Introduction to ML

May 22, 2023

What is Machine Learning?

- Making predictions or decisions from data
- “Programming computers to optimize a performance criterion using example data or past experience” (Ethem Alpaydin, Machine Learning, 2010)
- “A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.” (Tom Mitchell, Machine Learning, 1997)
- “Learning general models from a data of particular examples”
- “Build a model that is *a good and useful approximation* to the data.“

Today

Traditional Programming

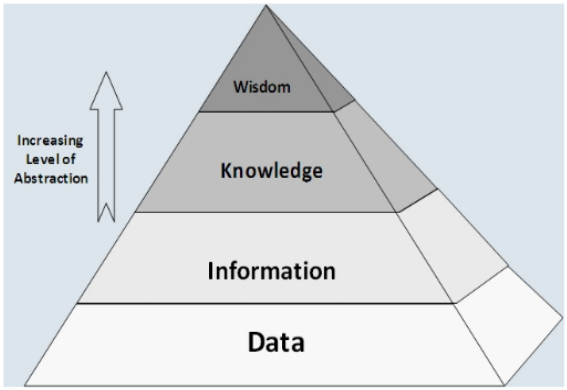
Machine Learning

Related Terms

Machine Learning, Data Mining, Knowledge Discovery, Artificial Intelligence, Statistical Learning, Pattern Recognition, Computational Learning

When is Machine Learning Used?

- Human expertise does not exist
  - E.g. navigating on Mars
- Humans are unable to explain their expertise
  - E.g. speech recognition
- Solution changes in time
  - E.g. routing on a computer network
- Solution needs to be adapted to particular cases
  - E.g. user biometrics
- Data is cheap and abundant; knowledge is expensive and scarce



Applications of Machine Learning

From: cheapsales@buystufffromme.com
To: ang@cs.stanford.edu
Subject: Buy now!

Deal of the week! Buy now!
Rolex w4tchs - $100
Med1cine (any kind) - $50
Also low cost M0rgages
available.

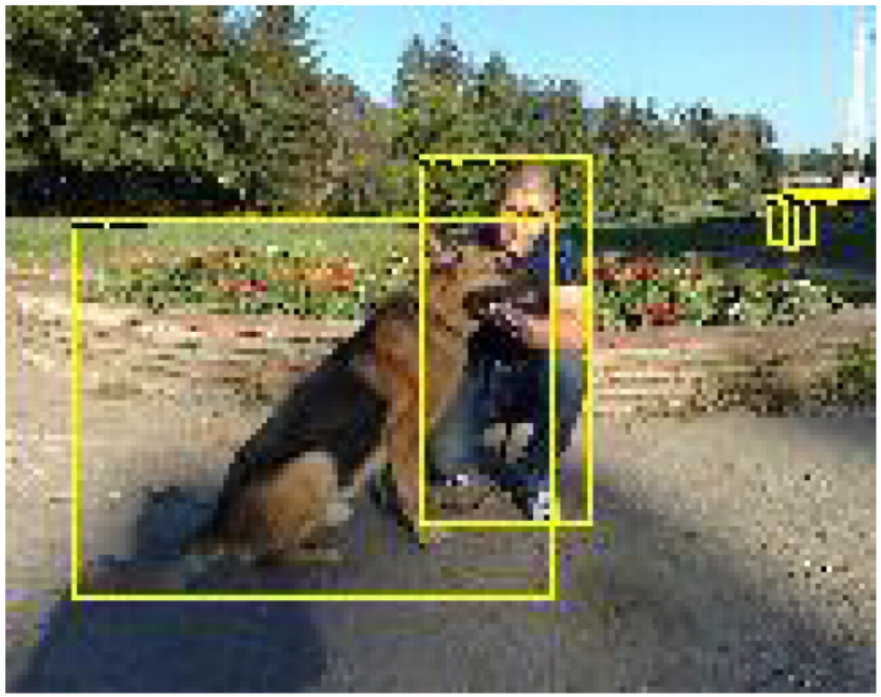
Spam

From: Alfred Ng
To: ang@cs.stanford.edu
Subject: Christmas dates?

Hey Andrew,
Was talking to Mom about plans
for Xmas. When do you get off
work. Meet Dec 22?
Alf

