Diego Rincon
2/29/2016

Homework 1

1. Replace Missing Values (min, max, or a specific value)

Given a data set with missing values we can fill out those values using the rest of the data. The usual options are using the minimum value, maximum value or average.

2. Missing Values Ratio

Given a data set with missing values we can distinguish those columns (attributes) that will carry useful information from those that will not. The Missing Values Ratio technique computes the number of missing values for each column and if this number is above a certain threshold, the column will be removed thus ignoring those columns that have too little information.

3. Low Variance Filter

Data columns whose values have very low variance don't carry significant information about the dataset. This often means that the column can be ignored. The Low Variance Filter technique is used to remove those columns that have low variance.

4. High Correlation Filter

This technique uses the fact that attributes that are very much alike will likely carry the same information. If two attributes have a correlation coefficient above a certain threshold, these attributes are reduced to only one.

5. Random Forests Ensemble Trees

This technique is based on the use of decision tree ensembles to reduce the dimension of a dataset. The way this works is by generating a large set of trees against a specific attribute and look closely at each attribute's usage to find the best subset.

6. Principle Component Analysis

This technique is used to transform the original n coordinates of a data set into a new set of coordinates called the principal components. The way to do this is by calculating the eigenvector matrix of the covariance matrix of the dataset. The first $m < n$ components will reduce the dimension of the dataset by (n-m). The key aspect is that the first principal component will have the largest possible variance; so by considering only the first m we are retaining the components that have the most variance.

7. Singular Value Decomposition

SVD is a form of matrix analysis that, as PCA, reduces the dimension of the dataset. The technique depends on building three matrices (U, Σ, and V) whose columns represent *concepts* that are hidden in the original matrix.

8. Backward Feature Elimination

This is another feature reduction technique where a given classification algorithm is trained on all *n* input features. Then we iterate over all features, removing one at a time and looking at the error rate. We then remove the feature that throws the biggest error – giving us a dataset with *n*-1 features.

9. Forward Feature Selection

This is the reverse process to Backward Feature Elimination; here we do not train the classifier with all the features but rather add one feature at a time, keeping track of their increase in performance. We remove the one(s) that don't produce a significant increase.

I used the following datasets – all labeled and binary-classifiable:
Retrieved from: http://archive.ics.uci.edu/ml/
Credit Card Default Set (# instances: 30000, # attributes: 24)
Ionosphere (351, 34)
Horse Colic (368, 27) – With Missing Values
Breast cancer (569, 32)
For the validation phase I used 10-fold cross-validation, using the Adaboost classifier with Decision Trees using RapidMiner software.

Horse Colic
Pre-Processing Method: Replace Missing Values
Result:

Replacement	**Accuracy**	Confusion Matrix
Average	47.8% +- 13.68%	True: 2 1 2: 115 152 1: 4 28
Maximum	45.45% +- 10.65%	True: 2 1 2: 113 157 1: 6 23
Minimum	46.49% +- 9.38%	True: 2 1 2: 115 156 1: 4 24

Credit Card Default
Pre-Processing Method: PCA

Replacement	**Accuracy**	Notes
Fixed: 20 attributes	81.95% +/- 0.40%	
Variance: Threshold: 0.95	77.88% +/- 0.02%	
No PCA	79.29% +/- 0.30%	

Ionosphere
Pre-Processing method: SVD

Replacement	**Accuracy**	Notes

Fixed: 20 attributes	81.95% +/- 0.40%	
Variance: Threshold: 0.95	77.88% +/- 0.02%	
No PCA	79.29% +/- 0.30%	

Ionosphere
Pre-Processing method: Backward Feature Elimination

Replacement	**Accuracy**	Notes
Max # of eliminations: 10. Stopping with decrease	93.17% +/- 3.17%	
No BFE	92.06% +/- 7.08%	

Breastcancer
Pre-Processing method: Forward Feature Selection

Replacement	**Accuracy**	Notes
Max # of attributes: 10. Stopping without increase	93.14% +/- 2.16%	
No FFS	94.91% +/- 2.28%	

2.

