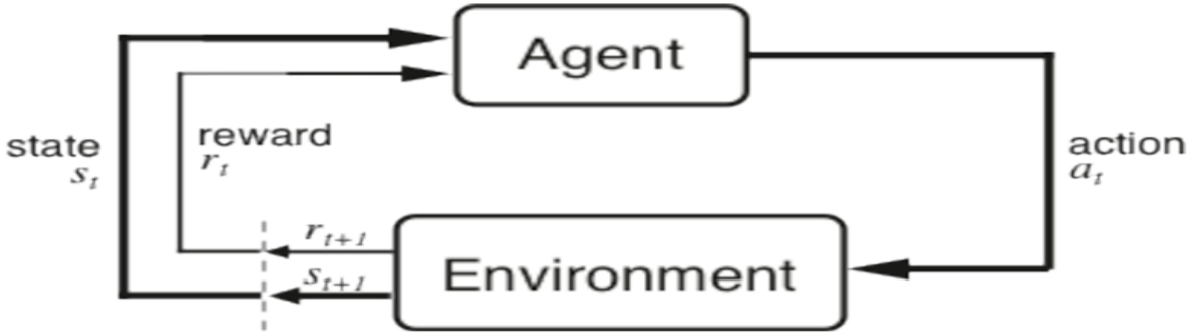


Methods: K-means, gaussian mixtures, hierarchical clustering, spectral clustering, etc.

Reinforcement Learning

- Given a sequence of states and actions with (delayed) rewards, output a policy
  - Policy is a mapping from states → actions that tells you what to do in a given state
- Examples:
  - Credit assignment problem
  - Game playing
  - Robot in a maze
  - Balance a pole on your hand

The Agent-Environment Interface



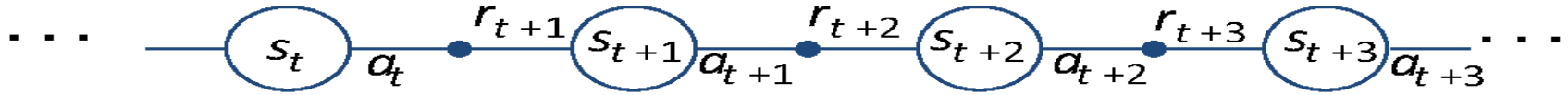
Agent and environment interact at discrete time steps : $t = 0, 1, 2, \mathrm{K}$