Non-spam

Applications of Machine Learning



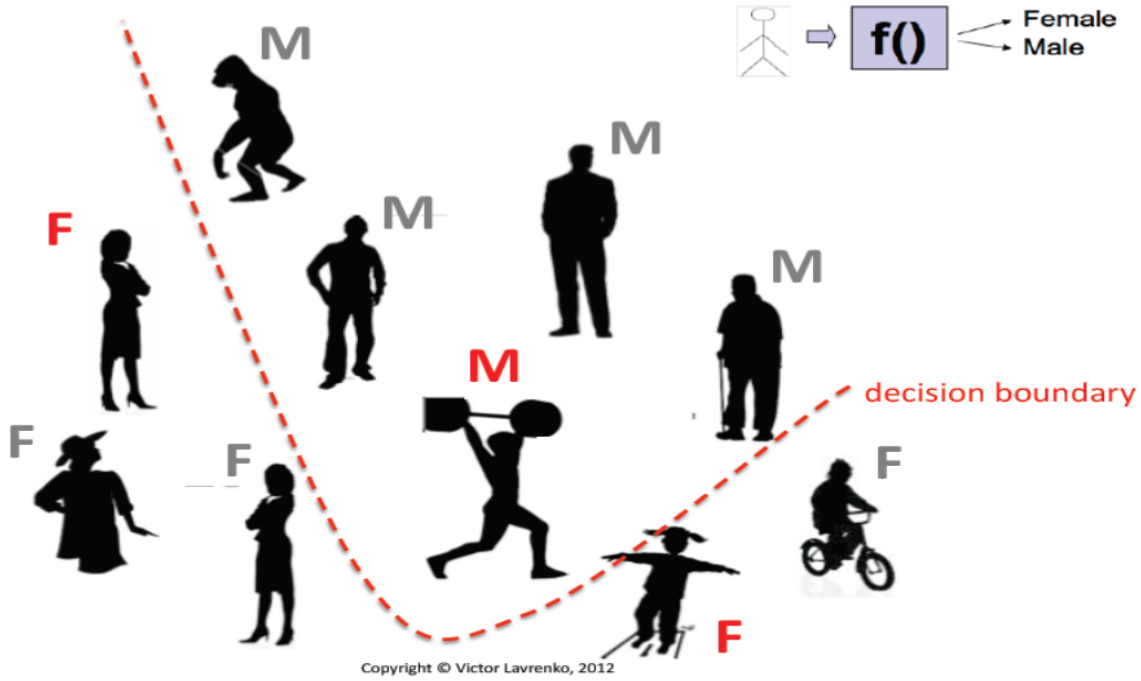
More ML Applications

- Science (Astronomy, neuroscience, medical imaging, bio-informatics)
- Environment (energy, climate, weather, resources)
- Retail (Intelligent stock control, demographic store placement)
- Manufacturing (Intelligent control, automated monitoring, detection methods)
- Security (Intelligent smoke alarms, fraud detection)
- Marketing (promotions, ...)
- Management (Scheduling, timetabling)
- Finance (credit scoring, risk analysis...)
- Web data (information retrieval, information extraction, ...)

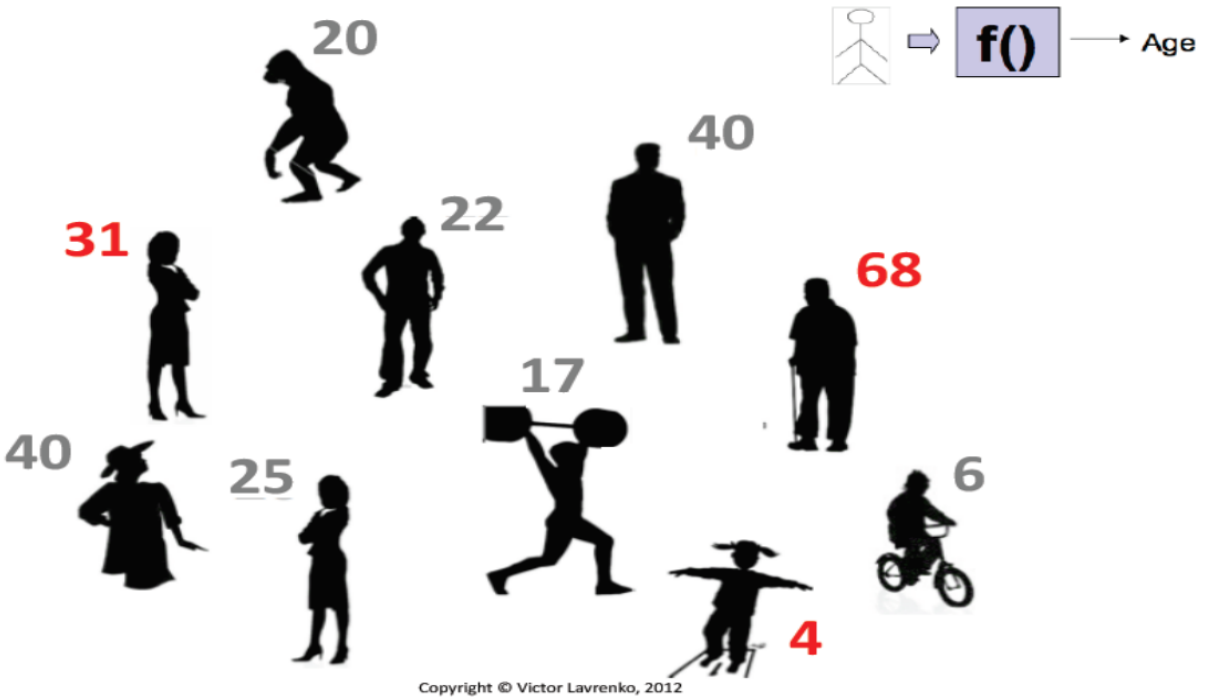
Overview of ML

- Supervised learning
  - Predict an output y when given an input **x**
  - For categorical y : classification.
  - For real-valued y : regression.
- Unsupervised learning
  - Create an internal representation of the input, e.g. clustering, dimensionality
  - This is important in machine learning as getting labels is often difficult and expensive
- Other settings of ML
  - Reinforcement learning (learning from “rewards”)
  - Semi-supervised learning (combines supervised + unsupervised)
  - Active learning, Transfer learning, Structured prediction

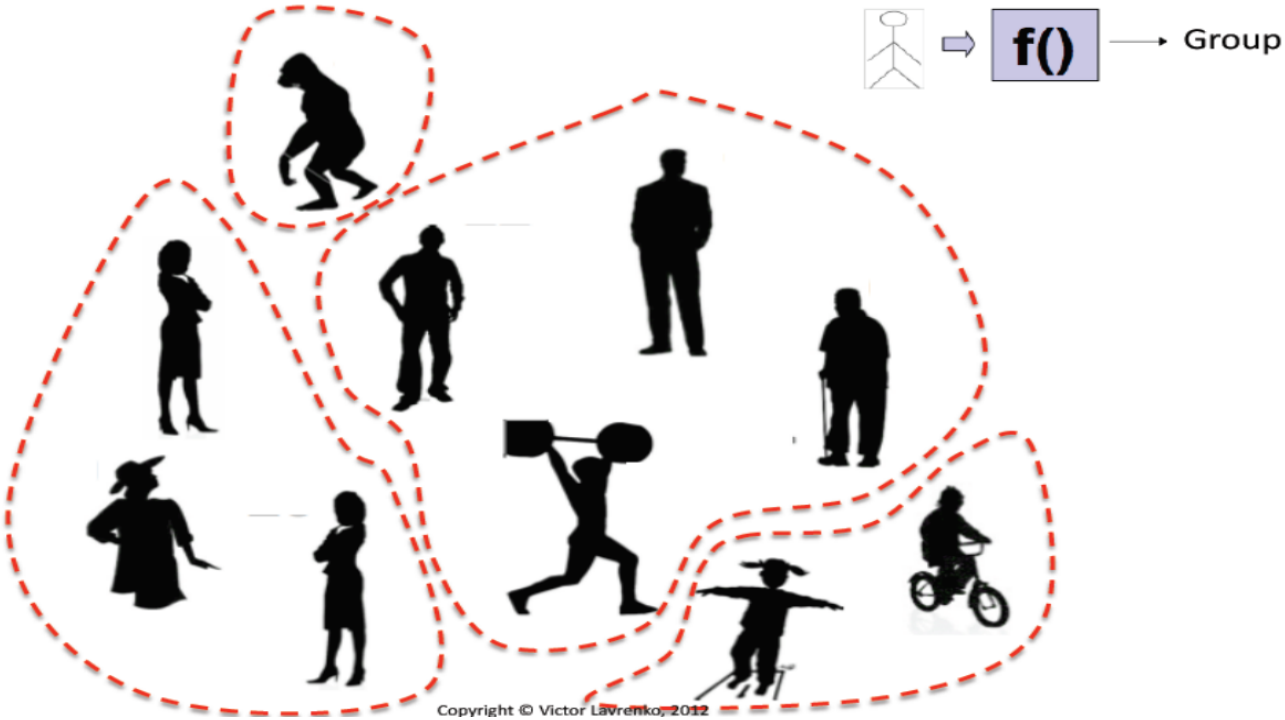
Classification (Supervised Learning)



