Lecture 8: Text classification

William Webber (william@williamwebber.com)

COMP90042, 2014, Semester 1, Lecture 8

What we'll learn in this lecture

- The classification process
- Two simple text classification methods tied closely to vector-space model:
  - k nearest neighbours
  - Rocchio
- How to evaluate classification systems

Classification vs. clustering

- Clustering: unsupervised; machine chooses classes
- Classification: supervised; we specify classes
- Clustering: docs clustered by self-similarity
- Classification: docs classified by similarity to examples

Classification, regression, ranking

Regression estimate real output variable for doc

Ranking rank docs by some quality

Classification assign class to doc

- Binary (two-class) classification:
  - Regressed score can be probability, degree
  - If scores only relative, $\rightarrow$ ranking
  - Bifurcation at score $\rightarrow$ classification
- Many binary classification methods go score $\rightarrow$ class
- c multi-class from c binary regressions

Classification: outline

Types of classification

Rule-based Human writes rules, machine applies

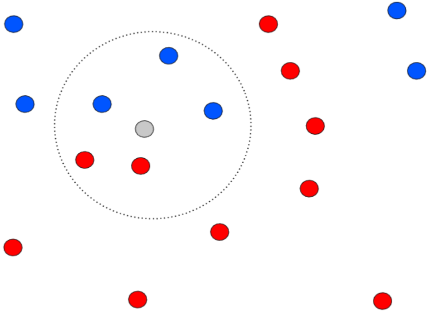
Decision tree Machine learns (discreet) rules

Statistical Machine learns statistical models

Statistical ML for classification

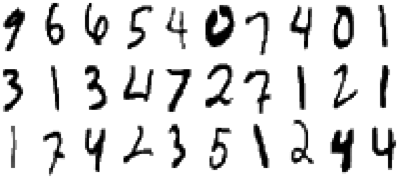
- Human labels example objects with classes (training data)
- Machine learns statistical model from examples
- Machine predicts class of unlabelled objects from model

k nearest-neighbours



- Predicted class of object *d*
- ... plurality class of *k* training objects "nearest" *d*
- Cosine distance a possible "nearness" metric for docs

k nearest-neighbours



Pros

- Good effectiveness for text
- Handles multi-class directly
- Doesn't require model to be built
- Handles any concept of "similar"

k nearest-neighbours

Cons

- Need to tune selection of k (≈ 40 for text)
- Need to adjust for unbalanced classes
- **Computationally intensive** at classification time
  - $O(n)$ for naive method (compare each item)
  - $O(\log n)$ for divide-and-conquer methods

Rocchio's method: intuition

- Saw Rocchio used for PRF (can you summarize?)
- Can also be used for classification
- Idea is:
  - Calculate mean from training docs in each class
  - Mean class document represents class
  - Classify new document by nearest class mean

Rocchio's method: implementation

- Let $\mathcal{T}_c$ be set of n training docs for class c
- Centroid docvec $\boldsymbol{\mu}_\mathbf{c}$ of c is:

$$\boldsymbol{\mu}_\mathbf{c} = \frac{1}{n} \sum_{d \in \mathcal{T}_c} \mathbf{v}(d) \tag{1}$$

  where $\mathbf{v}(d)$ is the docvec of d

- Then assigned class $c \in \mathcal{C}$ for unlabelled doc d is:

$$c = \operatorname*{argmax}_{c' \in \mathcal{C}} \cos\left(\boldsymbol{\mu}_{\mathbf{c}'}, \mathbf{v}(d)\right) \tag{2}$$

Rocchio's method: the model

- Generally less effective than kNN
- (though more effective on text data than Naive Bayes)
- Much faster to compute at run time

The model

- In Rocchio, $\boldsymbol{\mu}_\mathbf{c}$ is *model* of class c.
- Document d tested for (strength of) membership in class c using dot product
- Constant time (relative to collection size)

Classification: outline (bis)

- Human labels example objects with classes (training data)
- Machine learns statistical model from examples
- Machine predicts class of unlabelled objects from model

Classifier: labelling

- User identifies classes $\mathcal{C} = \{c_1, c_2, \ldots, c_n\}$
- User finds, or system samples, training documents $\mathcal{T}$
- User labels each document $d \in \mathcal{T}$ with its class
- Output is set $\mathcal{T}_c$ of training examples for each class c

Classifier: features

Require calculable representation of objects to be classified

- Identify set of discrete *features*
- Each object represented as a *feature vector*
  - each cell represents a feature
  - value of cell is object's weight for that feature
- Result is an object $\times$ feature matrix

