CS 276 / LING 286 Information Retrieval and Web Search

Problem Set 4 Solutions

May 21, 2015
Due: 23:59PM PDT, May 28, 2015

Problem 1 - Text Classification (16 points)

In this problem, we will explore various techniques for classifying documents. Consider the following supervised corpus of news headlines, where the document class is in bold (not considered as a part of the document):

[World News] "Iraq election", "French executive injured", "Teen survives avalanche"
[Business] "Chief executive smiles", "Krispy Kreme executive resigns"

Using this corpus, we will try to predict the class of the document "executive suite".

1. Rocchio Classification Algorithm

   (a) (2 points) Compute the centroid of each class. Express the centroids and the query document as raw term frequency vectors (e.g. sparse vectors).

   Solution: .

"Iraq election"	{ Iraq: $1/\sqrt{2}$, election: $1/\sqrt{2}$ }
"French executive injured"	{ French: $1/\sqrt{3}$, executive: $1/\sqrt{3}$, injured: $1/\sqrt{3}$ }
"Teen survives avalanche"	{ Teen: $1/\sqrt{3}$, survives: $1/\sqrt{3}$, avalanche: $1/\sqrt{3}$ }
[World News]	{ Iraq: $1/(3\sqrt{2})$, election: $1/(3\sqrt{2})$, French: $1/(3\sqrt{3})$, executive: $1/(3\sqrt{3})$, injured: $1/(3\sqrt{3})$, Teen: $1/(3\sqrt{3})$, survives: $1/(3\sqrt{3})$, avalanche: $1/(3\sqrt{3})$ }
"Chief executive smiles"	{ Chief: $1/\sqrt{3}$, executive: $1/\sqrt{3}$, smiles: $1/\sqrt{3}$ }
"Krispy Kreme executive resigns"	{ Krispy: 1/2, Kreme: 1/2, executive: 1/2, resigns: 1/2 }
[Business]	{ Chief: $1/(2\sqrt{3})$, executive: $1/4 + 1/(2\sqrt{3})$, smiles: $1/(2\sqrt{3})$, Krispy: 1/4, Kreme: 1/4, resigns: 1/4 }

The query vector is { executive: $1/\sqrt{2}$, suite: $1/\sqrt{2}$ }.

The Rocchio classification algorithm involves normalization, but for this problem, we also accepted solutions without this step:

"Iraq election"	{ Iraq: 1, election: 1 }
"French executive injured"	{ French: 1, executive: 1, injured: 1 }
"Teen survives avalanche"	{ Teen: 1, survives: 1, avalanche: 1 }
[World News]	{ Iraq: 1/3, election: 1/3, French: 1/3, executive: 1/3, injured: 1/3, Teen: 1/3, survives: 1/3, avalanche: 1/3 }
"Chief executive smiles"	{ Chief: 1, executive: 1, smiles: 1 }
"Krispy Kreme executive resigns"	{ Krispy: 1, Kreme: 1, executive: 1, resigns: 1 }
[Business]	{ Chief: 1/2, executive: 1, smiles: 1/2, Krispy: 1/2, Kreme: 1/2, resigns: 1/2 }

The query vector is { executive: 1, suite: 1 }.

(b) (2 points) Determine the class of the document using the Rocchio classification algorithm.

Solution: We measure the distance between the query vector and each centroid, and choose the class with the smallest distance. With normalization, the query is classified as [Business]. Without normalization, the query is labeled [World News].

2. k-Nearest Neighbors

(a) (2 points) Assuming raw term frequency, no idf, and cosine similarity, what are the three nearest neighbors to the query document? Show cosine similarity computations.

Solution: The cosine similarity is given by $u \cdot v/(\|u\|\|v\|)$.

Document	Cosine similarity
Iraq election	0.0
French executive injured	0.40825
Teen survives avalanceh	0.0
Chief executive smiles	0.40825
Krispy Kreme executive resigns	0.35356

The three nearest documents are "French executive injured," "Chief executive smiles" and "Krispy Kreme executive resigns."