Regression (Supervised Learning)

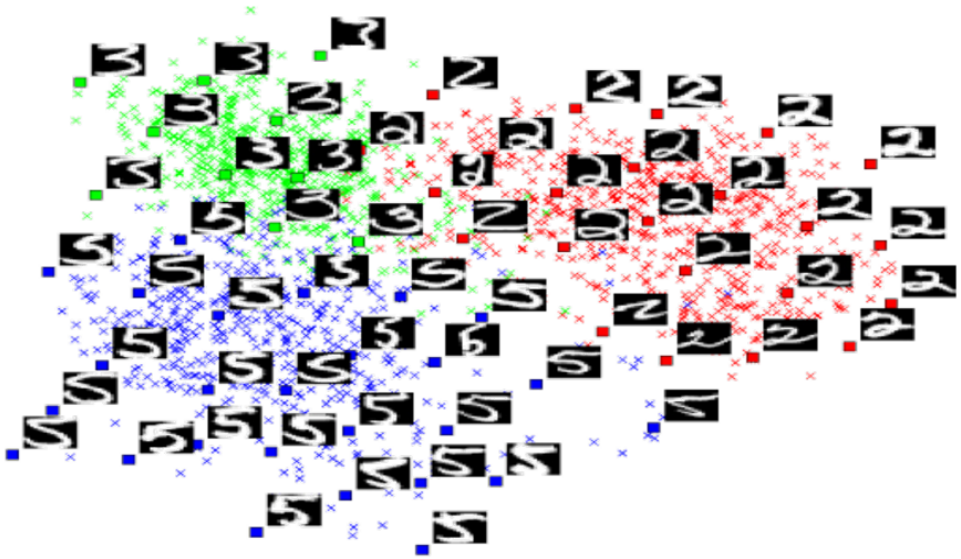


Clustering (Unsupervised Learning)



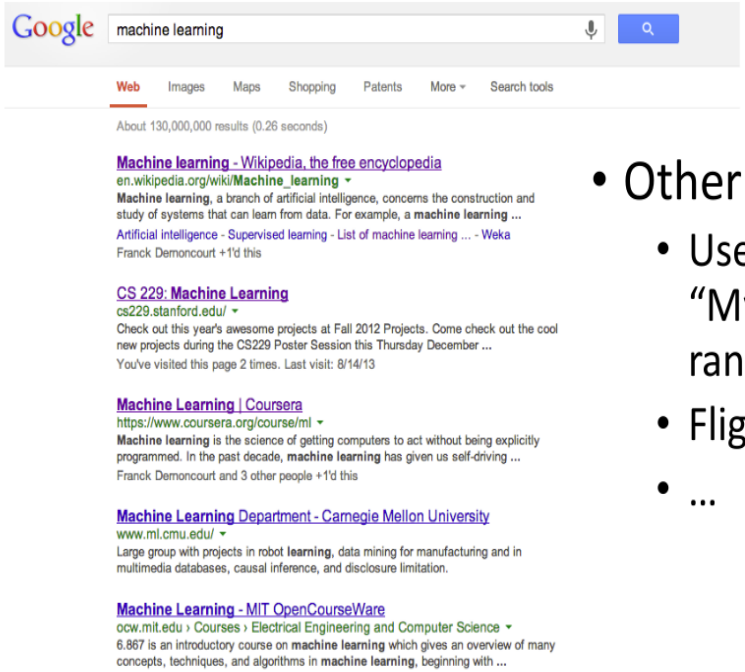
Dimensionality Reduction (Unsupervised Learning)

- Large sample size is required for high-dimensional data
- Query accuracy and efficiency degrade rapidly as the dimension increases
- Strategies
  - Feature reduction
  - Feature selection
  - Manifold learning
  - Kernel learning



Other Settings: Ranking (Supervised Learning)

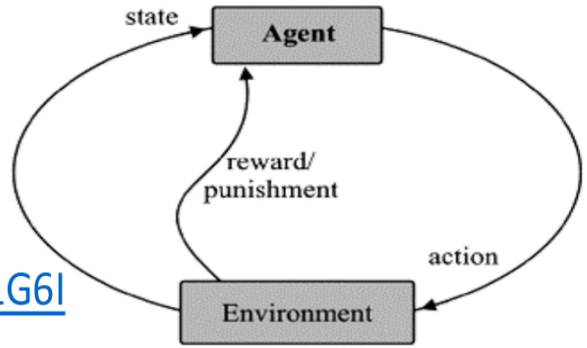
Given a query and a set of web pages, rank them according to relevance