Agent observes state at step t: $s_t \in S$

produces action at step t: $a_t \in A(s_t)$

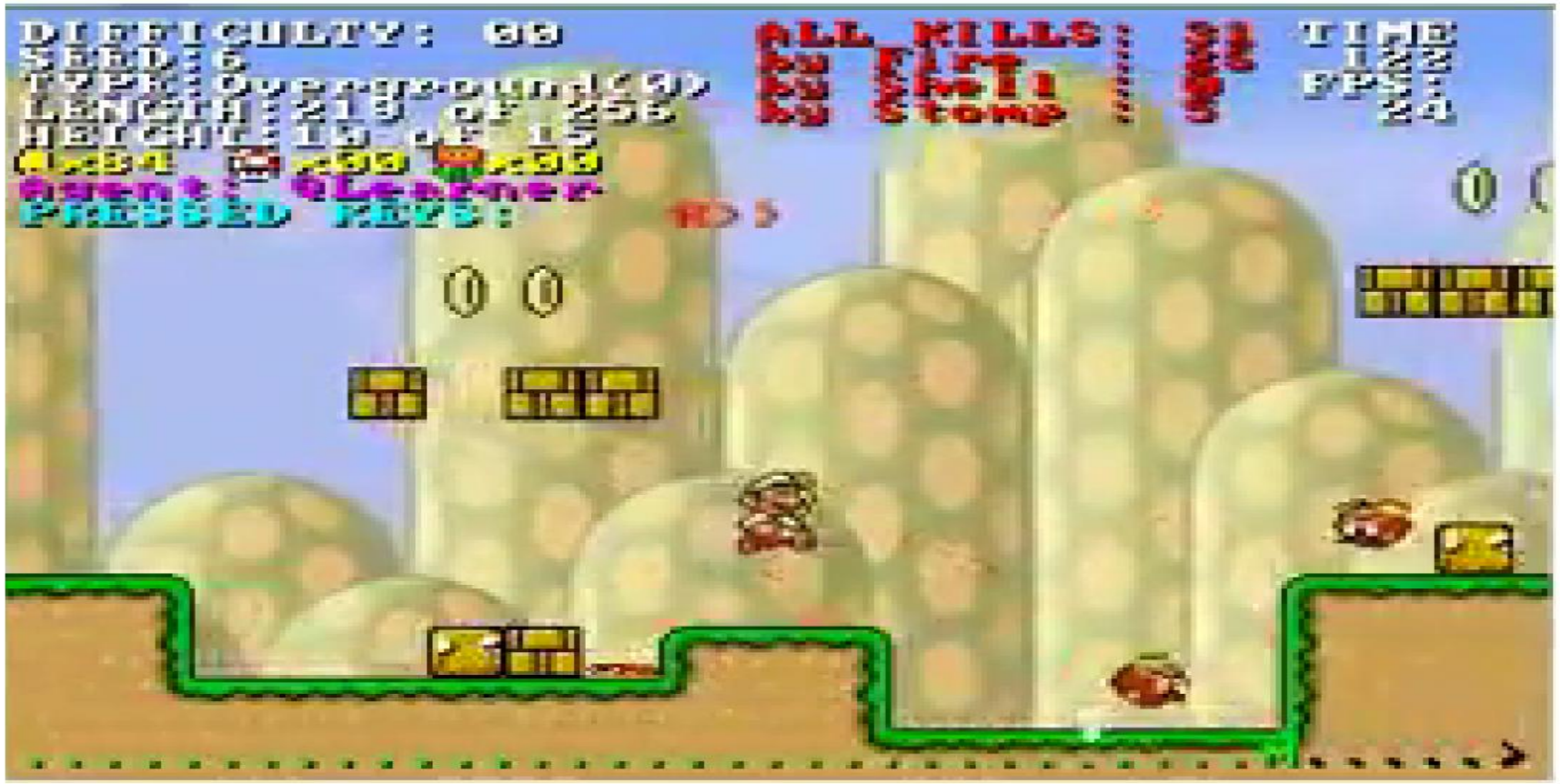
gets resulting reward : $r_{t+1} \in \Re$

and resulting next state : s_{t+1}



Slide credit: Sutton & Barto

Reinforcement Learning



https://www.youtube.com/watch?v=4cgWya-wjgY

Inverse Reinforcement Learning

- Learn policy from user demonstrations

Stanford Autonomous Helicopter

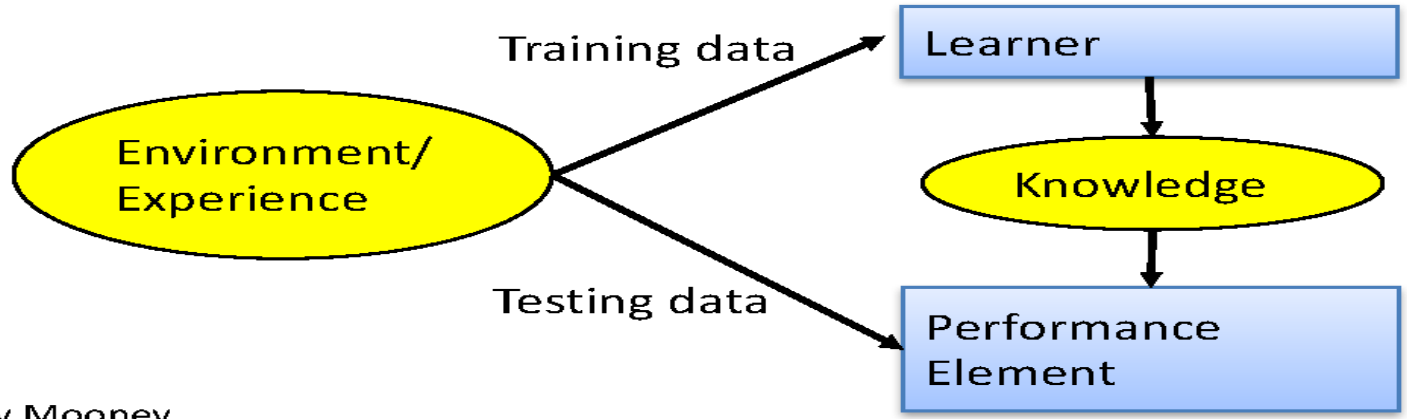
http://heli.stanford.edu/

https://www.youtube.com/watch?v=VCdxqn0fcnE

Framing a Learning Problem

Designing a Learning System

- Choose the training experience
- Choose exactly what is to be learned
  - i.e. the ***target function***
- Choose how to represent the target function
- Choose a learning algorithm to infer the target function from the experience



