Dimension Reduction

- G53MLE -

Assignment 3 Report

Group 2 - House Stark

Ting Pan, Junbin Wang, Yu Ding, Bolaji Abiodun, Bert Dzambulatov

December 2, 2016

University of Nottingham
Computer Science

Introduction

Machine learning with a data set that has an enormous amount of features is time consuming. Dimension Reduction (DR) is a data preprocessing technique that preserves the most salient information and removes the data that is irrelevant to analysis and prediction. By focusing on the smaller but more informative data, DR significantly improves the learning efficiency [1]. In this assignment, there are 612 samples (rows) of emotion data, each sample has 132 features (columns) and a specific target (emotion classification). Correlation Feature Selection (CFS) and Principal Component Analysis (PCA) are two DR approaches that we implemented and analyzed. The report will explain CFS and PCA and how we apply them to preprocess data before decision tree learning. It is also necessary to consider how performance of trained decision trees compare with different preprocessing techniques by analyzing the approaches. The 10-fold cross-validation is used in this assignment and the details of it is associated with the two DR approaches are demonstrated. Finally, the report ends up with answers to all additional questions and a conclusion of this assignment.

1 Technical Approaches and Implementations

1.1 Correlation Feature Selection (CFS)

CFS extracts a minimal subset of features that maximizes the correlation between the subset components and the classification targets (i.e. merit or CFS value) [1]. A higher merit comes from a higher correlation between features and targets and a lower inter-features correlation. The ideal CFS result is the smallest subsets which maximum $merit_{zc}$ in the formula 1.

$$merit_{S_k} = \frac{k\overline{r_{cf}}}{\sqrt{k + k*(k-1)*\overline{r_{ff}}}} = \frac{r_{cf_1} + r_{cf_2} + \cdots r_{cf_k}}{\sqrt{k + 2(r_{f_1f_2} + \cdots r_{f_if_j} + \cdots r_{f_kf_1})}} \tag{1}$$

k is the number of the selected components. r_{cf} is the feature-target correlation and r_ff is the inter-feature correlation. Numerous methods can be applied to compute the correlation. Pearson's product-moment correlation, which gives value between 1 and -1 to represent positive and negative correlation respectively, is the correlation formula that was used in this assignment. Since both negative and positive values should be considered to have the correlation, we used the **absolute** value to avoid elimination of the features with strong negative correlation to targets as shown in the formula 2.

$$r_{xy} = \left| \frac{\sum_{i=1}^{n}(x_i - \overline{x})\sum_{i=1}^{n}(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}} \right| \tag{2}$$

The implementation of CFS starts with an empty selected set S_k and a full set of initial features F. In each iteration, the feature leads to the highest merit is added to S_k and removed from F. The procedure keeps adding features to S_k until the merit of S_k starts to convergence, which ensures a minimal subset that maximum the merits of S_k is selected.

1.2 Principal Component Analysis (PCA)

PCA converts the original data by orthogonal transformation into a set of linearly independent principle components (PCs). By removing the dimension that has less variance, it performs the dimension reduction in a manner that keeps the approximately same information as the original dataset. This can be proved by Figure 1, which demonstrates the relation between the actual data and the reconstructed data from dimensional reduced PCs.

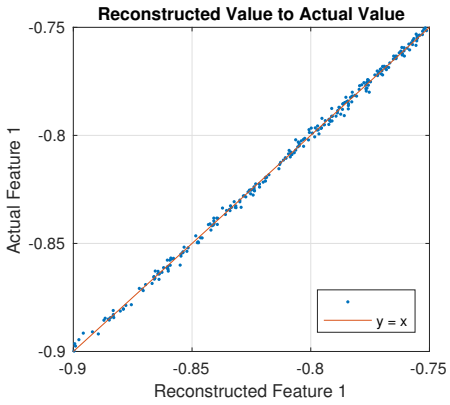

Figure 1: All points are closed to line y = x, which means the reconstructed value with 43 features can represent almost the same information as the actual data.