(b) (2 points) What class does this document belong to if we consider 3NN classification?

Solution: The query is classified as [Business] as two out of the three near neighbors belong to the business class.

3. Naive Bayes

(a) (2 points) Before making predictions, we will first learn some parameters from the corpus. Using the method of maximum likelihood estimation, evaluate

- $\hat{P}_{\text{MLE}}$(World News)
- $\hat{P}_{\text{MLE}}$(executive|World News)
- $\hat{P}_{\text{MLE}}$(suite|World News)
- $\hat{P}_{\text{MLE}}$(Business)
- $\hat{P}_{\text{MLE}}$(executive|Business)
- $\hat{P}_{\text{MLE}}$(suite|Business)

Solution: Recall that the MLE is the ratio of empirical counts from the training corpus. In the Bernoulli event model, $P(x|c)$ is the likelihood of whether the term x is included in a document if the document belongs to the class c. It can be estimated by dividing the number of documents containing x by the total number of documents in that class.

- $\hat{P}_{\text{MLE}}(\text{World News}) = 3/5$
- $\hat{P}_{\text{MLE}}(\text{executive}|\text{World News}) = 1/3$
- $\hat{P}_{\text{MLE}}(\text{suite}|\text{World News}) = 0/2$
- $\hat{P}_{\text{MLE}}(\text{Business}) = 2/5$
- $\hat{P}_{\text{MLE}}(\text{executive}|\text{Business}) = 2/2$
- $\hat{P}_{\text{MLE}}(\text{suite}|\text{Business}) = 0/2$

For the multinomial event model, x is a multinomial random variable whose distribution models the probability of generating a particular word in a document. The probability $P(x|c)$ is estimated by dividing the number of occurences of a certain word x in class c by the total number of words in that class.

- $\hat{P}_{\text{MLE}}(\text{World News}) = 3/5$
- $\hat{P}_{\text{MLE}}(\text{executive}|\text{World News}) = 1/8$
- $\hat{P}_{\text{MLE}}(\text{suite}|\text{World News}) = 0/8$
- $\hat{P}_{\text{MLE}}(\text{Business}) = 2/5$
- $\hat{P}_{\text{MLE}}(\text{executive}|\text{Business}) = 2/7$
- $\hat{P}_{\text{MLE}}(\text{suite}|\text{Business}) = 0/7$

(b) (2 points) What could be a potential problem if predictions are to be made with these estimates? Propose a method to cope with this and briefly discuss its justification.

Solution: Naive Bayes treats unseen examples as impossible. Laplace smoothing allows us to estimate a non-zero probability for unseen data. This is motivated from a uniform Dirichlet prior.

(c) (2 points) If we choose the class by $\arg\max_C \hat{P}(C|\text{“executive suite”})$, what class is this document assigned to?

Solution: **Bernoulli event model** To predict with this model, we also have to account for terms not appearing in the document: that is, we need to multiply the complementary probability of terms like avalanche. If we let x be a vector of whether term i appears in the document, then

$$\begin{aligned} P([\text{World News}]|\text{“executive suite”}) &\propto P([\text{World News}]) \prod_i P(x_i|[\text{World News}]) \\ &= \frac{3+1}{5+2} \cdot \frac{1+1}{3+2} \cdot \frac{0+1}{2+2} \cdot \left(1 - \frac{1+1}{3+2}\right)^7 \left(1 - \frac{0+1}{0+2}\right)^5 \\ &\approx 4.9989 \times 10^{-5} \end{aligned}$$

$$P([\text{Business}]|\text{“executive suite”}) \propto P([\text{Business}]) \prod_i P(x_i|[\text{Business}])$$
$$= \frac{2+1}{5+2} \cdot \frac{2+1}{2+2} \cdot \frac{0+1}{2+2} \cdot \left(1 - \frac{1+1}{2+2}\right)^5 \left(1 - \frac{0+1}{0+2}\right)^7$$
$$\approx 1.9618 \times 10^{-5}$$

The query belongs to World News.

