Question:-3

Report on I-Vector Classification Using LDA, Cosine Scoring, Overcomplete Representation, and Gaussian Classifier

spate235- Shaiv patel

1. Introduction

The task involves classifying I-vectors based on different machine learning techniques, including Linear Discriminant Analysis (LDA), cosine scoring, overcomplete representation, and Gaussian classifier. The I-vectors represent the acoustic features of speech samples, and the goal is to classify them correctly into one of the 24 language classes. This report discusses the approaches used, analyzes the obtained results, and provides a comparison between the different methods.

2. Approach and Methods

2.1 LDA + Cosine Scoring

In the first approach, LDA was applied to reduce the dimensionality of the I-vectors while maximizing the separability between the language classes. After the LDA projection, cosine similarity was used as the scoring function for classification. This method relies on the fact that LDA helps find a lower-dimensional subspace that best discriminates between the classes. Cosine scoring is a simple yet effective metric, which calculates the similarity between two vectors based on their angles.

Results:
Development Set Accuracy: 85.88%
Evaluation Set Accuracy: 70.29%

Analysis: The LDA + Cosine scoring method performed well on the development set, achieving a high accuracy of 85.88%. However, there was a noticeable drop in accuracy on the evaluation set (70.29%). This indicates that while the model generalized well during training, it struggled with unseen data, possibly due to the differences in distribution between the development and evaluation sets.

2.2 Overcomplete Space Representation

In this method, an overcomplete dictionary was created for sparse representation of the I-vectors. Sparse coding was applied using Lasso regression to find a sparse representation of each I-vector, and classification was based on the reconstruction error for each class. This method aims to represent the input data as a sparse linear combination of dictionary atoms, which can capture more complex patterns.

Results:
Development Set Accuracy: 64.12%
Evaluation Set Accuracy: 29.38%

Analysis: The performance of the overcomplete representation method was significantly lower compared to LDA + Cosine scoring, especially on the evaluation set. The accuracy dropped to 29.38%, suggesting that the overcomplete representation struggled with generalization. One possible reason is the high dimensionality and complexity of the sparse representation, which may have led to overfitting on the training data. Additionally, without the LDA projection, the I-vectors might not be well-separated in the sparse space.

2.3 Overcomplete + LDA

To address the issues observed in the previous method, LDA was applied first to the I-vectors before performing sparse coding. This combined approach leverages the discriminative power of LDA while still utilizing the overcomplete representation for classification.

Results:
Development Set Accuracy: 84.38%
Evaluation Set Accuracy: 70.50%

Analysis: The results improved significantly when LDA was applied before sparse coding. Both the development and evaluation set accuracies increased, suggesting that the LDA projection helped in separating the classes before sparse representation. The performance is now comparable to LDA + Cosine scoring, with a slight improvement on the evaluation set (70.50%). This indicates that the LDA projection enhances the separability of the I-vectors, making the subsequent sparse coding more effective.

2.4 K-Means Codebook + Overcomplete Representation

To further improve the efficiency of the classification process, a codebook was created using K-means clustering. Each language class was represented by 55 centroids, reducing the computational complexity while still capturing the essential features of each class. The sparse coding was then applied after the LDA projection.

Results:
Development Set Accuracy: 81.71%
Evaluation Set Accuracy: 68.54%

Analysis: This method showed good performance, though slightly lower than the previous approach (Overcomplete + LDA). The use of a K-means codebook reduced the number of atoms per class, leading to a faster and more efficient classification process. However, the reduction in centroids might have caused a loss of some discriminative information, leading to a small drop in accuracy.

2.5 Gaussian Classifier

In the final approach, a Gaussian classifier was used with the assumption of equal priors and a shared covariance matrix across all language classes. This method treats each class as a Gaussian distribution and calculates the likelihood of each I-vector belonging to a particular class.

Results:
Development Set Accuracy: 83.67%
Evaluation Set Accuracy: 70.54%

Analysis: The Gaussian classifier achieved comparable results to the LDA-based methods, with slightly lower accuracy on the development set (83.67%) but similar performance on the evaluation set (70.54%). The use of a shared covariance matrix helped stabilize the model and prevent overfitting. The performance indicates that the Gaussian assumption is a reasonable approximation for this dataset.