The process below details how we apply PCA to a dataset with m data records and n features:

1. Convert the original dataset into a $m \times n$ matrix X.
2. Apply Zero Mean Normalization to the matrix. For each value v_i in row r_i, subtract v_i with $\overline{v}$ of the row. (i.e. $v_i = v_i - \overline{v}$).
3. Calculate the covariance matrix C of X: $C = \frac{1}{m} X X^T$
4. Calculate the eigenvalue and eigenvector of C. Sort eigenvector in descending order based on eigenvalue. Store the first k rows which cover 95% of the variance of the original data in a new matrix C_k.
5. The dimensional reduced dataset Y is $Y = C_k X_{zeromean}$.

1.3 10-Fold Cross-validation

10-fold cross validation is again used to evaluate the performance of the trained decision trees. For each cross-validation iteration, 90% of the samples are used for training and the rest 10% are feed for validation. By separating training and validation set, it can accurately estimate the performance and discover potential overfitting.

To apply dimension reduction for the decision tree, we assigned eliminated feature with a value NaN to keep its original index unchanged. Meanwhile, CFS are applied six times in each iteration because decision tree of each emotion has distinct target set. Regarding the PCA, because it is not related to target set, we only used it once to the whole dataset to extract the PCs before training.

2 Feature Selection and Decision Tree Results

2.1 Feature Selection Results

Table 1 lists the selected features by CFS used in each decision tree.

labels	selected features (ordered by the importance)
1	106, 90, 107, 118, 22, 75
2	102, 88, 32, 2, 29, 74, 87
3	56, 96, 122, 59, 97, 55, 89
4	121, 49, 115, 50
5	131, 98, 121, 132
6	49, 87, 55, 60, 90, 75, 63

Table 1: feature selected by CFS for each emotion

Figures 2 and 3 demonstrate the distribution of classification results with respect to two primary features selected by CFS and PCA. Comparing the figures, the top two features selected by CFA separate the targets well while those extracted by PCA don't. One possible reason is that CFS is a supervised method that selects features closely related to the targets, but PCA is an unsupervised approach that preserves data covering most information (i.e. having higher variance).