Based on slide by Ray Mooney

Training vs. Test Distribution

- We generally assume that the training and test examples are independently drawn from the same overall distribution of data
  - We call this "i.i.d" which stands for "independent and identically distributed"
- If examples are not independent, requires ***collective classification***
- If test distribution is different, requires ***transfer learning***

Slide credit: Ray Mooney

Train, Validation & Test

1. Training set is a set of examples used for learning a model (e.g., a classification model).
2. Validation set is a set of examples that cannot be used for learning the model but can help tune model parameters (e.g., selecting K in K-NN). Validation helps control overfitting.
3. Test set is used to assess the performance of the final model and provide an estimation of the test error.

Note: Never use the test set in any way to further tune the parameters or revise the model.

K-fold Cross Validation

A method for estimating test error using training data.

Algorithm:

Given a learning algorithm $\mathcal{A}$ and a dataset $\mathcal{D}$

Step 1: Randomly partition $\mathcal{D}$ into k equal-size subsets $\mathcal{D}_1, \ldots, \mathcal{D}_k$

Step 2:

For $j = 1$ to k

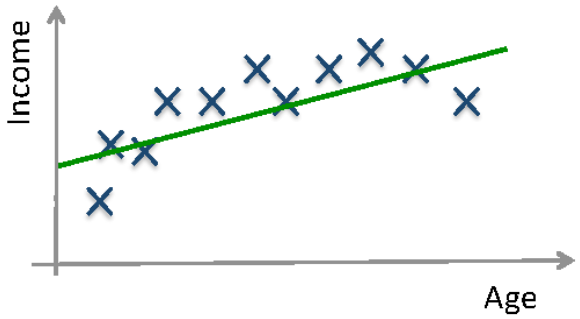
Train $\mathcal{A}$ on all $\mathcal{D}_i$, $i \in 1, \ldots k$ and $i \neq j$, and get f_j.

Apply f_j to $\mathcal{D}_j$ and compute $E^{\mathcal{D}_j}$

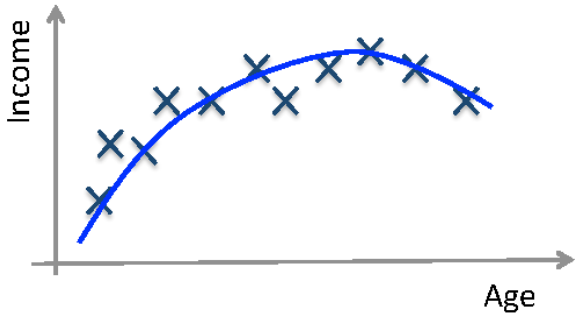
Step 3: Average error over all folds.