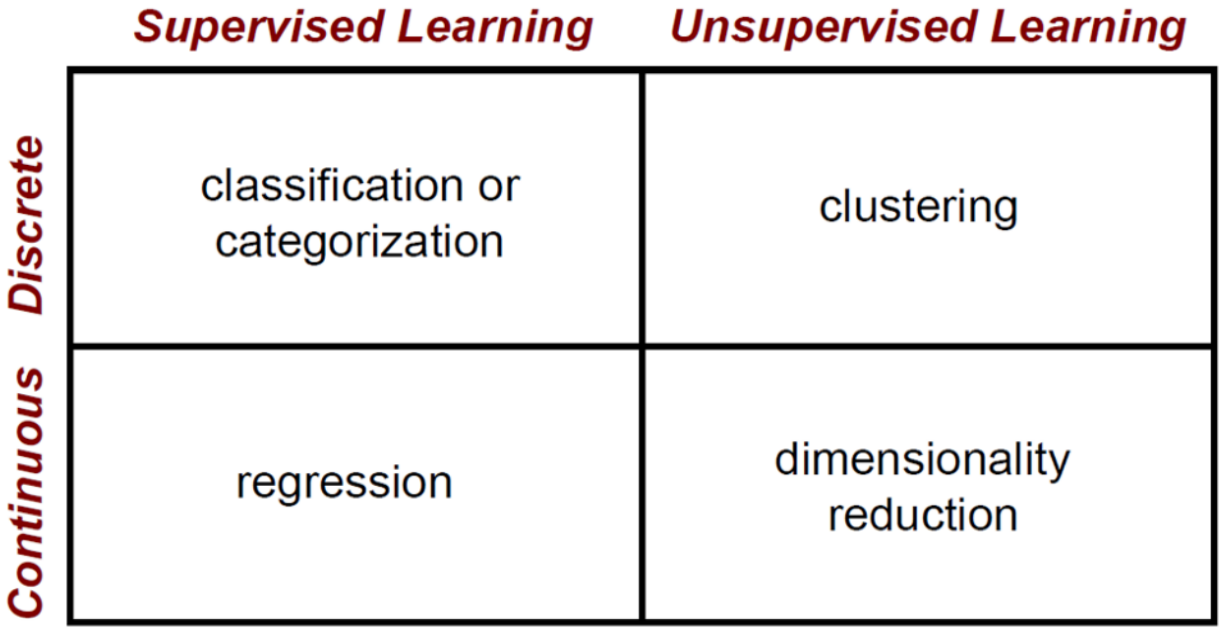
- Other applications
  - User preference, e.g. Netflix "My List" -- movie queue ranking
  - Flight search (search in general)
  - ...

Other Settings: Reinforcement Learning

- Learning a policy: A sequence of outputs
- No supervised output but delayed reward
  - E.g. Game playing
  - E.g. Robot in a maze
- Multiple agents, partial observability, ...
- Example (Simple Demo):
  - https://www.youtube.com/watch?v=DCjbk4m1G6I



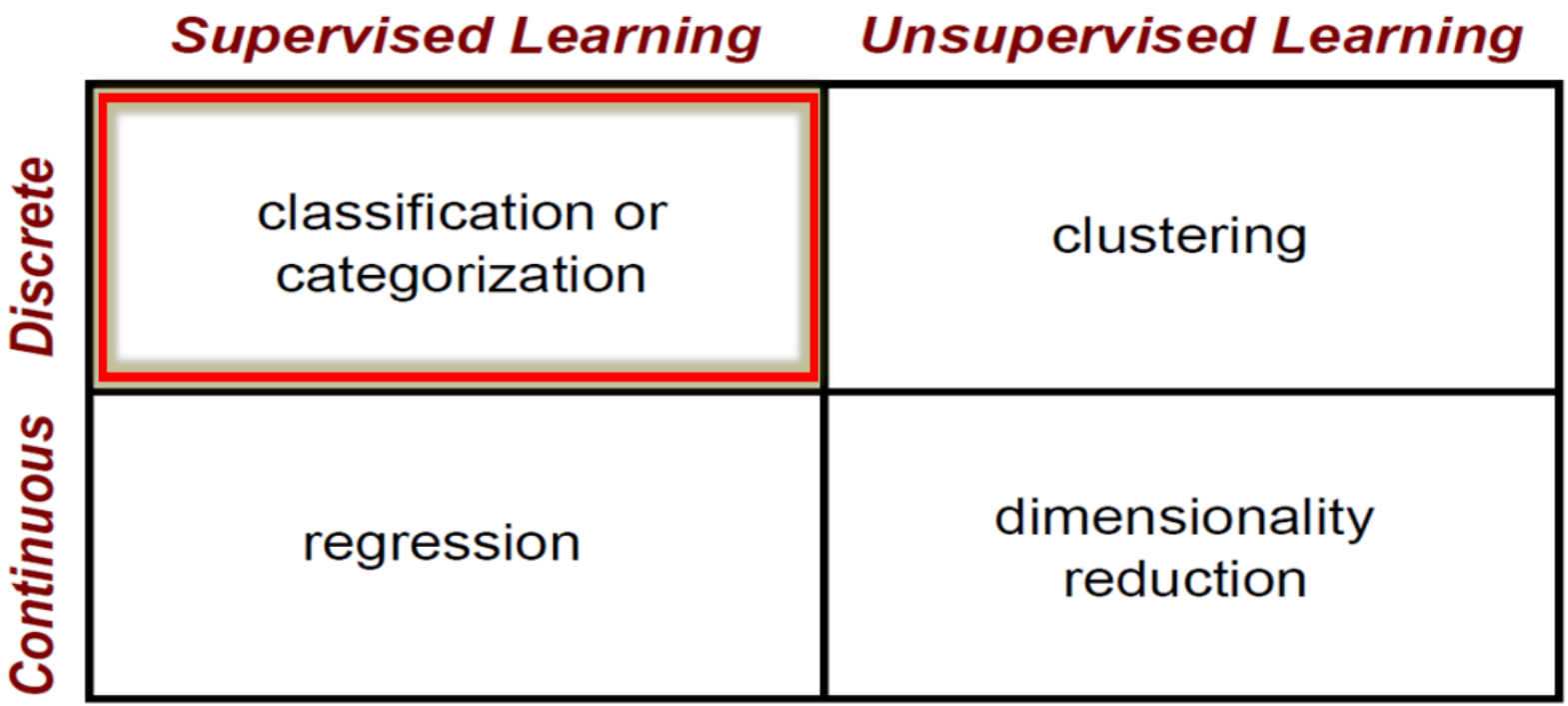
ML Problems


Supervised Learning
Unsupervised Learning
Discrete
Continuous
classification or categorization
clustering
regression
dimensionality reduction

