Weka – k-Means & Attribute Transformation (2)

Lab7 (in-class): 8 DIC 2016 -10:15-12:00 (TURING)

ACKNOWLEDGEMENTS: INFORMATION, EXAMPLES AND TASKS IN THIS LAB COME FROM SEVERAL WEB SOURCES.

Learning objectives

In this assignment you are going to:

- k-Means
- Attribute Transformation: scaling

Task 1: Warming-up – Scaling Numerical Attribute Values [maxtime: 10 min]

Data preprocessing plays a crucial role. Last lab we used two filters to binarize data and to transform data from string to word vectors.

In this warming-up task we will quicky go through transformation to rescale data.
Two methods are usually well known for *rescaling* data (**see the Math course**):

- *normalization*, which scales all numeric variables in the range [0,1];
- *standardization* tha will transform the data to have zero mean and unit variance

In this task you will learn:
-- How to normalize your dataset to the range 0 to 1.
-- How to standardize your data to have a mean of 0 and a standard deviation of 1.

Just for the sake of completeness, remember that both these techniques have their drawbacks. One of the drawbacks of normalization is when the data contains outliers (anomalies), because this will aggregate most of the data in a very small range and only outliers will lay on the boundaries. Therefore, one disadvantage of normalization over standardization is that it loses some information in the data, especially about outliers.

Normalization in weka: diabetes dataset

Normalization of the data: This step is very important when dealing with parameters of different units and scales. Therefore, all parameters should have the same scale for a fair comparison between them.
Essentially, data normalization is the process of rescaling one or more attributes to the range of 0 to 1. This means that the largest value for each attribute is 1 and the smallest value is 0.
Normalization is a good technique to use when you do not know the distribution of your data or when you know the distribution is not Gaussian/normal (a bell curve). (Generally speaking, normalization is very useful when your data has varying scales and the algorithm you are using does not make assumptions about the distribution of your data, such as k-nearest neighbors and artificial neural networks.)

You can normalize all of the attributes in your dataset with Weka by choosing the Normalize filter and applying it to your dataset.

Follow these steps to normalize your dataset.
Open the diabetes dataset:
< http://stp.lingfil.uu.se/~santinim/ml/2016/Datasets/diabetes.arff >

Analyse the dataset (features, class, data types, etc.)
In the Preprocess tab, click the "Choose" button to select a Filter and select *unsupervised.attribute.Normalize*. Click the "Apply" button to normalize your dataset.

The default value of the scaling factor is 1. But with the scale and translation parameters one can change that, e.g., with scale = 2.0 and translation = -1.0 you get values in the range [-1,+1] (read the synopsis). (*You can use other scales such as -1 to 1 when using support vector machines and adaboost*).

Action1: Review the details of each attribute in the "Selected attribute" window will give you confidence that the filter was successful and that each attribute was rescaled to the range of 0 to 1 (or the scale you choose). **This transformation might improve the performance of your classifier in some situations**.
Action2: Run a classifier of your choice on the diabetes dataset with original and normalized data. Can you notice any difference in the performance?

Standardization in weka: diabetes dataset

As stated above, data standardization is the process of rescaling one or more attributes so that they have a mean value of 0 and a standard deviation of 1.
Standardization assumes that your data has a Gaussian (bell curve) distribution. This does not strictly have to be true, but the technique is more effective if your attribute distribution is Gaussian. (Generally speaking, standardization is useful when your data has varying scales and the algorithm you are using does make assumptions about your data having a Gaussian distribution, such as linear regression, logistic regression and linear discriminant analysis.)

You can standardize all of the attributes in your dataset with Weka by choosing the Standardize filter and applying it your dataset.
Follow these steps to normalize your dataset. Use the diabetes dataset.
In the Preprocess tab, click the "Choose" button to select a Filter and select unsupervised.attribute.Standardize. Click the "Apply" button to normalize your dataset.
Click the "Save" button and type a filename to save the standardized copy of your dataset.

Action3: Review the details of each attribute in the "Selected attribute" window will give you confidence that the filter was successful and that each attribute has a mean of 0 and a standard deviation of 1. **This transformation might improve the performance of your classifier in some situations**.
Action4: Run a classifier of your choice on the diabetes dataset with original, normalized and stardardized data. Can you notice any difference in performance?

Task 2: Building the Clustering (Simple k-means) Model [maxtime: 15 min]

In this exercise, we will create the simple k-means models for predicting the thyroid disease outcome.