$$\sum_{j=1}^{k} (E^{\mathcal{D}_j})$$

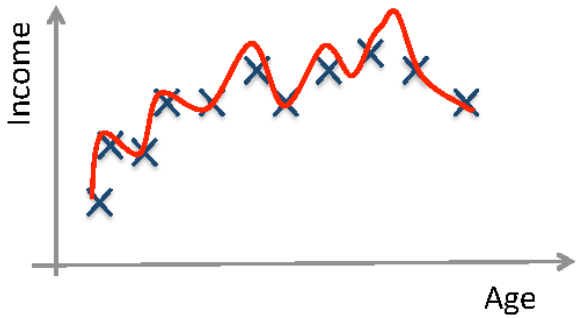
Training & Testing



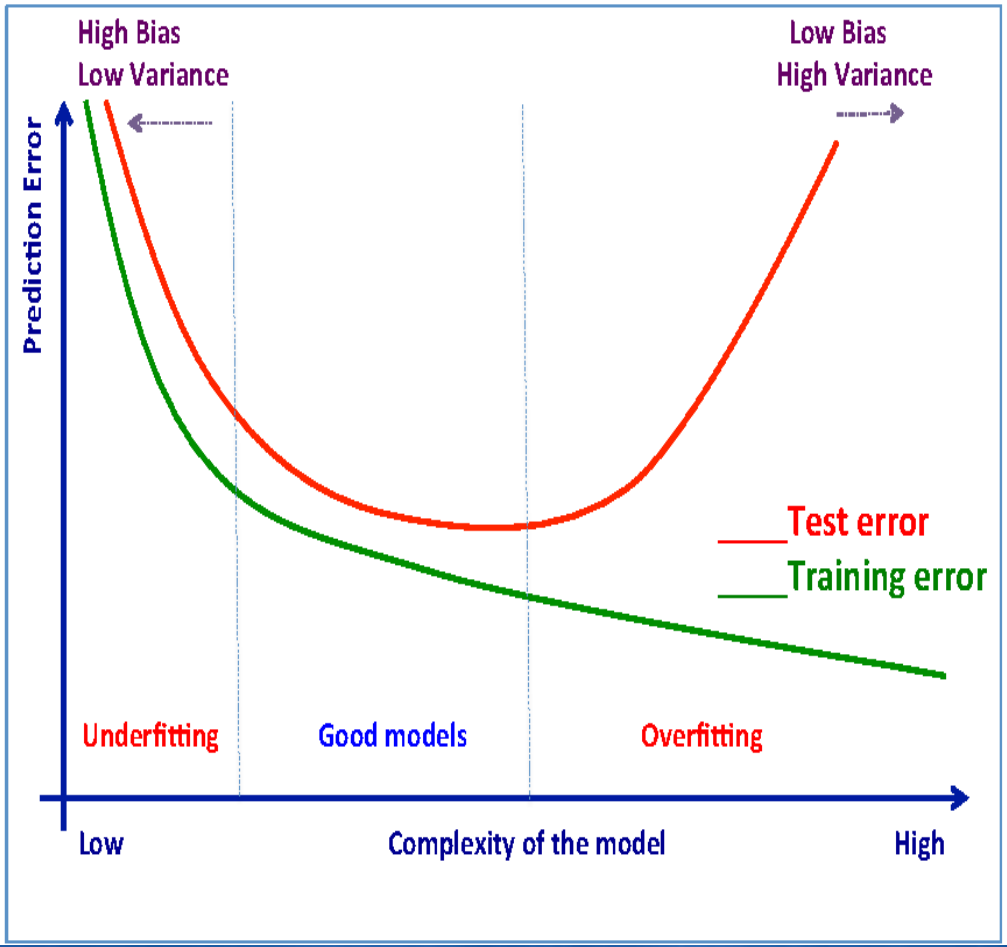
High bias (underfitting)



Just right!



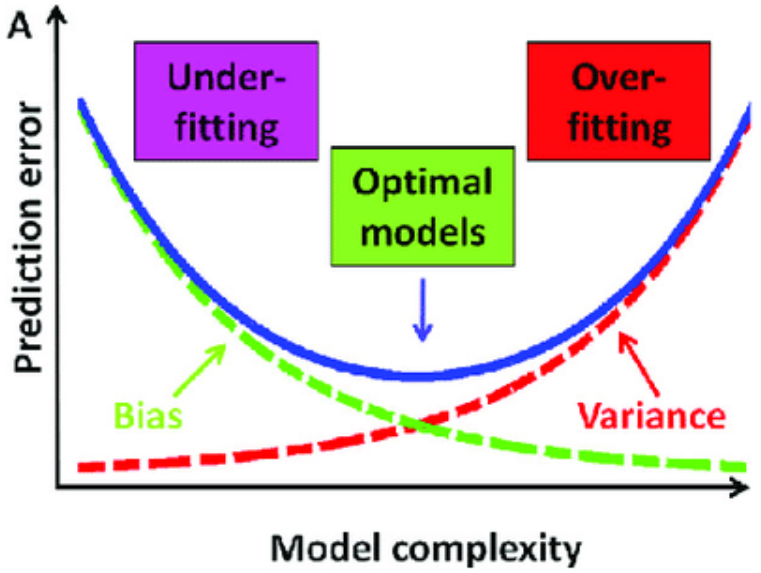
High variance (overfitting)



Overfitting Vs Underfitting

- Overfitting occurs when a statistical model or machine learning algorithm captures the noise of the data.
- Intuitively, overfitting occurs when the model or the algorithm fits the data too well.
- Specifically, if the model or algorithm shows low bias but high variance.
- Overfitting is often a result of an excessively complicated model, and it can be prevented by fitting multiple models and using validation or cross-validation to compare their predictive accuracies on test data.

