Final Report

Group: zijufeng

Member: Ziju Feng

1. **Implemented methods**

* ***Blogs***

The implemented methods for blogs data include:

* Nearest Neighbor method
* SVM with vector space kernel
* SVM with generalized vector space kernel
* SVM with latent semantic kernel
* SVM with string kernel

The SVM with string kernel method regards the whole document as one string feature and the kernel function of two documents refers to the similarity of the two corresponding strings. The preprocessing stage for this method is as follows: Punctuation and the words that occur in a stop list are removed, while spaces are kept in their original places in the documents.

All other methods share the same preprocessing procedures: Stop words and punctuation are removed from the documents. Words that have the same form after word stemming are mapped to the same term. Entries of feature vectors are weighted by tf-idf.

Vector space kernel is computed simply by the dot product of two weighted vector feature, each of which is represented as a list of weighted terms so that the cost of computation is much smaller.

Generalized vector space kernel represents a document by a vector of its similarities with the different documents in the corpus, in other words a document is represented by the product of its vector feature and the term-document matrix so that semantic similarity between terms is introduced.

Latent semantic kernel project a vector feature of a document onto the subspace spanned by the first k singular vectors of the feature space with SVD of the term by document matrix so that semantic relations between terms are introduced.

* ***Images***

The implemented methods for images data include:

* SVM with a radial basis function kernel
* Eigenface method
* Fisherface method

The differences between the SVM method and the baseline method include: using HSV color space, creation of some virtual examples, position of cropping window, some parameters.

Eigenface method performs PCA method to reduce the number of dimensions and then uses a nearest neighbor method to classify the test data.

Fisherface method tries to reduce dimensions in such a way that the ratio of the between-class scatter and the within-class scatter is maximized.

1. **Results**

* ***Blogs***

Some results are based on test set, which are denoted by (t), while some results are based on cross-validation, which are denoted by (c). The result of string kernel is based on a much smaller training set and test set because of the extremely high cost of computation. The results are as follows:

|  |  |  |
| --- | --- | --- |
| Method | Gender accuracy | Age accuracy |
| Nearest Neighbor | 0.553(c) | 0.392(c) |
| Vector space kernel | 0.722(t) | 0.746(t) |
| Generalized vector space kernel | 0.662(t) | 0.617(t) |
| Latent semantic kernel | 0.671(c) | 0.636(c) |
| String kernel | 0.643(c) | 0.575(c) |

* ***Images***

Some results are based on test set, which are denoted by (t), while some results are based on cross-validation, which are denoted by (c). The results are as follows:

|  |  |  |
| --- | --- | --- |
| Method | Gender accuracy | Age accuracy |
| SVM | 0.775(t) | 0.500(t) |
| Eigenface | 0.543(c) | 0.335(c) |
| Fisherface | 0.642(c) | 0.389(c) |

1. **Analysis**

* ***Blogs***

Nearest neighbor method is not suitable for this task because the dimensions are very high and there are numerous irrelevant features. One possible way to fix these problems are to perform clustering to reduce the number of dimensions and reduce the impact of irrelevant features.

SVM with vector space kernel performs quite well here. The introduction of tf-idf weight and stop list and word stemming provides some semantic information of the documents, which improves the performance of this method.

SVM with generalized vector space kernel is too naïve in its use of the co-occurrence information and therefore it doesn’t help to improve the performance.

SVM with latent semantic kernel is a better way to make use of the co-occurrence information and create more refined semantics. The performance of LSK depends greatly on the choice of parameter k. I did try several values, but a better way of doing this is to use the cross-validation to decide the optimal value.

String kernel suffers from the problem of high computational cost. One possible way to fix it is to apply some approximation approach, which can dramatically reduce the running time. However, any approximation approach may impair the performance of string kernel.

* ***Images***

SVM with HSV color space and virtual examples performs a little better than the baseline method. However, pixels of colors are not a good set of features because they are sensitive to transformation and they provide little information about the shape. One possible way solving this problem may be integrating canny edge and high level histogram features.

Eigenface method performs badly in this task. This might be caused by an obvious drawback of this approach that the scatter being maximized is due not only to the between-class scatter that is useful for classification, but also to the within-class scatter that, for classification purposes, is unwanted information.

Fisherface method performs better than eigenface method because it make use of the label information and maximizes the ratio of between-class and within-class scatter. It performs really well in certain kind of face recognition task, for example for one seeking insensibility to lighting conditions. As for this task, possible ways of improving this method include doing a more complex classification rather than the trivial nearest neighbor, adding virtual examples. I tried the average distance across class instead of nearest neighbor, and it performed a little bit better, but it still was not as good as the SVM method.

1. **Visulization**

The following is an eigenface with 40 eigenvectors.