Multinomial event model We first let the dictionary be the set of all tokens in the corpus and query (in practice, we handle cases like suite by including a special token, UNK, for unknown words). Then, with smoothing we obtain

$$P([\text{World News}]|\text{“executive suite”}) \propto P(\text{executive}|[\text{World News}])P(\text{suite}|[\text{Worlds News}])$$
$$P([\text{World News}])$$
$$= \frac{1+1}{8+14} \cdot \frac{0+1}{8+14} \cdot \frac{3+1}{5+2} \approx 0.00236$$

$$P([\text{Busines}]|\text{“executive suite”}) \propto P(\text{executive}|[\text{Business}])P(\text{suite}|[\text{Business}])P([\text{Business}])$$
$$= \frac{2+1}{7+14} \cdot \frac{0+1}{7+14} \cdot \frac{2+1}{5+2} \approx 0.00292$$

The query is classified as Business since it has a higher posterior probability.

4. (2 points) Intuitively, which class should “executive suite” be labeled as? Did all three methods agree on this? Give a brief explanation for this behavior.

   Solution: (This depends on the student's answer). Not every method agreed on the label, despite the fact that these techniques tend to work nicely even with small datasets. This shows that there is no free lunch; a method does not always work for all data. This is why we need to have a good evaluation and evaluate the performace of various models.

Problem 2 - Support Vector Machine (18 points)

Suppose we are building a ranking function for a search engine using a linear combination of the cosine score and the width of the smallest window in the document containing the query terms (and no other features). Specifically, we train an SVM to classify query-document pairs as Relevant or Non-relevant, from the following 6 examples:

Query	DocID	Cosine score	Window width	Judgment
social media	7	0.1	2	Non-relevant
new york	7	0.2	3	Non-relevant
asparagus soup	12	0.3	3	Relevant
new york	12	0.1	3	Non-relevant
asparagus soup	31	0.2	2	Relevant
new york	31	0.3	2	Relevant

1. (2 points) Identify the support vectors that separate the Relevant examples from the Non-relevant ones.

Solution: Support vectors that separate relevant from non-relevant documents are the following (the coordinates are specified as (cosine score, window width)):

$$\{(0.2, 2), (0.3, 3), (0.1, 2), (0.2, 3)\}$$

2. (2 points) Write an equation of the optimal separating hyperplane. What is the corresponding (geometric) margin?

   Solution: The SVM's separating line is $20(\text{cosine score}) - 2(\text{window width}) + 1 = 0$. The geometric margin is 0.0995.

3. (2 points) Given the following query-document pair, would our SVM classify this pair as Relevant or Non-relevant?

Query	DocID	Cosine score	Window width	Judgment
new york chowder	31	0.2999	4	?

   Solution: To classify $(0.2999, 4)$, we check the sign of $20 \cdot 0.2999 - 2 \cdot 4 + 1 = -1.002 < 0$. This pair is non-relevant.

4. (2 points) Why might it be misleading to use window width independent of the number of query terms as a feature?

   Solution: It is misleading to use window width independent of the number of query terms because longer queries will generally cause window width to increase, independent of whether or not the document is relevant.

5. (3 points) Suppose we use the training set above, but replace the fourth (window width) column of the table above to instead be the ratio of the number of query terms to the window width. Thus, the first example would have a ratio of 1 and the second example a ratio of 2/3, and so on. Retrain the SVM based on this new table. Namely, identify the support vectors and find an equation of the decision boundary. What is the corresponding (geometric) margin?

   Solution: The new support vectors are the following (here the second coordinate represents the ratio of the number of query terms to window width):

$$\{(0.2, 1), (0.3, 2/3), (0.1, 1), (0.2, 2/3)\}$$

   The SVM's separating line here is

$$20(\text{cosine score}) + 6(\text{ratio}) - 9 = 0.$$

   The geometric margin is 0.09578.