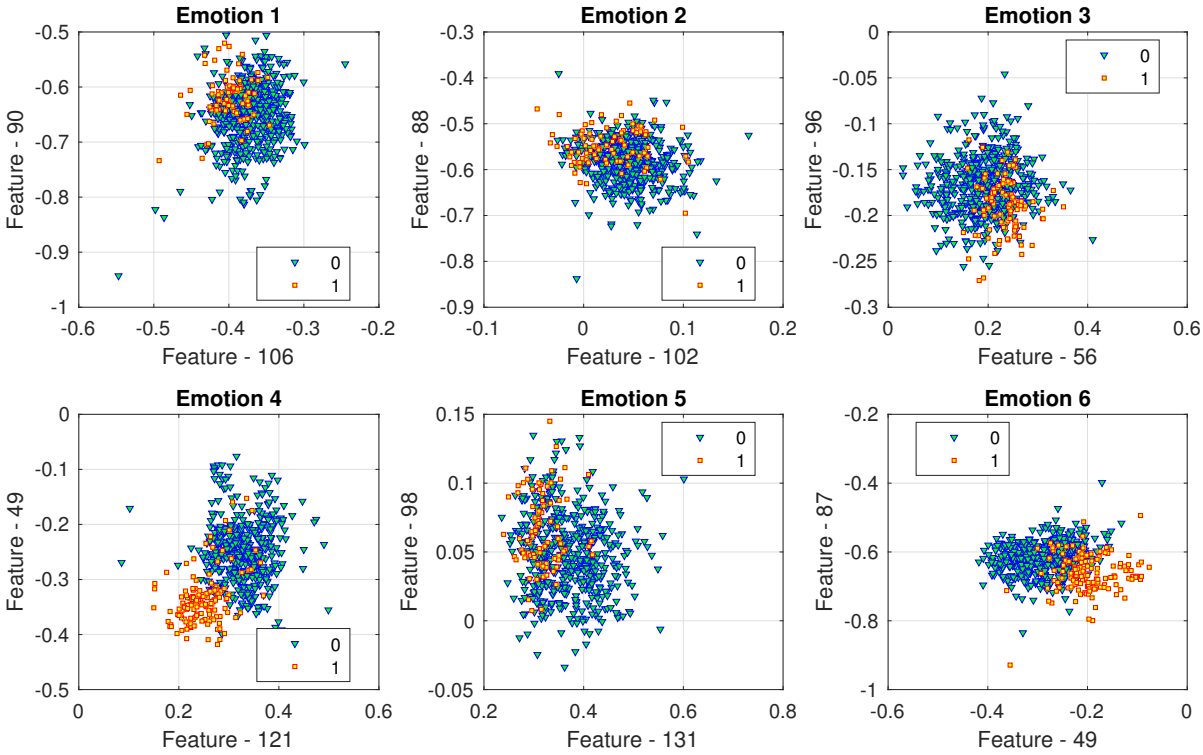

Figure 2: Top two important features of each emotion

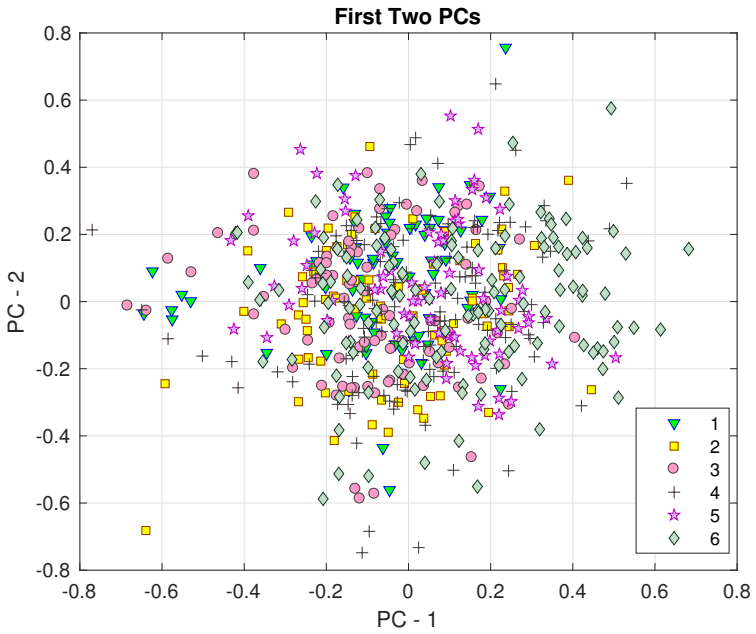

Figure 3: First Two Principle Components

2.2 Confusion Matrix for PCA and CFS

The below tables show the evaluation of the decision tree after 10-fold cross-validation using data processing separately by CFS and PCA. Confusion Matrices for every iteration of cross-validation are included in appendix B.

CFS						
label	1	2	3	4	5	6
1	**25**	16	10	8	8	7
2	15	**41**	6	6	8	8
3	14	10	**32**	17	10	7
4	10	5	14	**105**	5	4
5	14	15	13	6	**20**	16
6	1	9	10	9	9	**99**

PCA						
label	1	2	3	4	5	6
1	**23**	12	5	13	13	8
2	13	**22**	16	11	10	12
3	5	13	**27**	19	15	11
4	14	7	20	**77**	6	19
5	6	15	16	7	**31**	9
6	7	9	20	18	17	**66**

Table 2: Confusion Matrix of CFS and PCA

CFS				PCA			
label	recall	precision	F1	label	recall	precion	F1
1	0.3378	0.3165	0.3268	1	0.3108	0.3382	0.3239
2	0.4881	0.4271	0.4556	2	0.2619	0.2821	0.2716
3	0.3556	0.3765	0.3657	3	0.3000	0.2596	0.2784
4	0.7343	0.6954	0.7143	4	0.5385	0.5310	0.5347
5	0.2381	0.3333	0.2778	5	0.3690	0.3370	0.3523
6	0.7226	0.7021	0.7122	6	0.4818	0.5280	0.5038
AVG	**0.4794**	**0.4751**	**0.4754**	AVG	**0.3770**	**0.3793**	**0.3775**

