Classification

April 2010

Outline

Background

- Machine Learning: algorithms that “learn” to interpret empirical data
- Often involves imposing structure on the data
- Common Example: assigning data elements to classes

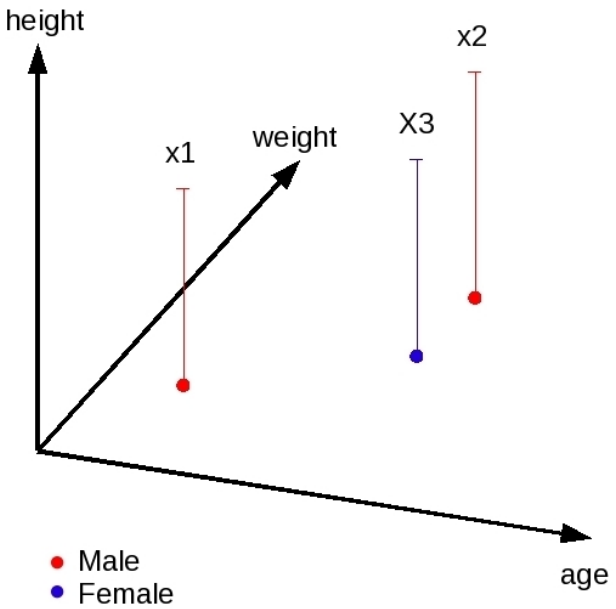
Examples

- Binary Classification - yes/no
- Multiple Classes
  - discretization: cold/warm/hot, baby/child/teenager/adult/senior
  - secondary protein structure (alpha helix / beta sheet / other)
  - named entity (person/place/organization/other)

  Note: can be reduced to multiple binary classifications
- Multi-Label
  - objects in an image
  - topics in a document
  - genes in a sequence
  - symptoms in a patient record

Data Representation

- Units of the data are *instances*.
- Instances are usually represented as a list of *features* with associated *feature values*.
- In most cases, this translates to a *vector*, where each dimension represents the value of a different feature.



$X_i(height, weight, age, sex)$

Data Representation

Issues

- different kinds of feature values
  - database record = age (discrete), sex (binary), height (real), percent disability (bounded range)
- sparse data - many zero values
  - instances = documents, features = English words, values = number of times they appear in the document
  - instances = patients, features = possible lab tests, values = results on administered tests
  - instances = images, features = objects of interest, values = whether the object is in the image

Formal Task Definition

- A classifier is a function $h : X \rightarrow Y$ that maps any instance $x \in X$ to its classification label $y \in Y$.
- In vector representation, $X = \mathbb{R}^k$.
- In binary classification, $Y = \{0, 1\}$ or $Y = \{-1, +1\}$
- In multi-class classification, Y is a finite set of possible labels, w.l.o.g. $Y = \{1, ..., m\}$
- In the multi-label setting, $Y = 2^m$, where m is the number of possible labels.

Outline

Supervision in Machine Learning

Three learning frameworks:

- Supervised
- Unsupervised
- Semi-Supervised

Supervised Learning

Two sets of data: training set and test set

- The training set: labeled examples (x_i, y_i)
  x_i is a vector of feature values
  y_i is the correct (*gold-standard*) label.
- The test set: unlabeled instances $\{x_i\}$
  (labels only available at evaluation)
- The classifier learns a classification function from training set, and is evaluated on the test set.

Unsupervised Learning

No labeled data

- find a meaningful (?) devision of the data
- detect classes that are helpful for some task
- the user might not be aware
- most common example: clustering (many variations).
- no labels, so evaluation is problematic.

