HW1: Text Classification

Alex Saich
asaich@college.harvard.edu

Colton Gyulay
cgyulay@college.harvard.edu

February 8, 2016

1 Introduction

For the first assignment of *CS287* we are tasked with text classification, specifically within the realm of movie reviews. Our datasets consisted of short strings of words each labeled with a rating in the range $1-5$. These ratings correspond to each reviewers' sentiment in regard to the film, where lower ratings correspond generally to negative language.

We utilized three different models to learn the statistical associations between specific vocabulary and sentiment. These include a naive Bayes model (Murphy, 2012), a logistic regression model, and a linear support vector machine (Wang and Manning, 2012). The naive Bayes model learned the underlying class distribution and token distribution within these classes, which was then used to predict ratings on examples. The logistic regression and linear SVM models were trained using mini-batch stochastic gradient descent.

2 Problem Description

2.1 Dataset

Our data initially came in the format of a string of words (for example, "undeniably fascinating and playful fellow") associated with a class y in the range $1-5$. The total set of vocabulary $\mathcal{V}$ seen across training examples included around 17000 words. We converted each example to a vector $\boldsymbol{x}_{1\times\mathcal{V}}$ where each index represented the number of occurrences of a specific token within that example.

2.2 Naive Bayes

For the naive Bayes model, we first compiled a prior probability matrix $\boldsymbol{Y}_{1\times c}$ to model $P(y)$. We then built a class-word probability matrix $\boldsymbol{M}_{c\times\mathcal{V}}$ that indicates the probability that a word is associated with a given class. $\boldsymbol{M}$ models $P(x|y)$.

2.3 Logistic Regression and L_2 SVM

In general, homeworks will be specified using informal language. As part of the assignment, we expect you to write-out a definition of the problem and your model in formal language. For this class, we will use the following notation:

- $\boldsymbol{b}, \boldsymbol{m}$; bold letters for vectors.
- $\boldsymbol{B}, \boldsymbol{M}$; bold capital letters for matrices.
- $\mathcal{B}, \mathcal{M}$; script-case for sets.
- b_i, x_i; lower case for scalars or indexing into vectors.

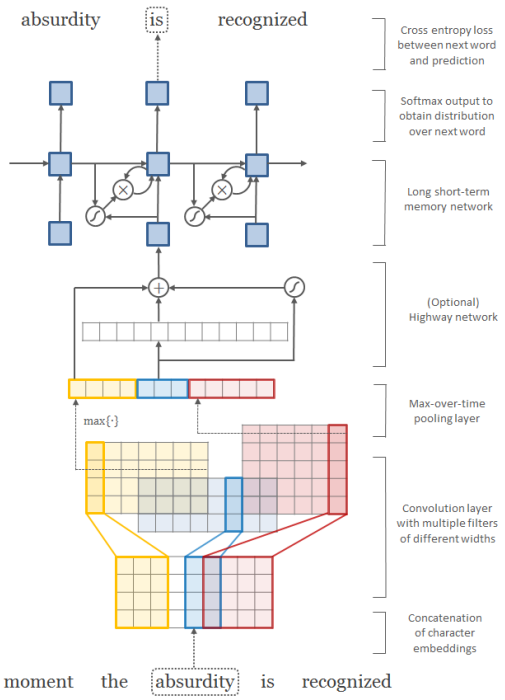
For instance in natural language processing, it is common to use discrete sets like $\mathcal{V}$ for the vocabulary of the language, or $\mathcal{T}$ for a tag set of the language. We might also want one-hot vectors representing words. These will be of the type $\boldsymbol{v} \in \{0,1\}^{|\mathcal{V}|}$. In a note, it is crucial to define the types of all variables that are introduced. The problem description is the right place to do this.

3 Model and Algorithms

Here you specify the model itself. This section should formally describe the model used to solve the task proposed in the previous section. This section should try to avoid introducing new vocabulary or notation, when possible use the notation from the previous section. Feel free to use the notation from class, but try to make the note understandable as a standalone piece of text.

This section is also a great place to include other material that describes the underlying structure and choices of your model, for instance here are some example tables and algorithms from full research papers:

- diagrams of your model,



- feature tables,

Mention Features	
Feature	Value Set
Mention Head	$\mathcal{V}$
Mention First Word	$\mathcal{V}$
Mention Last Word	$\mathcal{V}$
Word Preceding Mention	$\mathcal{V}$
Word Following Mention	$\mathcal{V}$
# Words in Mention	$\{1, 2, \ldots\}$
Mention Type	$\mathcal{T}$

- pseudo-code,

procedure LINEARIZE($x_1 \ldots x_N$, K, g)
  $B_0 \leftarrow \langle(\langle\rangle, \{1, \ldots, N\}, 0, \boldsymbol{h}_0, \mathbf{0})\rangle$
  for $m = 0, \ldots, M-1$ **do**
    for $k = 1, \ldots, |B_m|$ **do**
      for $i \in \mathcal{R}$ **do**
        $(y, \mathcal{R}, s, \boldsymbol{h}) \leftarrow \text{copy}(B_m^{(k)})$
        for word w in phrase x_i **do**
          $y \leftarrow y$ append w
          $s \leftarrow s + \log q(w, \boldsymbol{h})$
          $\boldsymbol{h} \leftarrow \delta(w, \boldsymbol{h})$
        $B_{m+|w_i|} \leftarrow B_{m+|w_i|} + (y, \mathcal{R} - i, s, \boldsymbol{h})$
        keep top-K of $B_{m+|w_i|}$ by $f(x, y) + g(\mathcal{R})$
  return $B_M^{(k)}$

4 Experiments

Finally we end with the experimental section. Each assignment will make clear the main experiments and baselines that you should run. For these experiments you should present a main results table. Here we give a sample Table 1. In addition to these results you should describe in words what the table shows and the relative performance of the models.

Besides the main results we will also ask you to present other results comparing particular aspects of the models. For instance, for word embedding experiments, we may ask you to show a chart of the projected word vectors. This experiment will lead to something like Figure 1. This should also be described within the body of the text itself.

5 Conclusion

End the write-up with a very short recap of the main experiments and the main results. Describe any challenges you may have faced, and what could have been improved in the model.

Model	Acc.
BASELINE 1	0.45
BASELINE 2	2.59
MODEL 1	10.59
MODEL 2	13.42
MODEL 3	7.49

Table 1: Table with the main results.

Figure 1: Sample qualitative chart.

References

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

Wang, S. and Manning, C. D. (2012). Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 90–94. Association for Computational Linguistics.