Table 3: Evaluation of CFS and PCA

2.3 Performance Comparison

	no reduction	reduction by CFS	reduction by PCA
time	48.76	20.46	21.45
f1	0.5223	0.4754	0.3775

Table 4: Comparison Between Different Dimension Reduction Method

Table 4 shows that both CFS and PCA can significantly reduce the training time. However, prediction performance is disparaged after applying these DR methods. CFS processed data performs better than PCA. One reasonable explanation is that CFS is a supervised method while PCA is in an unsupervised approach.

3 Answers of Additional Questions

3.1 Explain why you needed to repeat CFS six times, but could apply PCA only once.

Because in CFS, we need to calculate the Pearson's product-moment correlation between features and targets. Since for each decision tree of a certain emotion, the targets are in 1-vs-all manner and distinct. Hence, we need to repeat CFS with different targets six times for each tree. However, PCA perform dimensional reduction only based on the variance of features. It is independent to to target labels and thus we can only apply it once to the whole dataset.

3.2 Why can't you directly infer what features are most informative after applying PCA?

Because not like CFS, PCA is unsupervised method (i.e. doesn't use labels), it can not reveal which feature is more related to the target. Moreover, the result of PCA are PCs generated by projecting all original features to these new components and are directly related to the variance rather than the correlation to the targets, thus we can not infer what features are most informative from these PCs.

3.3 How can you use PCA to analyze latent variables affecting the variance in your data?

The latent is a vector containing eigenvalues of the covariance matrix of the original data. The eigenvalues are positively proportional to the variance of the whole dataset. Since every PC is linear independent and has no correction between each other, the proportion of the latent of each PC to the sum of all latent represents the amount of the information each PC contains. If a PC has a latent Pl_k, a larger value of $\frac{pl_k}{\sum_{i=1}^{n} Pl_i}$ means a stronger capability to explain the original dataset.

4 Conclusion

In this report, we demonstrate that CFS and PCA are two DR methods which can apparently increase the efficiency of machine learning. However, through our evaluation, these DR approaches may lower the results of prediction. It is necessary that we consider the balance between efficiency and accuracy when applying DR to pre-processing the dataset.

References

[1] Hall, M.A., 2000. *Correlation-based feature selection of discrete and numeric class machine learning.*

Appendix

A Confusion Matrix for Each Iteration

Validation_confusion_matrix(:,:,N) indicates the Nth confusion matrix. Every confusion matrix is 6 X 6 in which validation_confusion_matrix(x,y,N) indicate that the number of samples which are with x as its actually value and y as the classification result of the decision tree.

A.1 CFS Results

Below shows confusion matrixes of decision trees for each cross validation. The data used to train decision trees are preprocessed using CFS algorithm.

```
cross_conf_cfs(:,:,1) =

     1     1     1     1     3     0
     1     4     1     1     0     1
     0     0     3     5     1     0
     0     0     3    10     0     1
     1     2     1     3     0     1
     0     2     0     1     0    11


cross_conf_cfs(:,:,2) =

     2     2     1     2     1     0
     2     2     1     1     3     0
     4     2     2     1     0     0
     2     0     0    12     1     0
     0     2     1     0     5     1
     1     1     0     0     0    11


cross_conf_cfs(:,:,3) =

     2     1     0     1     1     2
     1     4     2     1     0     0
     0     2     3     3     0     1
     2     1     2     9     0     0
     1     1     4     0     2     0
     0     1     1     0     0    12


cross_conf_cfs(:,:,4) =

     6     1     1     0     0     0
     0     6     0     1     0     2
```

```
     2     1     4     0     0     2
     2     0     2     9     1     0
     2     2     0     1     3     1
     0     0     1     3     0    10
```

```
cross_conf_cfs(:,:,5) =
```

```
     3     2     2     0     0     0
     0     4     0     0     1     3
     1     0     7     1     0     0
     0     2     1    12     0     0
     0     0     2     0     3     3
     0     1     1     0     0    12
```

```
cross_conf_cfs(:,:,6) =
```

```
     1     2     1     1     1     1
     3     4     1     0     0     0
     1     1     3     2     2     0
     2     1     2     7     1     1
     2     2     0     1     1     2
     0     0     1     0     3     9
```

```
cross_conf_cfs(:,:,7) =
```

```
     2     2     1     1     0     2
     2     6     0     0     0     1
     2     0     3     2     1     1
     1     1     0    11     1     0
     3     1     1     0     1     3
     0     0     1     2     0    11
```

```
cross_conf_cfs(:,:,8) =
```

```
     3     1     1     0     1     1
     3     2     0     1     2     0
     2     2     0     0     4     1
     0     0     4    10     0     0
     1     0     1     1     1     4
     0     1     2     2     1     8
```

```
cross_conf_cfs(:,:,9) =
```

```
     2     1     2     2     1     0
     2     6     0     0     0     1
     2     1     4     1     1     0
     0     0     0    13     0     2
```

