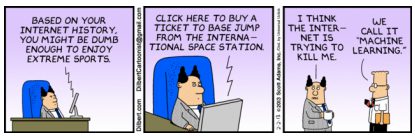
Learning from Data

Russell and Norvig Chapter 18



Learning

- Essential for agents working in unknown environments
- Learning is useful as a system construction method
  - Expose the agent to reality rather than trying to write it down
- Learning modifies the agent's decision mechanisms to improve performance

Learning from examples

- Supervised learning: Given labeled examples of each digit, learn a classification rule

Machine learning is ubiquitous

- Examples of systems that employ ML?

Examples of learning tasks

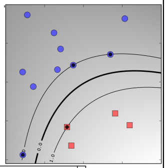
- OCR (Optical Character Recognition)
- Loan risk diagnosis
- Medical diagnosis
- Credit card fraud detection
- Speech recognition (e.g., in automatic call handling systems)
- Spam filtering
- Collaborative filtering (recommender systems)
- Biometric identification (fingerprints, iris scan, face)
- Information retrieval (incl. web searching)
- Data mining, e.g. customer purchase behavior
- Customer retention
- Bioinformatics: prediction of properties of genes and proteins.

Learning

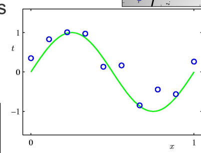
- The agent tries to learn from the data (examples) provided to it.
- The agent receives feedback that tells it how well it is doing.
- There are several learning scenarios according to the type of feedback:
  - Supervised learning: correct answers for each example
  - Unsupervised learning: correct answers not given
  - Reinforcement learning: occasional rewards (e.g. learning to play a game).
- Each scenario has appropriate learning algorithms

ML tasks

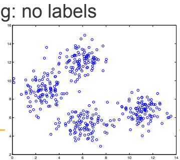
Classification: discrete/categorical labels



Regression: continuous labels



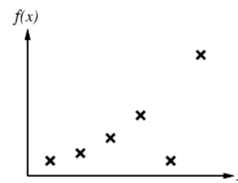
Clustering: no labels



7

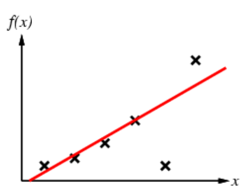
Inductive learning

- Construct a function that agrees with the training data
- Example: curve fitting



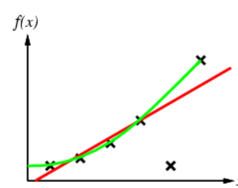
Inductive learning

- Construct a function that agrees with the training data
- Example: curve fitting



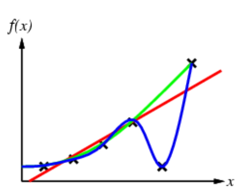
Inductive learning

- Construct a function that agrees with the training data
- Example: curve fitting



Inductive learning

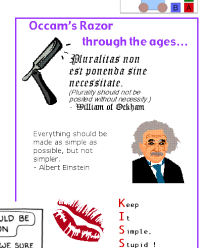
- Construct a function that agrees with the training data
- Example: curve fitting

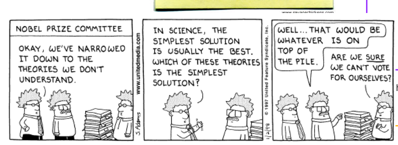


Ockham's razor: prefer the simplest hypothesis consistent with data

Learning is concerned with accurate prediction of future data, *not* accurate prediction of training data.

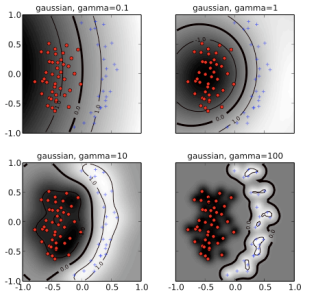
Occam's Razor





http://old.aitopics.org/AIToons

Overfitting in classification



Supervised Learning

Example: want to classify  versus 

Data: Labeled images

$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$$
$$y_i \in \{\text{monkey}, \text{human}\}$$

x_i is a vector that represents the the image

Task: Here is a new image: 
What species is it?

The Nearest Neighbor Method

(your first classification algorithm!)

NN(image):

1. Find the image in the training data which is closest to the query image.
2. Return its label.


query


closest image

Distance measures

- How to measure closeness?

Distance measures

- How to measure closeness?
- Discrete data: Hamming distance
- Continuous data: Euclidean distance
- Sequence data: edit distance
- Alternative: use a similarity measure (or dot product) rather than a distance

k-NN

- Use the closest k neighbors to make a decision instead of a single nearest neighbor
- Why do you expect this to work better?

Remarks on NN methods

- Very easy to implement
- No training required. All the computation performed in classifying an example (complexity: O(n))
- Need to store the whole training set (memory inefficient).
- Flexible, no prior assumptions (a type of non parametric classifier: does not assume anything about the data).
- Curse of dimensionality: if data has many features that are irrelevant/noisy distances are always large.

Take home question

- How would you convert the k-nearest-neighbor classification method to a regression method?

Measuring classifier performance

- Or how accurate is my classifier.

Measuring classifier performance

- The error rate on a set of examples $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$:

$$\frac{1}{n}\sum_{i=1}^{n} I(f(\mathbf{x}_i) \neq y_i)$$

I is the indicator function that returns 1 if its argument is True and zero otherwise

- What is the error rate of a nearest neighbor classifier applied to its training set?

Measuring classifier performance

- The error rate on a set of examples $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$:

$$\frac{1}{n}\sum_{i=1}^{n} I(f(\mathbf{x}_i) \neq y_i)$$

I is the indicator function that returns 1 if its argument is True and zero otherwise

- Report error rates computed on an independent **test set** (classifier was trained using **training set**): classifier performance on the training set is not indicative of performance on unseen data.

Measuring classifier performance

- The error rate on a set of examples $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$:

$$\frac{1}{n}\sum_{i=1}^{n} I(f(\mathbf{x}_i) \neq y_i)$$

I is the indicator function that returns 1 if its argument is True and zero otherwise

- Issue when classes are imbalanced.
- There are other measures of performance that address this.

Measuring classifier performance

- Split data into training set and test set (say 70%, 30%).
- Compare several classifiers trained on this split.
- Train final best classifier on the full dataset.

- A better method: **cross-validation**

Cross-validation

- Split data into k parts (E_1,…,E_k)
  for i = 1,…,k :
  training set = D\E_i
  test set = E_i
  classifier.train(training set)
  accumulate results of classifier.test(test set)

- This is called k-fold cross-validation
- Extreme version: Leave-One-Out
- Assumptions?

Uses of CV

Cross Validation is used to choose:

- Classifier parameters
  - k for k-NN
- Normalization method
- Which classifier
- Feature selection (which features provide best performance).
- This is called **model selection**

CV-based model selection

- We're trying to determine which classifier to use

Classifier	Training error	CV-error	choice
f_1			
f_2			
f_3			✔
f_4			
f_5			
f_6			

CV-based model selection

- Example: choosing k for the k-NN algorithm:

Classifier	Training error	CV-error	choice
K = 1			
K = 2			
K = 3			✔
K = 4			
K = 5			
K = 6			

- Show demo

The general workflow

- Formulate problem
- Get data

- Decide on a representation (what features to use)
- Choose a classifier
- Assess the performance of the classifier
- Depending on the results: modify the representation, classifier, or look for more data

Next

More classifiers:

- Decision trees
- How to use a probabilistic model such as a Bayesian network as a classifier