Mathematical Basis

- Functions, Logarithms and Exponentials
- Vectors, Dot Products, Orthogonality
- Matrices, Matrix Operations, Linear Transformations, Eigendecomposition
- Calculus, Differentiation, Integration
- Probability and Statistics
- Functional Analysis, Hilbert Spaces

ML Problems: Recall



Classification Methods

- k-Nearest Neighbors
- Decision Trees
- Naïve Bayes
- Support Vector Machines
- Logistic Regression
- Neural Networks
- Ensemble Methods (Boosting, Random Forests)

How to evaluate?

Training vs Generalization Error

- Training Error
  - Not very useful
  - Relatively easy to obtain low error
- Generalization Error
  - How well we do on future data

$$E_{train} = \frac{1}{n} \sum_{i=1}^{n} error(f_D(\mathbf{x}_i), y_i)$$

same? different by how much?

training examples

value we predicted

true value

$$E_{gen} = \int error(f_D(\mathbf{x}), y) p(y, \mathbf{x}) d\mathbf{x}$$

over all possible x,y

error as before

how often we expect to see such x and y

How to compute generalization error?

Estimating Generalization Error

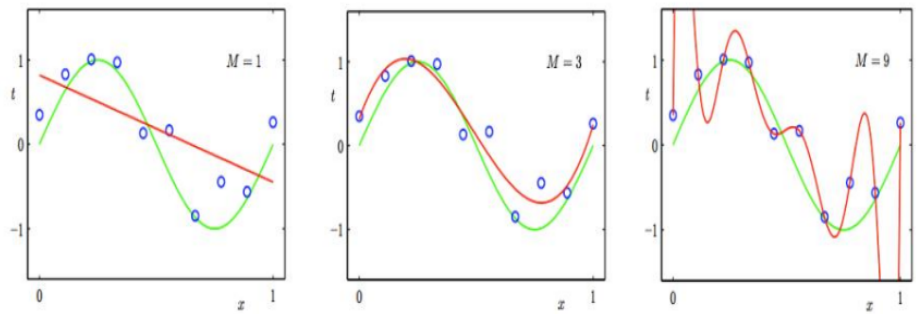
- Testing Error
  - Set aside part of training data (testing set)
  - Learn a predictor without using any of this test data
  - Predict values for testing set, compute error
  - This is an estimate of generalization error

$$E_{test} = \frac{1}{n} \sum_{i=1}^{n} error(f_D(\mathbf{x}_i), y_i)$$

over testing set

Underfitting and Overfitting

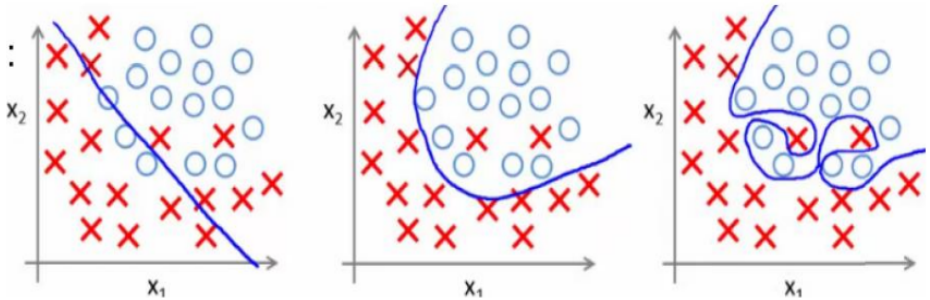
Regression



predictor too inflexible:
cannot capture pattern

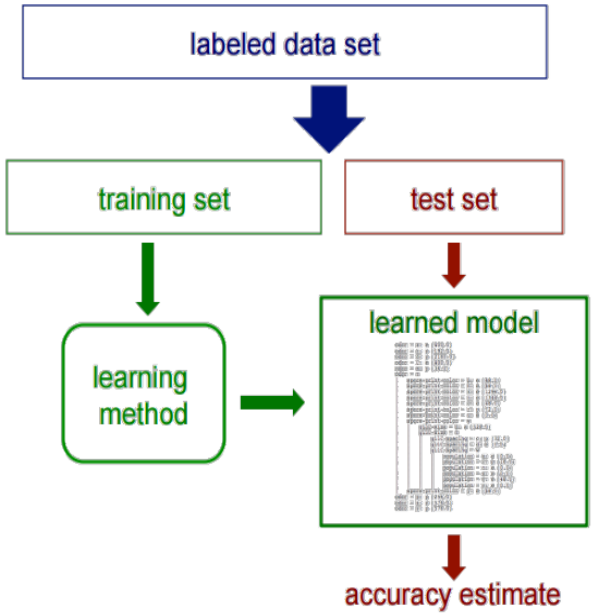
predictor too flexible:
fits noise in the data