2	4	2	0	1	0
0	2	1	1	2	7

cross_conf_cfs(:,:,10) =

3	3	0	0	0	1
1	3	1	1	2	0
0	1	3	2	1	2
1	0	0	12	1	0
2	1	1	0	3	1
0	1	2	0	3	8

A.2 PCA Results

Below shows confusion matrixes of decision trees for each cross validation. The data used to train decision trees are preprocessed using PCA algorithm.

cross_conf_pca(:,:,1) =

1	1	1	1	1	2
1	2	2	2	1	0
1	1	5	2	0	0
0	2	2	9	0	1
0	1	2	4	1	0
2	1	1	2	3	5

cross_conf_pca(:,:,2) =

2	2	1	2	1	0
4	1	2	0	2	0
0	3	1	1	2	2
3	1	1	9	1	0
0	1	1	0	5	2
0	0	2	1	5	5

cross_conf_pca(:,:,3) =

3	3	0	1	0	0
0	4	0	2	0	2
0	0	3	2	2	2
1	1	1	9	1	1
0	3	2	0	3	0
0	0	2	1	2	9

cross_conf_pca(:,:,4) =

1	0	0	3	3	1
2	1	2	2	0	2
0	3	3	2	1	0

```
    1     1     1     6     1     4
    1     1     1     2     3     1
    0     3     2     2     1     6
```

```
cross_conf_pca(:,:,5) =
```

```
    3     0     0     1     2     1
    0     2     1     2     1     2
    0     1     4     1     1     2
    0     1     2    10     0     2
    0     1     2     0     5     0
    1     0     1     4     1     7
```

```
cross_conf_pca(:,:,6) =
```

```
    2     2     1     0     1     1
    1     4     1     0     0     2
    0     2     2     2     1     2
    1     1     2     9     0     1
    0     2     3     1     2     0
    1     0     1     2     0     9
```

```
cross_conf_pca(:,:,7) =
```

```
    3     1     0     2     0     2
    0     3     1     0     3     2
    1     0     2     1     5     0
    3     0     3     7     0     1
    1     1     0     0     4     3
    2     1     1     1     0     9
```

```
cross_conf_pca(:,:,8) =
```

```
    4     0     0     0     3     0
    2     1     3     1     1     0
    0     1     2     2     2     2
    2     0     4     3     2     3
    1     4     2     0     1     0
    0     2     1     2     3     6
```

```
cross_conf_pca(:,:,9) =
```

```
    2     2     0     3     1     0
    1     1     4     1     2     0
    2     0     4     2     0     1
    2     0     3     7     0     3
    2     0     2     0     4     1
```

```
     1     2     6     1     1     2


cross_conf_pca(:,:,10) =

     2     1     2     0     1     1
     2     3     0     1     0     2
     1     2     1     4     1     0
     1     0     1     8     1     3
     1     1     1     0     3     2
     0     0     3     2     1     8
```