Data preprocessing

Download the sick dataset from here:
< http://stp.lingfil.uu.se/~santinim/ml/2016/Datasets/sick.arff >

The dataset contains 30 attributes of patient data describing patient information regarding the thyroid diagnoses obtained from the Garvan Institute, consisting of 9172 records from 1984 to early 1987.

It is important to properly preprocess the data. Some of the key factors that need to be considered are total number of instances, number of attributes, number of continuous and/or discrete attributes, number of missing values, etc.

In the starting interface of Weka, click on the button ***Explorer***. In the ***Preprocess*** tab, click on the button ***Open File*** and upload the dataset.

Information about the selected attribute is given in the ***Selected attribute*** frame in which a histogram depicts the attribute distribution.

One can see that the value of the currently selected descriptor T4U shows the distribution of the attribute values in the dataset. Take a note of the number of *missing*, *unique* and *distinct values*. **Do you want to keep or remove this attribute?** (let's remove it for now).

Select the last attribute "class" in the ***Attributes*** frame. One can read from the ***Selected attribute*** frame that there are 3541 negative and 321 sick class examples in the dataset. Negative compounds are depicted by the blue color whereas "sick" compounds are depicted by the red color in the histogram. Note the ratio of the number of represented class values for each class. **Does it seem balanced?**

Note that the attribute number 28 named "TBG" has 100% missing values. This attribute together with the related attribute 27 should be removed by adding the checkmark in from of the attribute name and clicking on the Remove button below.

Consider attribute 29 - the referral source which seems irrelevant, but that may depend on nature of the disease (keep it for now).

Click on **Visualize** all button on the lower right – to look at class distribution across the entire set of attributes.

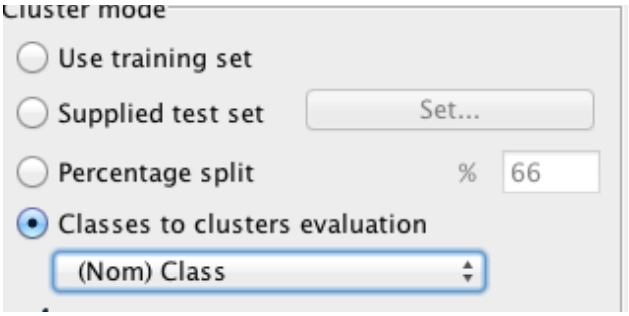
Examine each one of the variables. *Did you notice anything? Are there any variables you think should be removed or manipulated in any way? Are there any duplicates? etc*.

Now let's do some clustering, ie we want to find "spontaneous" grouping.

In the ***Clustering*** frame, click ***Choose***, than select the ***Simple K-Means*** method. Use **Training set** as Cluster mode. Click on the ***Start*** button to build the simple k-means model.

Q1: how many clusters have been created? What is the default value for the number of clusters? What happens if you specify a higher number?

Clustering algorithms ignore the class (even if stated in the dataset) when building the model. If stated in the dataset, the class label can be used for evaluation.
Suppose that (thanks to your domain expertise) you know the appropriate number clusters in advance. Choose the following testing option:



Rerun k-Means after you have set the number of clusters to 2 (see parameters' window).

Q2: what happens now? what's the error rate?

Task 3: Analyzing the Output [maxtime: 20 min]

For this task we will use the customer dataset
< http://stp.lingfil.uu.se/~santinim/ml/2016/Datasets/customers.arff >

Purpose: an international online catalog company wishes to group its customers on common features. Based on the outcome of the grouping, they will target marketing and advertising campaigns to the different groups. For example, suppose the advertising is for a special sale on children's clothes. We will target the advertising only to the persons with young children.
The clustering that you will perform in this task is as follows. The **first** group of people has young children and a high school degree; the **second** group does not have children but has a high school degree; the **third** group has both children and a college degree; the **fourth** group has higher income and at least a college degree; the **fifth** group has children and higher degree.
Different clustering would have been found by examining either age or marital status.

In the Preprocess tab, upload the dataset. In this task we use only a trimmed dataset, with just a few records.

In the Cluster tab, choose SimpleKMeans (remember some implementations of k-means only use numerical data; in case your dataset is not numerical, you have to apply a filter).
Display the parameter window, set numClusters to 5 (the default is 2). Leave the value of seeds as is. *Remember that this seed value is used in generating a random number, which is used for making the initial assignment of instances to clusters. Generally speaking, k-Means is quite sensitive to how clusters are initially assigned. Thus, in practical terms, this means that it is often necessary to try different values and evaluate the results*.

