Applied Machine Learning Midterm Project

Justin Choi
jc3547@cornell.edu

Andrew Park
abp73@cornell.edu

Gabriel Zhen
gz267@cornell.edu

Abstract

In this project, we confronted a challenging task of classifying a dataset with 60% unlabeled data. We initiated our approach by executing a thorough preprocessing of the data to fill in the missing labels, employing diverse unsupervised learning techniques, and eventually predicting outcomes via supervised learning models. Our team employed various combinations of preprocessing methodologies including [1] GloVe, and N-grams. For the labeling process, we applied two unsupervised learning techniques, which are K-Means and Gaussian Mixture Model (GMM), to each preprocessed dataset.

The newly labeled data served as a foundation for the application of supervised learning algorithms. We employed K-Nearest Neighbors (KNN), Softmax Regression, and Multinomial Naive Bayes, each offering unique insights into the data's underlying patterns and correlations. This methodological diversity allowed us an empirical exploration of the intricate relationships among varied preprocessing, unsupervised, and supervised learning combinations.

Through a rigorous and comprehensive application of these methodologies, we derived valuable insights into the correlational dynamics amongst the different combinations of applied techniques. The final prediction accuracy of our models culminated at a score of 0.52, indicative of the strategies' effectiveness and the complexities inherent in the semi-labeled dataset. This project underscores the intricate balance and synergies between diverse preprocessing techniques and machine learning algorithms in enhancing the accuracy and reliability of predictions in scenarios complicated by incomplete data labeling.

1. Methods

We systematically addressed a dataset with 60% missing labels by employing a three-staged approach detailed in the Methods section: preprocessing, labeling by unsupervised learning, and evaluating predicted labels with supervised algorithms. Given the challenge of discerning the dataset's nature due to a significant portion of missing labels, we adopted diverse and logically appropriate strategies at each phase to ensure a comprehensive and informed analysis. Each methodology was selected for its suitability and potential efficacy in the context of the obscured dataset characteristics, aiming for an optimal balance of accuracy and insight derivation amidst the prevalent data incompleteness.

1.1. Preprocessing

In the preprocessing phase of our project, we crafted a code for data cleaning, serving as a foundation for subsequent processing techniques like N-grams, and GloVe. This preprocessing step involved the elimination of various special characters and redundant elements, coupled with the conversion of text to lowercase to achieve data simplification. This thorough cleaning process, essential for enhancing the data's quality and consistency, ensured us to perform intricate machine learning applications at Part 1, and Part 2. Consequently, the cleaned data, stripped of noise and inconsistencies, became instrumental for in-depth analysis and model training, ensuring that the ensuing machine learning applications were grounded on accurate, consistent, and analyzable text data.

Following the initial preprocessing, our methodology is split into two distinct preprocessing pathways: the implementation of N-grams, GloVe separately. Each technique was selected for specific, strategic reasons to extract maximal insights from the data.

a. N-grams: On the other hand, N-grams offered a nuanced approach by capturing the contextual dependencies and sequential order of terms. Despite the missing labels, N-grams was expected to facilitate a more in-depth analysis, enabling the extraction of meaningful patterns and relationships from the sequential arrangement of words. This was crucial in unveiling implicit connections and insights that were not immediately apparent due to the absence of labels.

b. GloVe: GloVe embeddings were utilized to convert words into vector representations that encapsulate semantic meanings, offering a deeper, semantic layer of analysis beyond term frequency and sequence. Despite the substantial portion of unlabeled data we had, GloVe enabled the revelation of underlying semantic structures and themes within the text.

By implementing each of the methodologies, our team wanted to examine both broad and detailed analyses, ensuring a comprehensive extraction of features and insights essential for the subsequent machine learning phases. N-grams enriched this analysis by unveiling intricate patterns and contexts embedded within the sequential data, and lastly, GloVe added a layer of semantic understanding.

1.2. Part 1: Labeling with Unsupervised Learning Algorithms

In the context of a dataset with 60% unlabeled data, we applied unsupervised learning algorithms post a sophisticated preprocessing phase utilizing N-grams and GloVe methods.

1. K-Means Application: K-Means was chosen because it's simple and effective at grouping data into clear clusters. Different results were expected based on the specific preprocessing method used. With N-grams, the algorithm was expected to find more complex patterns due to the rich sequence and context information, and when it's paired with GloVe data, it was hoped that K-Means would identify clusters with deeper semantic meanings, revealing subtle connections and main ideas. We decided to initialize the cluster centroids randomly because we conducted various different initialization and found random initialization give us better results.

2. GMM Application: GMM, which can handle data variance and allows for flexible clustering, was expected to produce different results depending on the preprocessing method. With N-grams, GMM was likely to reveal clusters showing detailed relationships between terms. For data processed with GloVe, the hope was that GMM would identify clusters with strong semantic meanings, capturing the deep connections present in the data. We also decided to initialie the GMM clusters with random values due to high dimensionality of the data. For training of the GMM model, we experimented with different values of random state and max iteration. Any iteration higher than 50 doesn't seem to change the final results at all.