Classification

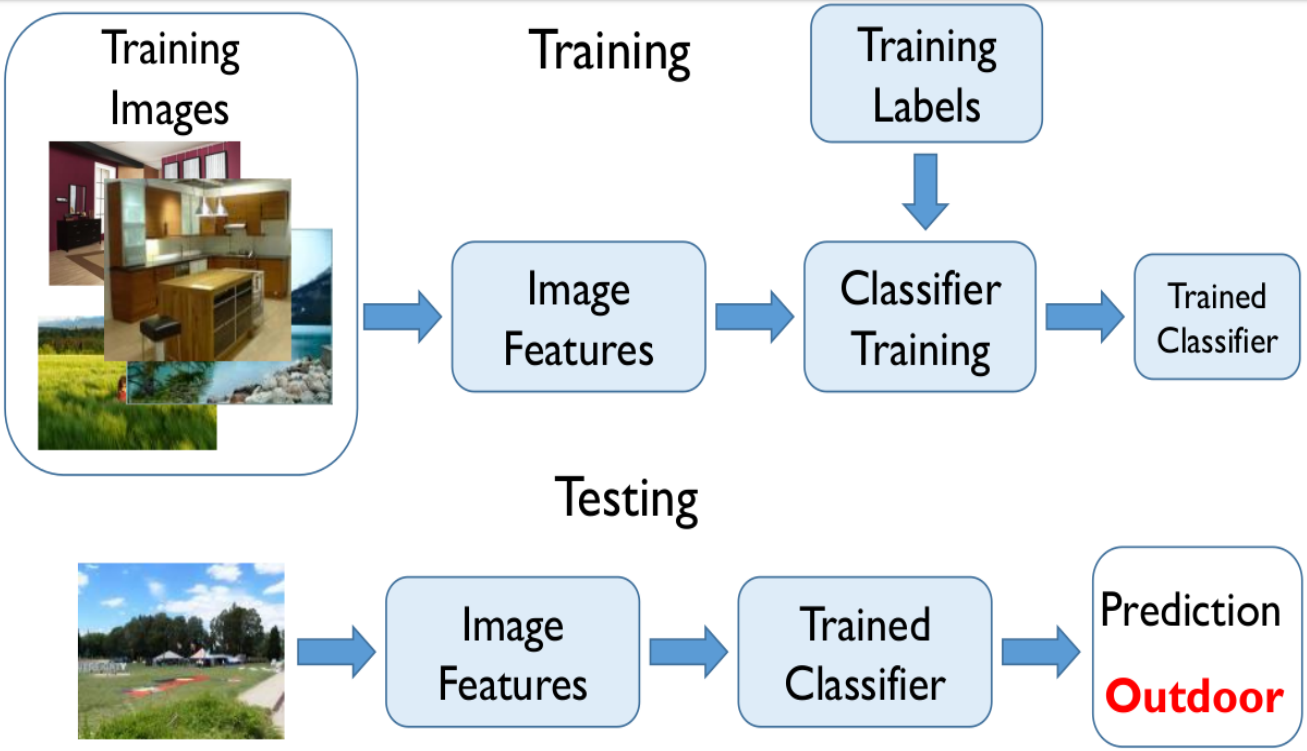


Estimating Generalization Error

- Getting an unbiased estimate of the accuracy of a learned model



Example: Image Classification



Training, Validation, Test Sets

Training set

- NB: Count frequencies, DT: Pick attributes to split on

Validation set

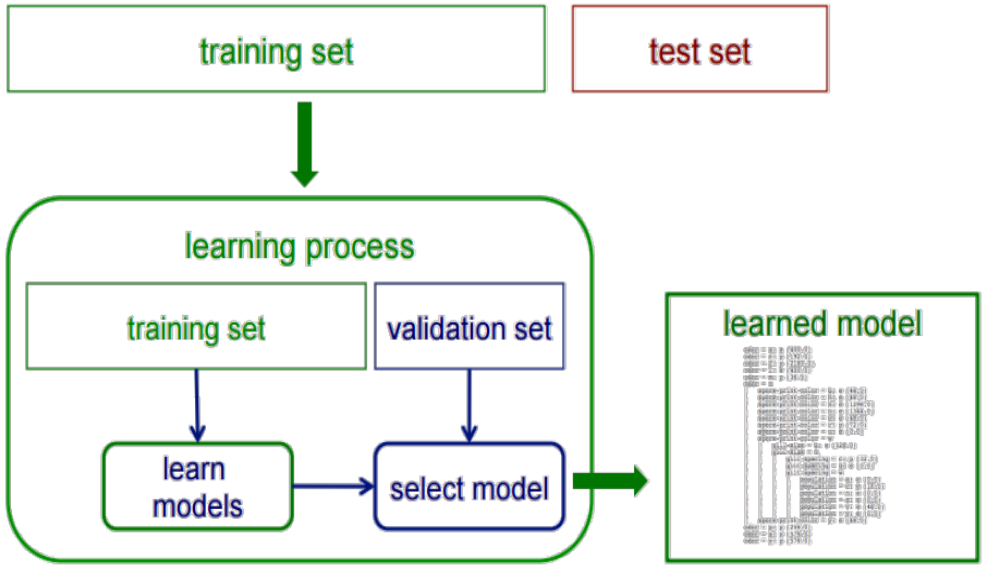
- Pick best-performing algorithm (NB vs DT vs..)
- Fine-tune parameters (Tree depth, k in kNN, c in SVM)