2.1 PCA

This technique is used to transform the original *n* coordinates of a data set into a new set of orthogonal coordinates called the principal components. The way to do this is by calculating the eigenvector matrix of the covariance matrix of the dataset. The first $m < n$ components will reduce the dimension of the dataset by (n-m). The key aspect is that the first principal component will have the largest possible variance; so by considering only the first *m* we are retaining the components that have the most variance.

Besides being applied to data reduction, PCA is also used for feature construction and data visualization. It is used in many different areas such as signal processing, mechanical engineering, psychometrics, meteorological science and, of course, linear algebra.

2.2

$$M = \begin{matrix} 2 & 3 & 1 & 1 \\ 4 & 4 & 1 & 1 \\ 0 & 1 & 4 & 4 \\ 0 & 1 & 2 & 2 \end{matrix}$$

2.3

Applying PCA:

1 Compute $M^TM =$

$$\begin{matrix} 20 & 22 & 6 & 6 \\ 22 & 27 & 13 & 13 \\ 6 & 13 & 22 & 22 \\ 6 & 13 & 22 & 22 \end{matrix}$$

2 Compute Eigenvalues and Eigenvectors

Eigenvalues = 0, 0.3714, 26, 64.6286

Eigenvectors =

0.0000	0.7073	-0.5685	0.4202
-0.0000	-0.6901	-0.4264	0.5847
-0.7071	0.1084	0.4975	0.4907
0.7071	0.1084	0.4975	0.4907

3 Use the eigenvectors from the largest eigenvalues to reduce the dimension of the matrix

0.4202	-0.5685	0.7073
0.5847	-0.4264	-0.6901
0.4907	0.4975	0.1084
0.4907	0.4975	0.1084

Result:

3.5759	-1.4212	-0.4389
5.0010	-2.9846	0.2856
4.5103	3.5536	0.1771
2.5475	1.5636	-0.2565

2.4

SVD

SVD is a form of matrix analysis that, as PCA, reduces the dimension of the dataset. The technique depends on building three matrices (U, Σ, and V) whose columns represent *concepts* that are hidden in the original matrix.

Decomposing M using SVD:

U =

-0.4448	-0.2787	0.7206	-0.4529
-0.6221	-0.5854	-0.4681	0.2265
-0.5610	0.6969	-0.2902	-0.3397
-0.3169	0.3066	0.4211	0.7926

∑ =

8.0392	0	0	0
0	5.0990	0	0
0	0	0.6094	0
0	0	0	0.0000

V =

-0.4202	-0.5685	-0.7073	0
-0.5847	-0.4264	0.6901	0.0000
-0.4907	0.4975	-0.1084	0.7071
-0.4907	0.4975	-0.1084	-0.7071

2.5

The *r* (rank of M) columns of U, ∑, and V represent concepts hidden in the original matrix. From the above, we can see that there are really only two clear concepts—the magnitude of the ∑'s values give us indication of the presence of a "concept". The way to reduce the dimensions of the matrix is to replace the smallest values in ∑ by 0 and eliminating the corresponding rows of U and V.

2.6

Query = [5,0,0,1]

2.7

In order to find similar users, we can map the query vector into the concept space represented by the modified version of V

U =

-0.4448	-0.2787
-0.6221	-0.5854
-0.5610	0.6969
-0.3169	0.3066

$\sum$ =

8.0392	0
0	5.0990

V =

-0.4202	-0.5685	-0.7073	0
-0.5847	-0.4264	0.6901	

Thus if we multiply [5,0,0,1] by V, we get [-2.5917, -2.3450] which maps the query with the concept space. This representation, however, is different from X's representation of the concept space. To get that representation back, we can multiply the resulting vector by V^T. This gives us: [2.4222, 2.5153, 0.1051, 0.1051] which suggests that X's "preferences" will probably be more aligned with the first two columns.