Learning algorithm

- Machine learner learns *model*
  - Of class c from training examples $\mathcal{T}_c$
  - Or of overall classification decision (esp. multi-class)
- A model is a function that:
  - Takes a feature vector as input
  - Produces either:
    - Strength of membership to each class $c \in \mathcal{C}$, or
    - Single class assignment c, as output
- Models can work by:
  - Similarity (kNN, Rocchio)
  - Formula (esp. for regression; e.g. linear least squares)
  - Discrimination (finding "dividing line", e.g. SVM)

Features in text classification

For text classification:

- Objects are documents
- Terms are features
- Weights are (e.g TF*IDF) weights

Text, compared to other forms of classification:

- Very large feature set ("for free")
  - Feature design big issue elsewhere (e.g. image recognition)
- Highly correlated
  - NB works poorly without feature selection
- Sparse (most features have 0 weight for most objects)

Enhancing the feature space

- Can add non-text document aspects as features:
  - Author, length, date (with caution) of document
  - Sender, recipient of email
  - Noun phrases or *n*-grams
  - Number of punctuation marks, etc. etc.
- Enhancing features a "value add" for specialist applications

(Rough) decreasing order of importance for good classifier:

1. More training data
2. Better features
3. Better classification algorithm

Evaluation of (text) classification

- Classifier tested against labelled datasets
  - Dataset should be fully labelled
  - Often re-use set created by real-world process
- Classifier trained against one set of docs
- Then asked to predict labels of another set
  - Training and test set must be kept separate!
- Effectiveness measured by accuracy of prediction

Two cases:

1. Output is class assignment (set-based evaluation)
2. Output is strength of class membership (esp. for binary classification)

Set-based Evaluation metrics

Label		True	
		1	0
Predicted	1	TP	FP
	0	FN	TN

$$\frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}}$$ Accuracy

$$\frac{2 \cdot \mathrm{TP}}{2 \cdot \mathrm{TP} + \mathrm{FP} + \mathrm{FN}}$$ F1 score

$$\frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$ Sensitivity (TPR, Recall)

$$\frac{\mathrm{TN}}{\mathrm{FP} + \mathrm{FN}}$$ Specificity (TNR)

Set-based evaluation metrics

- Accuracy is sensitive to imbalanced classes
  - If 95% objects in class c, always guessing class c gets 95% accuracy
- F1 score (harmonic mean of recall and precision)
  - Also an IR metric
  - More robust to imbalance
  - Doesn't generalize (easily) to multiple classes
- Sensitivity and specificity generally used as ingredients in rank metrics (see next)

Rank metrics

- Binary classification often a "A" vs. "not-A" task
  - E.g. "about sports" vs. "not about sports"
  - I.e. "relevant" vs. "not relevant" to sports
- Many classifiers give real-valued prediction
- Can rank by decreasing association to class A
  - Cutoff point may be selected for binarization
- Ranking can be independently evaluated:
  - To evaluated quality of ranking (vs. of cutoff)
  - Because ranking might be end product

Rank metrics

- General IR rank metrics (e.g. AP) can be used
- Common alternative to graph contrasting measures down ranking
  - e.g. TPR vs FPR (sensitiy vs. $1 -$ specificity) at increasing ranks
- Then calculate "area under curve" (AUC) to give single measure
  - Area under TPR vs. FPR known as receiver operating characteristic, or ROC curve, or (confusingly) area under the ROC curve, or AUROC, or even AUC

RCV1-v2

CCAT	Corporate/Industrial
C11	Strategy/Plans
C15	Performance
C151	Accounts / Earnings
C1511	Annual Results
C152	Comment / Forecasts

Figure : Some RCV1v2 categories

- LYRL-30k drawn from RCV1-v2
- 800k-odd Reuters news articles
- 103 topical labels, manually assigned by Reuters curators
- Topics arranged in hierarchy
- One document can be labelled with more than one topic

Looking back and forward

Back

- Classification process: train, learn, predict
- kNN and Rocchio, simple VSM classifiers
- ... follow directly from VSM search, clustering approaches
- Set-based and ranking-based classifier evaluation

Looking back and forward

Forward

- Next lecture: support vector machines (SVM)
  - Robust and popular classifier family
  - Also based on a geometric model
- Later in course: probabilistic classification models

Further reading

- Lewis, Yang, Rose, and Li, "RCV1: A New Benchmark Collection for Text Categorization Research" (JMLR, 2004) (describes the RCV1v2 collection; also gives comparative scores for kNN, Rocchio, and SVM)
- Yang and Liu, "A re-examination of text categorization methods" (SIGIR, 1999) (compares kNN, Naive Bayes, and SVM)