Testing set

- Run multiple trials and average

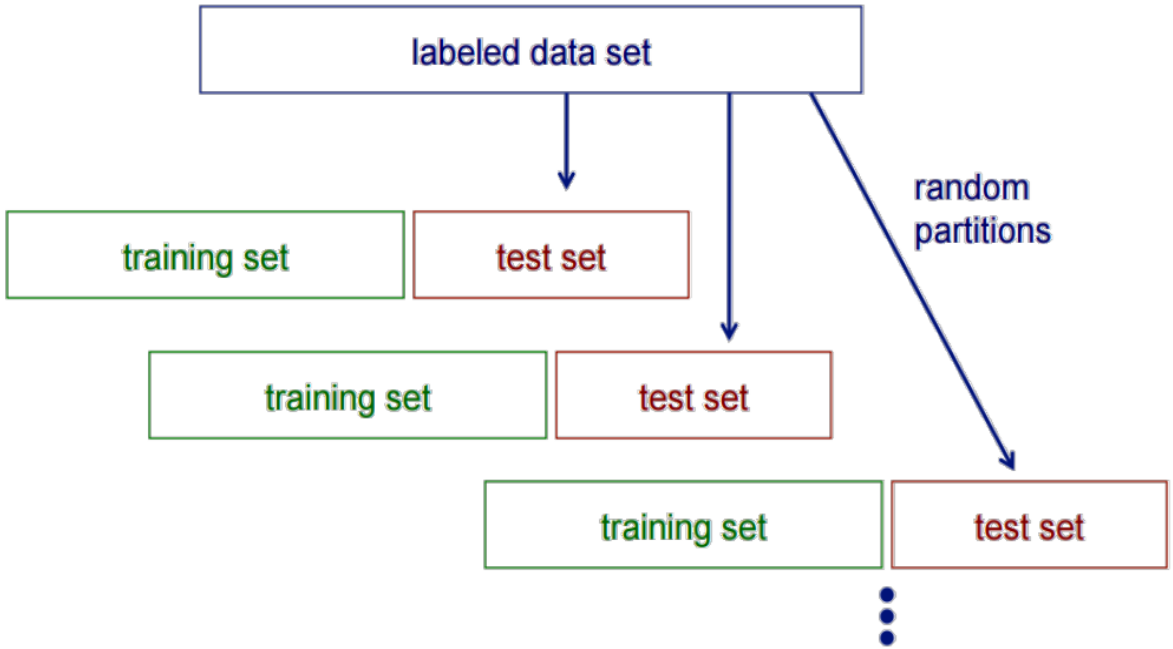
Use of Validation Sets

- If we want unbiased estimates of accuracy during the learning process:



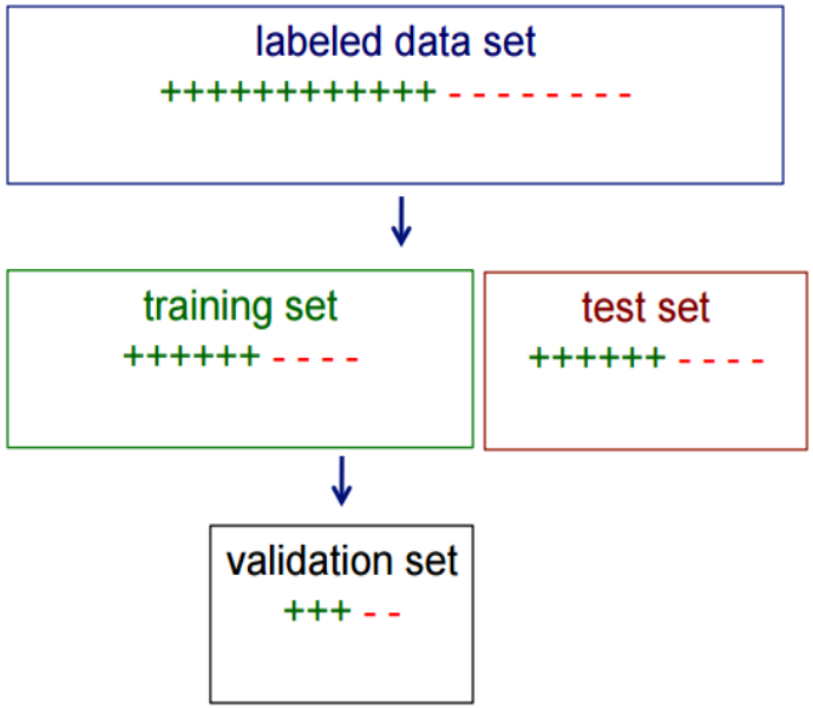
Random Resampling

- We can artificially increase training set size using random resampling:



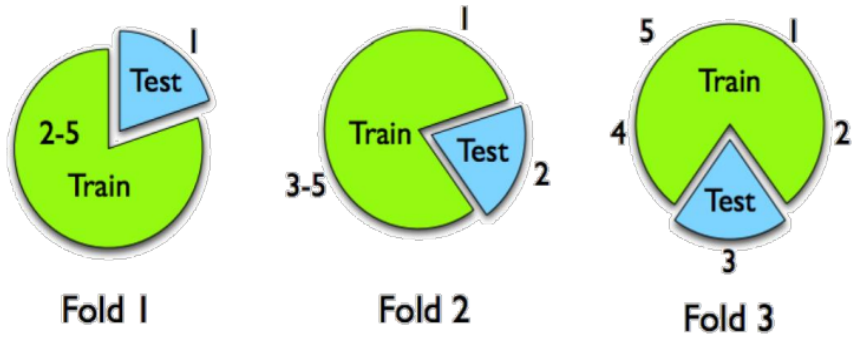
Stratified Sampling

- When randomly selecting training or validation sets, we may want to ensure that class proportions are maintained in each selected set
- This can be done via stratified sampling: first stratify instances by class, then randomly select instances from each class proportionally.



Model Selection

- Resubstitution
- K-fold cross-validation



- Leave-one-out
  - N-fold cross-validation

Cross-Validation: Example

- Suppose we have 100 instances, and we want to estimate accuracy with cross validation