- Underfitting occurs when a statistical model or machine learning algorithm cannot capture the underlying trend of the data.
- Intuitively, underfitting occurs when the model or the algorithm does not fit the data well enough.
- Specifically, underfitting occurs if the model or algorithm shows low variance but high bias. Underfitting is often a result of an excessively simple model.



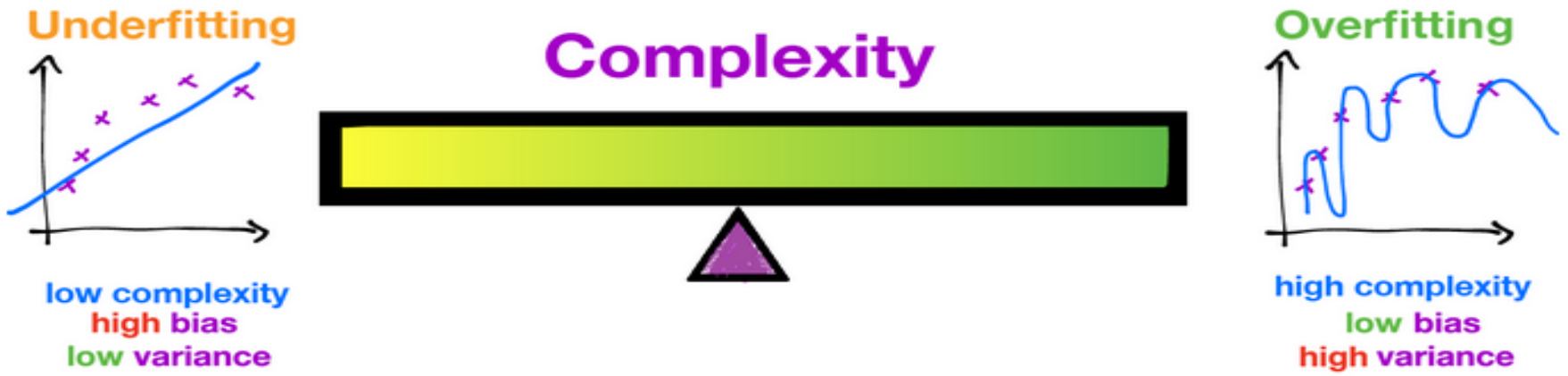
Overfitting Vs Underfitting

OVERFITTING

1. A mode is built using so many predictors that it captures noise along with the underlying pattern then it tries to fit the model too closely to the training data leaving very less scope for generalizability.
2. Low bias error, High variance error
3. A case of complex representation of a simpler reality
4. Example- Decision tress are prome to Overfitting

UNDERFITTING

1. A model is unable to capture the essence of the training data properly because of low number of parameters High bias error, Low variance error
2. It happen when we have very less amount of data to build an accurate model.
3. Arise if we try to build linear model with non-linear data.
4. Example- Linear regression and logistic regression models might face this



Avoid Overfitting

In general, use simple models!

- **Reduce the number** of features manually or do feature selection.
- Do a **model selection** (ML course).
- Use **regularization** (keep the features but reduce their importance by setting small parameter values) (ML course).
- Do a **cross-validation** to estimate the test error.

ML in a Nutshell

- Tens of thousands of machine learning algorithms
  - Hundreds new every year

- Every ML algorithm has three components:
  - **Representation**
  - **Optimization**
  - **Evaluation**

Slide credit: Pedro Domingos

Various Function Representations

- Numerical functions
  - Linear regression
  - Neural networks
  - Support vector machines
- Symbolic functions
  - Decision trees
  - Rules in propositional logic
  - Rules in first-order predicate logic
- Instance-based functions
  - Nearest-neighbor
  - Case-based
- Probabilistic Graphical Models
  - Naïve Bayes
  - Bayesian networks
  - Hidden-Markov Models (HMMs)
  - Probabilistic Context Free Grammars (PCFGs)
  - Markov networks

Slide credit: Ray Mooney

Various Search/Optimization Algorithms

- Gradient descent
  - Perceptron
  - Backpropagation
- Dynamic Programming
  - HMM Learning
  - PCFG Learning
- Divide and Conquer
  - Decision tree induction
  - Rule learning
- Evolutionary Computation
  - Genetic Algorithms (GAs)
  - Genetic Programming (GP)
  - Neuro-evolution

Slide credit: Ray Mooney

Evaluation

- Accuracy
- Precision and recall
- Squared error
- Likelihood
- Posterior probability
- Cost / Utility
- Margin
- Entropy
- K-L divergence
- etc.