After each model is trained, we fill in the missing labels using a two-step approach. For each cluster clustered by the model, we first filter out the labeled data in the cluster and compute the most dominant label in the cluster, then we assign the most dominant label to the unlabeled data in the same cluster.

The combination of K-Means and GMM, augmented by the two preprocessing strategies, promised a holistic and diverse approach to attributing labels to the 60% unlabeled data. Each pair of preprocessing and unsupervised learning algorithms was anticipated to provide unique, complementary insights. This well-rounded strategy was instrumental in ensuring that the labeled dataset emerged enriched, diverse, and reflective of multifaceted perspectives, setting a solid groundwork for the subsequent phases of supervised learning applications in Part 2.

1.3. Part 2: Evaluating the predicted labels with Supervised Learning Algorithms

Following the intricate processes of preprocessing and unsupervised labeling from Part 1, which resulted in four distinct variations, Part 2 of our project focuses on evaluating these labeled data through the application of supervised learning algorithms. Specifically, we deployed K-Nearest Neighbors (KNN), Softmax Regression, and Multinomial Naive Bayes, each contributing unique evaluative perspectives based on the labeled data variations.

1. K-Nearest Neighbors (KNN) Application: KNN was picked because it's straightforward and focuses on learning from specific instances. Its strength lies in looking at how close instances are to each other, providing a detailed understanding of labeled data. When paired with the data processed by K-Means, the team believed that the simple structure of K-Means might work well with KNN's focus on individual instances, giving detailed insights into specific patterns. When used with more intricate methods like N-grams or GMM, KNN might reveal detailed patterns rich in context.

2. Softmax Regression Application: We used Softmax Regression because it's good at handling multiple classes and gives a chance-based view, which is useful when looking at the flexible clusters from GMM. When combined with data from K-Means, the team believed that Softmax Regression would give clear insights into the likelihood of class memberships. For more detailed combinations using N-grams or GloVe, the algorithm might offer detailed views on class probabilities.

3. Multinomial Naive Bayes Application: We used Multinomial Naive Bayes because it's based on probability and assumes features are independent, giving us a unique way to evaluate the data. For the data processed with K-Means, we thought Multinomial Naive Bayes would give clear insights based on probabilities, especially when understanding patterns based on word counts. With richer data from N-grams and GMM, the algorithm might reveal detailed themes due to its probability focus.

2. Results

Preprocessing Method	Unsupervised Model	Supervised Model	Score
N-grams	K-Means	KNN	0.511
		Softmax Regression	**0.502**
		Multinomial NB	0.504
	GMM	KNN	0.510
		Softmax Regression	0.505
		Multinomial NB	0.503
GloVe	K-Means	KNN	**0.520**
		Softmax Regression	0.504
		Multinomial NB	0.506
	GMM	KNN	**0.520**
		Softmax Regression	0.504
		Multinomial NB	0.506

Table 1. Results for each combination

3. Discussions

3.1. What resulted in good performance?

K-Nearest Neighbors performed the best among all three supervised learning models. Since KNN is non-parametric, meaning it makes no underlying assumptions about the distribution of the dataset, it is the best fit for this dataset. We know beforehand that the labels are not evenly distributed.

3.2. What resulted in bad performance?

Softmax Regression did not perform as good as we'd expected due to model bias where one class/label is significantly more prevalent than others in the training data, the model becomes biased towards predicting that class.

3.3. What about label filling in part 1?

Both K-Means and GMM cluster data in a very similar fashion where one of the clusters tend to have more than five-holds of data than the other clusters combined. Again, this is due to the labels are not evenly distributed in the first place.

However, it's worth noticing that GMM seems to have more spreadout distributions among the rest of the clusters. We weren't able to visualize the data in a 2D plane, but our assumption was the shape of the clusters must be ellipsoid.

3.4. GloVe embeddings are better than N-grams in this context.

Scores above have shown that, in most cases, using GloVe embeddings result in higher accuracy compared to N-grams in the context of this project.

We believe this is because GloVe captures the semantic relationship between words by analyzing global statistical information from the vocabulary.

This means that words with similar meanings will have similar vector representations. For sentiment analysis, where subtle differences in sentiment can hinge on nuanced word meanings, this deep semantic understanding is crucial and makes a big difference.

In addition to semantic understanding, GloVe also reduces noise and capture words that play a bigger role in determining the sentiment of a phrase. On the other hand, N-grams introduce noise, especially as the n-gram size increases.

4. References

[1] Pennington, Jeffrey, et al. "GloVe: Global Vectors for Word Representation." Glove: Global Vectors for Word Representation, Stanford University, nlp.stanford.edu/projects/glove/. Accessed 20 Oct. 2023.