Semi-Supervised Learning

Labeled and unlabeled training data + test set with (hidden) labels

- usually, little labeled data, lots of unlabeled data
- use unlabeled data to improve classification
- simple example: cluster the data, use labeled portion to assign labels to clusters

Outline

Dimensionality Reduction

- Too many features/dimensions
  - pixels in an image
  - words in a document
- Goals:
  - reduce learning requirements - time, memory
  - improve generalization: eliminate noise, avoid over-fitting
- Methods:
  - feature selection - choose the ones most likely to be helpful
  - new features that combine several existing ones
  - replace weight and height with BMI ($\frac{weight}{height^2}$)
  - replace gross income and tax with net income

Feature Selection

- Information Theory: use Mutual Information to pick features which are most informative to the class
- Hypothesis testing statistics: use Chi-square test (χ^2) to determine confidence of association between feature and class
- χ^2 measures *confidence* of association, MI measures *extent*

Mutual Information

Choose the top k features with regard to the mutual information between them and the classes:

$$I(c; f) = \sum_{\substack{e_c \in \{0,1\} \\ e_f \in \{0,1\}}} p(e_c, e_f) \log \frac{p(e_c, e_f)}{p(e_c) \cdot p(e_f)}$$

χ^2 statistic

χ^2 is interested in the relation between the observed correlations, and what you'd expect from the marginals.

$$\chi^2(class, feature) = \sum_{i \in \{c, \neg c\} \times \{f, \neg f\}} \frac{(O_i - E_i)^2}{E_i}$$

$A = count(c, f)$	$C = count(c, \neg f)$
$B = count(\neg c, f)$	$D = count(\neg c, \neg f)$

$O_i =$ the observed counts (A, B, C, or D)

$E_i =$ expected counts from the marginal,

e.g. $E_{(c,\neg f)} = p(c) \cdot p(\neg f) \cdot N = \frac{A+C}{N} \cdot \frac{C+D}{N} \cdot N$

$$\chi^2(class, feature) = \frac{N \times (AD - CB)^2}{(A+C) \times (B+D) \times (A+B) \times (C+D)}$$

χ^2 statistic

- χ^2: relation between the observed correlations, and what you'd expect from the marginals.
- $\chi^2(\mathit{class}, \mathit{feature}) = \sum_{i \in \{c, \neg c\} \times \{f, \neg f\}} \frac{(O_i - E_i)^2}{E_i}$
- The bigger the difference, the more confident we are in the correlation.
- choose top k most confident features

Outline

Accuracy

- When a test set is available, the most straightforward evaluation is accuracy:
- $Accuracy(Classifier)_{\{x_i\}_{i=1}^{n}} = \frac{1}{N}\sum_{i=1}^{N}\delta\{classifier(x_i) = y_i\}$

  i.e., the percentage of correctly classified instances in the test.

* $\delta\{statement\} = \begin{cases} 1 & \text{if } statement \text{ is true} \\ 0 & \text{otherwise} \end{cases}$

Precision, Recall

- In many situations, the classifier can't give a class for every instance.
- e.g.,: web search, low confidence
- In such cases, we can evaluate two aspects of the classifier's performance:
- $Precision(Classifier) = \frac{correctly\ labeled\ instances}{total\ labeled\ instances}$
- $Recall(Classifier) = \frac{correctly\ labeled\ instances}{total\ instances\ in\ the\ class}$

Precision, Recall and F-measure

- In general, there is a trade-off between precision and recall.

  e.g.,: what confidence threshold to use.

- F-measure attempts to give a single score accounting for both.
- $F\text{-}measure(Classifier) = \frac{2 \cdot Precision \cdot Recall}{(Precision + Recall)}$
- When $Precision = Recall$ (i.e., when the classifier labels all instances), $F\text{-}measure = Accuracy$.

Micro/Macro Average

- When dealing with multiple classes, there are two ways of reporting average performance:
  - $\textit{Macro Average} = \frac{1}{k} \sum_{i=1}^{k} score(classifier, c_i)$
  - $\textit{Micro Average} = \sum_{i=1}^{k} \frac{|c_i|}{N} score(classifier, c_i)$
- Macro-average considers all classes equal, whereas micro-average considers their relative size.

Cross Validation

- Cross validation creates artificial test sets out of the training data.
- The training data is (randomly) split into *folds*.
- Each of the folds is used in turn as a test set, with the rest of the data used for training.
- The overall performance gives an estimate of the performance of the classifier.
- Cross validation is used for:
  - tunning parameters during training
  - evaluating performance of unsupervised algorithms