Slide credit: Pedro Domingos

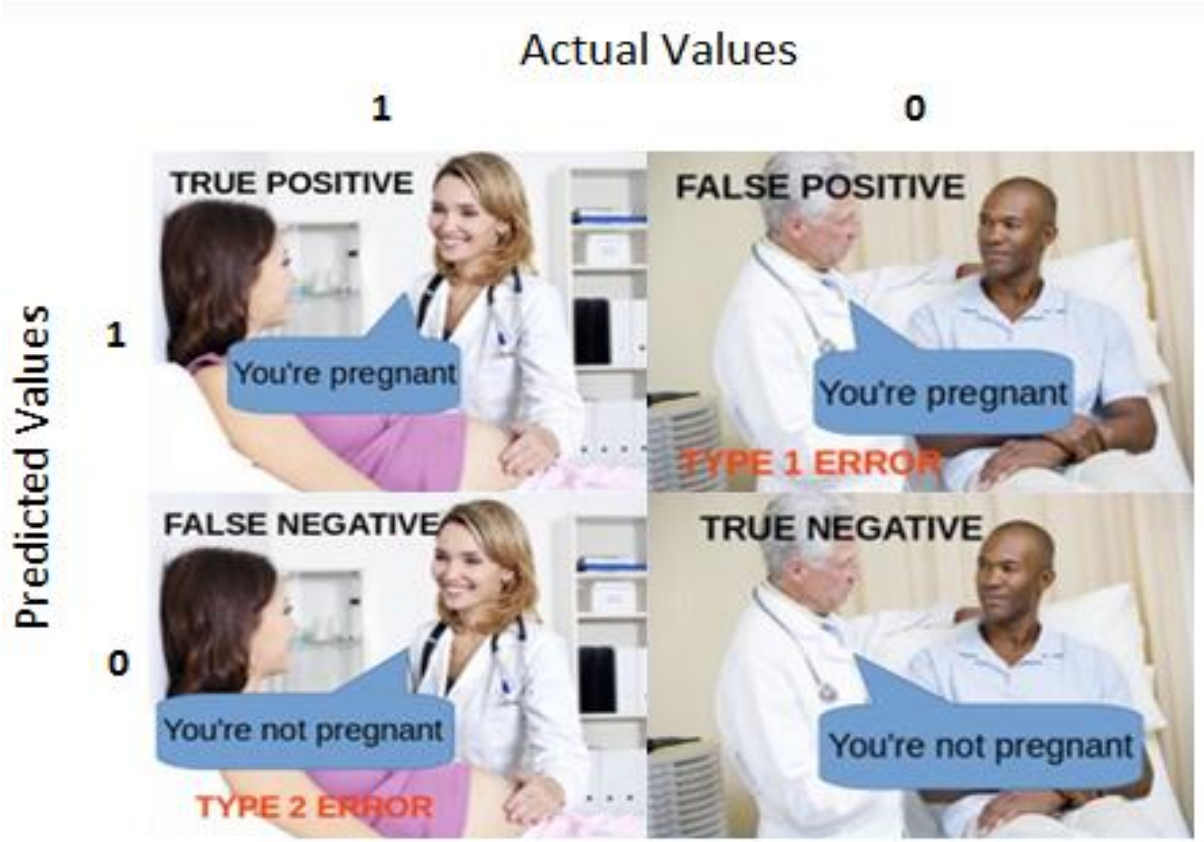
Confusion Matrix

A confusion matrix is a table that is often used to **describe the performance of a classification model** (or "classifier") on a set of test data for which the true values are known.

		Actual Label	
		Positive	Negative
Predicted Label	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Confusion Matrix : Example

Let’s understand TP, FP, FN, TN in terms of pregnancy analogy.