Choose **Classes to cluster evaluation** as "Cluster mode". Select "marital status" in the pull-down. It means that you will compare how well the chosen clusters match up with the pre-assigned class (marital status) in the data.

Run the k-means and read the output.
Q: Interpret the following sections:

1. Run information
2. How many iterations?
3. Cluster centroids
4. Cluster instances
5. Confusion matrix
6. Error rate.

Task 4: Getting familiar with the visualization of Clustering Results [maxtime: 15 min]

Another way of representing clustering results is through visualization.
Right-click on the entry related to the customer clustering. Select "Visualize cluster assignments" in the pull-down window. This brings up the weka clusterer visualizer window.

On the 'Weka Clusterer Visualize' window, beneath the X-axis selector there is a dropdown list, 'Colour', for choosing the color scheme. This allows you to choose the color of points based on the attribute selected.

Below the plot area, there is a legend that describes what values the colors correspond to. In your example, seven different colors represent seven numbers (number of children). For better visibility you should change the color of label '3'. Left-click on '3' in the 'Class colour' box and select lighter color from the color palette.

To the right of the plot area there are series of horizontal strips. Each strip represents an attribute, and the dots within it show the distribution values of the attribute. You can choose what axes are used in the main graph by clicking on these strips (left-click changes X-axis, right click changes Y-axis). Set X - axis to 'Cluster' attribute, Y - axis to 'Age'. Select 'Children' as the color dimension. You can see the result in a visual rendering of the relationship within each cluster. For instance, you can note that 'cluster 0' represents a group of people of age 25 and 35, who have 3 children, 'cluster 1' represents a group of people of age 30 and 45 who do not have children, 'cluster 2' represents 50 and 60 year old people with no children, 'cluster 3' represents 25 year old

married people with one child, and 'cluster 4' represents 20 and 40 year old people without children.

The initially correctly clustered instances are represented by crosses, incorrectly clustered once represented as squares. By changing the color dimension to other attributes, you can see their distribution within each of the clusters.

You may want to save the resulting data set, which included each instance along with its assigned cluster. To do so, click 'Save' button in the visualization window and save the result as the file "customers_kmeans.arff".

Task 5: Clustering the junk dataset [maxtime: 15 min]

Apply k-Means to the junk dataset and compare the performance of this unsupervised machine learning algorithm to any supervised algorithm of your choice. In the clustering algorithm try and change the distance function and see what happens

Appendix

How to read the output, see Task 3

'**Run Information'** gives you the following information:
- the clustering scheme used: SimpleKMeans with 5 clusters
- the relation name "customers"
- number of instances in the relation – 9
- number of attributes in the relation – 6
- list of attributes used in clustering
- the ignored cluster 'marital_status' is an attribute the clustering is performed on.

'**Sum of squared errors**':
we calculate the error of each data point, ie its Eucledian distance to the closest centroid, and then compute the total sum of the squared errors. Given two different sets of clusters that are produced by two different runs of k-means, we prefer the one with the smallest squared error since this means that the centroids of this clustering are better representation of the points in the cluster.

The clustering model shows the centroid of each cluster and statistics on the number and percentage of instances assigned to different clusters. Cluster centroids are the **mean vectors** for each cluster; so, each dimension value and the centroid represents the mean value for that dimension in the cluster. Thus, centroids can be used to characterize the clusters. WEKA generated clusters are:

1. Cluster 0 shows that this is a segment of cases representing 25 and 35 year old, either single or divorced, people with income $22,500 in average, who have 3 children.
2. In cluster 1 there are 30 and 45 year old married people who do not have children.
3. In cluster 2 there are 50 and 60 year old married and divorced people with higher income college degree and no children.
4. Cluster 3 represents 25 year old married people with one child lower income and high school degree.
5. Cluster 4 represents 20 and 40 year old single and divorced people with lower income, high school degree and no children.

Sum of errors within the clusters is recalculated.

'**Cluster Instances'** section shows the number of instances in each new cluster.
For example, cluster 3 has 1 instance: people of age 25 who have one child. Cluster 4 has 2 instances: people of age 30 in average (including 20 and 40 y.o.), whose average income is $25,000, with high school education and no children. 'Classes to Clusters" represents class ('marital-status') assigned to clusters. The last line displays the you have 5 number incorrectly classified instances, which is 55.5 %.

---The end---