6. (2 points) For this new SVM, what is the classification of the query-document pair in Part 3?

Solution: The new point is $(0.2999, 3/4)$. To classify, we check $20 \cdot 0.2999 + 6 \cdot 3/4 - 9 = 1.498 > 0$. Hence, $(0.2999, 3/4)$ is relevant.

7. (3 points) A pairwise ranking classifier takes two query-document features as input and predicts which of the two is more relevant for the given query. Show that is is possible to construct such a classifier using the result from Part 5. Write an equation of the separating hyperplane.

   Solution: Given two feature vectors $(\text{cosine score}_1, \text{ratio}_1)$ and $(\text{cosine score}_2, \text{ratio}_2)$, we want to classify a pair that yields a greater relevance score. So, we classify pair 1 to be more relevant if $20(\text{cosine score}_1) + 6(\text{ratio}_1) - 9 > 20(\text{cosine score}_2) + 6(\text{ratio}_2) - 9$. The equation of the separating hyperplane is

$$20(\text{cosine score}_1) - 20(\text{cosine score}_2) + 6(\text{ratio}_1) - 6(\text{ratio}_2) = 0.$$

8. (2 points) Use the classifier from Part 7 to determine which document is more relevant for the query below.

Query	DocID	Cosine score	Window width
new york chowder	31	0.2999	4
	30	0.25	3

Solution: The query pairs are $(0.2999, 3/4)$ and $(0.25, 3/3)$. We check $20(0.2999) - 20(0.25) + 6(3/4) - 6(1) = -0.502 < 0$. Document 30 is more relevant than 31.

Problem 3 - Information Retrieval Evaluation (16 points)

Suppose we have a collection of 8 documents, $d_1, \ldots, d_8$, which have been judged for relevance to a query. A 3-point relevance scale was used, so relevant documents (on a binary relevance scale) have been divided into Perfect and just Relevant results. Weights for these levels are shown below for NDCG:

Perfect	2
Relevant	1
Non-relevant	0

Here are the documents and their judgments:

$$\begin{aligned} \text{Perfect} &= \{d_1, d_4, d_8\} \\ \text{Relevant} &= \{d_2, d_7\} \\ \text{Non-relevant} &= \{d_3, d_5, d_6\} \end{aligned}$$

Consider now these two ordered result lists:

$$\begin{aligned} R_1 &= \langle d_1, d_2, d_7, d_3, d_4, d_8, d_6, d_5 \rangle \\ R_2 &= \langle d_1, d_4, d_8, d_2, d_6, d_3, d_5, d_7 \rangle \end{aligned}$$

1. (2 points) Is one result list better than the other based on Precision @3?

   Solution: No, both precisions are 1.

2. (2 points) Is one result list better than the other based on Precision @5?

   Solution: No, both precisions are 4/5.

3. (4 points) What is the average precision of each result list?

   Solution: .

List	Average Precision	NDCG
R_1	0.9267	0.8821
R_2	0.925	0.9900

4. (4 points) Work out the NDCG of each result list:

$$NDCG(Q,k) = \frac{1}{|Q|} \sum_{j=1}^{Q} Z_{j,k} \sum_{m=1}^{k} \frac{2^{R(j,m)} - 1}{\log_2(1+m)}.$$

   Solution: The maximum DCG score of 7.210312 is achieved when the result list is sorted in Perfect, Relevant and Non-relevant order. Normalization yields the NDCG scores in the table above.

5. (2 points) Intuitively, which ranked result list seems better for web search? Why?

Solution: R_2 is intuitively better, because it has all three perfect results in the top 3 spots so that the user does not have to scroll further.

6. (2 points) What does this say about choice of evaluation metrics for web search?

   Solution: We need to choose metrics carefully when training and evaluating a web search engine, as poor choices of metrics can lead to unideal search results.