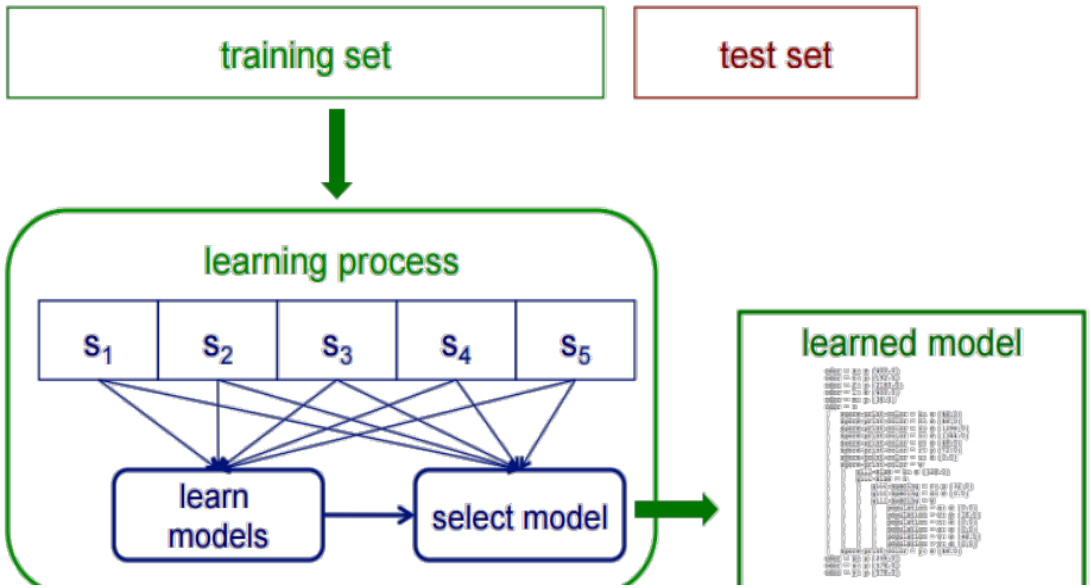
iteration	train on	test on	correct
1	s_2 s_3 s_4 s_5	s_1	11 / 20
2	s_1 s_3 s_4 s_5	s_2	17 / 20
3	s_1 s_2 s_4 s_5	s_3	16 / 20
4	s_1 s_2 s_3 s_5	s_4	13 / 20
5	s_1 s_2 s_3 s_4	s_5	16 / 20

Classification Accuracy = 73/100 = 73%

Note: Whenever we use multiple training sets, as in CV and random resampling, we are evaluating a learning method as opposed to an individual learned model

Cross-Validation: Example

- Instead of a single validation set, we can use cross-validation within a training set to select a model (e.g. to choose the best k in k-NN)



Evaluation Measures

- Classification
  - How often we classify something right/wrong
- Regression
  - How close are we to what we're trying to predict
- Ranking/Search
  - How correct are the top-k results?
- Clustering
  - How well we describe our data (Not straightforward)

Is accuracy adequate?

- Accuracy may not be useful in cases where
  - There is a large class skew
    - Is 98% accuracy good if 97% of the instances are negative?
  - There are differential misclassification costs – say, getting a positive wrong costs more than getting a negative wrong
    - Consider a medical domain in which a false positive results in an extraneous test but a false negative results in a failure to treat a disease
  - We are most interested in a subset of high-confidence predictions

Classification Error: Beyond Accuracy

In 2-class problems:

predicted class \ actual class	positive	negative
positive	true positives (TP)	false positives (FP)
negative	false negatives (FN)	true negatives (TN)

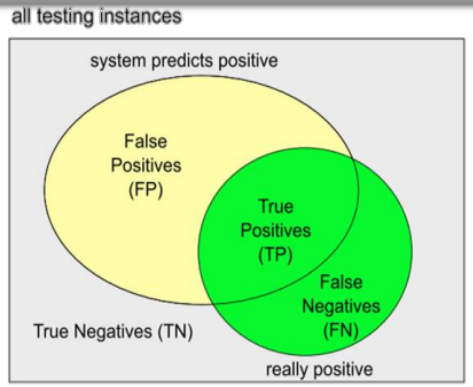
$$\text{accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

Classification Performance Measures

Predict positive?

Really positive?

	Yes	No
Yes	TP	FN
No	FP	TN



- Classification Error: $\frac{errors}{total} = \frac{FP+FN}{TP+TN+FP+FN}$
- Accuracy = 1-Error: $\frac{correct}{total} = \frac{TP+TN}{TP+TN+FP+FN}$

} meaningless if classes imbalanced

- False Alarm = False Positive rate = FP / (FP+TN)
- Miss = False Negative rate = FN / (TP+FN)
- Recall = True Positive rate = TP / (TP+FN)
- Precision = TP / (TP+FP)

} always report in pairs, e.g.: Miss / FA or Recall / Prec.

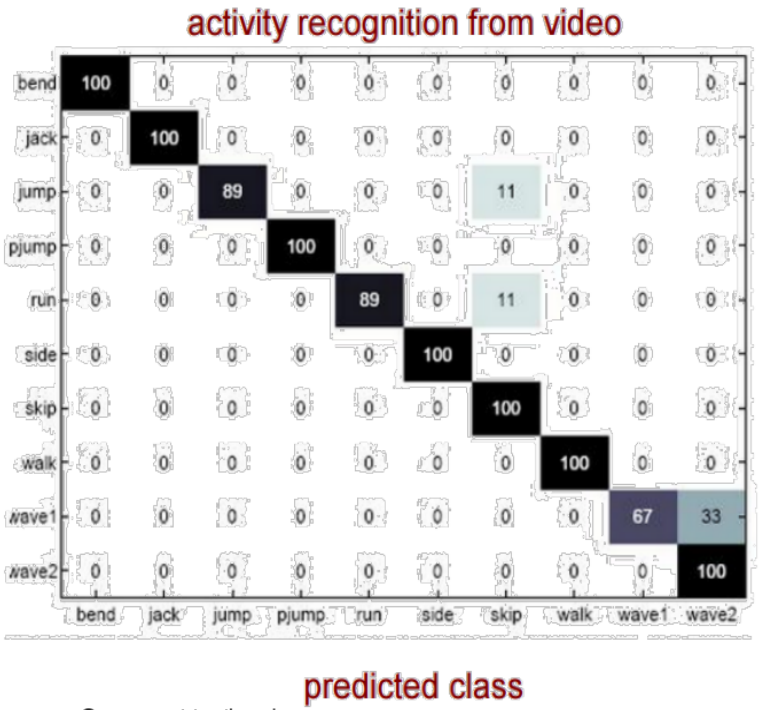
- True Positive Rate also called "Sensitivity"
- "Specificity" = 1 – False Alarm

- "Sensitivity" = Probability of a positive test given a patient has the disease
- "Specificity" = Probability of a negative test given a patient is well

Classification Error: Beyond Accuracy

For multi-class problems?

Confusion Matrix



Courtesy: vision.jhu.edu