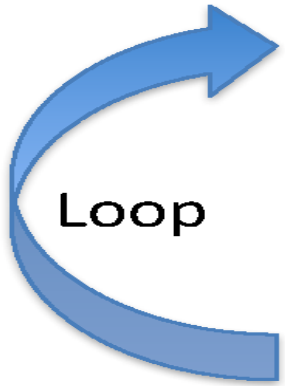
True Positive:
Interpretation: You predicted positive and it’s true.
You predicted that a woman is pregnant and she actually is.
True Negative:
Interpretation: You predicted negative and it’s true.
You predicted that a man is not pregnant and he actually is not.
False Positive: (Type 1 Error)
Interpretation: You predicted positive and it’s false.
You predicted that a man is pregnant but he actually is not.
False Negative: (Type 2 Error)
Interpretation: You predicted negative and it’s false.
You predicted that a woman is not pregnant but she actually is.

Evaluation Metrics

		Actual Label	
		Positive	Negative
Predicted Label	Positive	**True Positive (TP)**	**False Positive (FP)**
	Negative	**False Negative (FN)**	**True Negative (TN)**

Accuracy	(TP + TN) / (TP + TN + FP + FN)	The percentage of predictions that are correct
Precision	TP / (TP + FP)	The percentage of positive predictions that are correct
Sensitivity (Recall)	TP / (TP + FN)	The percentage of positive cases that were predicted as positive
Specificity	TN / (TN + FP)	The percentage of negative cases that were predicted as negative

ML in Practice



- Understand domain, prior knowledge, and goals
- Data integration, selection, cleaning, pre-processing
- Learn models
- Interpret results
- Consolidate and deploy discovered knowledge

Based on a slide by Pedro Domingos

Lessons Learned about Learning

- Learning can be viewed as using direct or indirect experience to approximate a chosen target function.
- Function approximation can be viewed as a search through a space of hypotheses (representations of functions) for one that best fits a set of training data.
- Different learning methods assume different hypothesis spaces (representation languages) and/or employ different search techniques.

Slide credit: Ray Mooney

A Brief History of Machine Learning

History of Machine Learning

- 1950s
  - Samuel's checker player
  - Selfridge's Pandemonium
- 1960s:
  - Neural networks: Perceptron
  - Pattern recognition
  - Learning in the limit theory
  - Minsky and Papert prove limitations of Perceptron
- 1970s:
  - Symbolic concept induction
  - Winston's arch learner
  - Expert systems and the knowledge acquisition bottleneck
  - Quinlan's ID3
  - Michalski's AQ and soybean diagnosis
  - Scientific discovery with BACON
  - Mathematical discovery with AM

Slide credit: Ray Mooney

History of Machine Learning (cont.)

- 1980s:
  - Advanced decision tree and rule learning
  - Explanation-based Learning (EBL)
  - Learning and planning and problem solving
  - Utility problem
  - Analogy
  - Cognitive architectures
  - Resurgence of neural networks (connectionism, backpropagation)
  - Valiant's PAC Learning Theory
  - Focus on experimental methodology
- 1990s
  - Data mining
  - Adaptive software agents and web applications
  - Text learning
  - Reinforcement learning (RL)
  - Inductive Logic Programming (ILP)
  - Ensembles: Bagging, Boosting, and Stacking
  - Bayes Net learning

Slide credit: Ray Mooney

History of Machine Learning (cont.)

- 2000s
  - Support vector machines & kernel methods
  - Graphical models
  - Statistical relational learning
  - Transfer learning
  - Sequence labeling
  - Collective classification and structured outputs
  - Computer Systems Applications (Compilers, Debugging, Graphics, Security)
  - E-mail management
  - Personalized assistants that learn
  - Learning in robotics and vision
- 2010s
  - Deep learning systems
  - Learning for big data
  - Bayesian methods
  - Multi-task & lifelong learning
  - Applications to vision, speech, social networks, learning to read, etc.
  - ???

Based on slide by Ray Mooney

What We'll Cover in this Course

- **Supervised learning**
  - Decision tree induction
  - Linear regression
  - Logistic regression
  - Support vector machines & kernel methods
  - Model ensembles
  - Bayesian learning
  - Neural networks & deep learning
  - Learning theory
- **Unsupervised learning**
  - Clustering
  - Dimensionality reduction
- **Reinforcement learning**
  - Temporal difference learning
  - Q learning
- **Evaluation**
- **Applications**

Our focus will be on applying machine learning to real applications


Thank You!