Data Mining
Final project report

Implementation of a single classification algorithm and several algorithms for feature reduction.

Przemysław Piórkowski, Michał Debski, Damian Hermanowski

June, 2015

Contents

Chapter 1

Introduction

The project objective is to implement single classification algorithm and several algorithms for feature reduction. In the addition, developed algorithms have to be evaluated against chosen data set and classification method.

Basing on the document provided by project's Supervisor and the initial research the Team decided to implement three types of the feature reduction algorithms: Document Frequency Thresholding (DFT), Mutual Information (MI) and χ^2 statistic (CHI). As a classification algorithm the k-nearest neighbors (kNN) method was chosen. The Team decided to use the C++ programming language for implementation purposes.

The source data set: SPAM E-mail Database, consists of 4601 instances of e-mails. Each e-mail is considered either as a spam or not. Classification is based on 57 continuous attributes denoting particular words and characters frequency as well as lengths of uninterrupted sequences of capital letters in a message. The data set does not have any missing attribute values. Attributes were normalized with knowledge of attributes values meaning, for example, 54 attributes denotes words and characters frequency - each of these values were divided by 100. For continuous attributes min-max normalization was applied.

Chapter 2

Project Structure

Source code of project is simply organized. Project root includes *main.cpp* file which contains *main* function - application starting point. Alongside to *main.cpp* file there are also sources of some utility classes and structures.

Data folder holds implementation of classes providing abstraction for datasets loading, reading and manipulation. Support for creating datasets for cross validation is also implemented in this directory.

CHI, *DFT* and *MI* folders holds implementation of feature reduction algorithms for respectively CHI reduction, Document Frequency Thresholding and Mutual Information. More detailed information about algorithms implementation are described in next chapter.

KNNClassifier folder holds implementation of KNN classifier algorithm. It is standard KNN implementation. To simplify comparison of different feature reduction algorithms this implementation takes as parameter array of doubles values. Each value represents weight of corresponding feature. This design implies that feature reduction is preformed simply by set reduced feature weight to 0.

Chapter 3

Algorithms implementation

DFT thresholding is the simplest technique for attribute reduction. Document frequency is the number of documents in which a term occurs. Our implementation counts occurences of not null values of each attribute. It utilizes the training set for this step. After that a vector of frequencies is created. An attribute which frequency is below given threshold is filtered out for further classification steps. The basic assumption is that rare terms are either non-informative for category prediction, or not influential in global performance.

MI tries to measure the influence of existence of particular attribute/term on a given class value. If one considers the two way contingency table of a term t and a category c, where A is the number of times t and c co-occur, B is the number of time the t occurs without c, C is number of times c occurs without t, and N is the total number of documents, then the mutual information criterion between t and c is estimated using formula:

$$I(t,c) \approx \log \frac{A \times N}{(A+C) \times (A+B)}$$

I(t ,c) equal to zero means that attribute/term t and category c are independent. The goodness of aterm/attribute is scored in two ways: as a average of MIs for a given attribute (MI_AVG mode) and as a maximal MI for a given attribute (MI_MAX mode). The outcome of the algorithm is a reduction vector of attributes scores. An attribute with a score below a given threshold is filtered out.

$$I_{avg}(t) = \sum_{i=1}^{m} P_r(c_i) I(t, c_i)$$

$$I_{max}(t) = \max_{i=1}^{m} \{I(t, c_i)\}$$

The χ^2 statistic measures the lack of independence between an attribute t and a class c. Similarly to MI algorithm utilizes the two way contingency table with an additional value D, which equals to case when neither c nor t occurs. The term-goodness is calculated with a given formula:

$$\chi^2(t, c) = \frac{N \times (AD - CB)^2}{(A + C) \times (B + D) \times (A + B) \times (C + D)}$$

Similarly to the MI method, the CHI statistic between the given attribute and classes may be computed as two scores: average value (CHI_AVG mode) and maximum value (CHI_MAX mode). The outcome of the algorithm is a vector of attributes scores. An attribute with a score below a given threshold is filtered out.

$$\chi^2_{avg}(t) = \sum_{i=1}^{m} P_r(c_i) \chi^2(t, c_i)$$

$$\chi^2_{max}(t) = \max_{i=1}^{m} \{\chi^2(t, c_i)\}$$

The project implements three feature reduction algorithms within 3 classes:

- CHIReduction
- DFTReduction
- MIReduction

The kNN algorithm is implemented in the **KNNClassifier** class. There were developed additional classes for data loading and data interpretation: **DataLoader**, **Data**. **DataAdapter** class is used to iterate over data in function defined manner. There is the **Logger** class for debugging and echoing purposes. There is the class supporting multithreading for parallelization capabilities: **ParallelExecutor** with a support of *AtomicIfPossible* and *AtomicInternal* classes.

The developed project application can be run with given parameters:

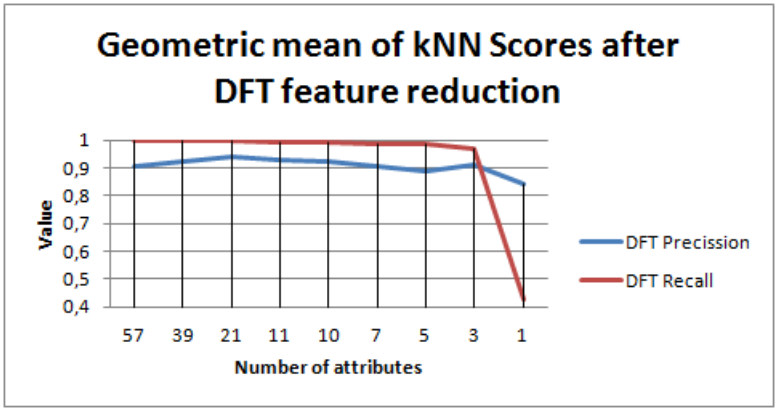
- `-ftrain trainFileName` - csv file name containing train dataset
- `-ftest testFileName` - csv file name containing test dataset

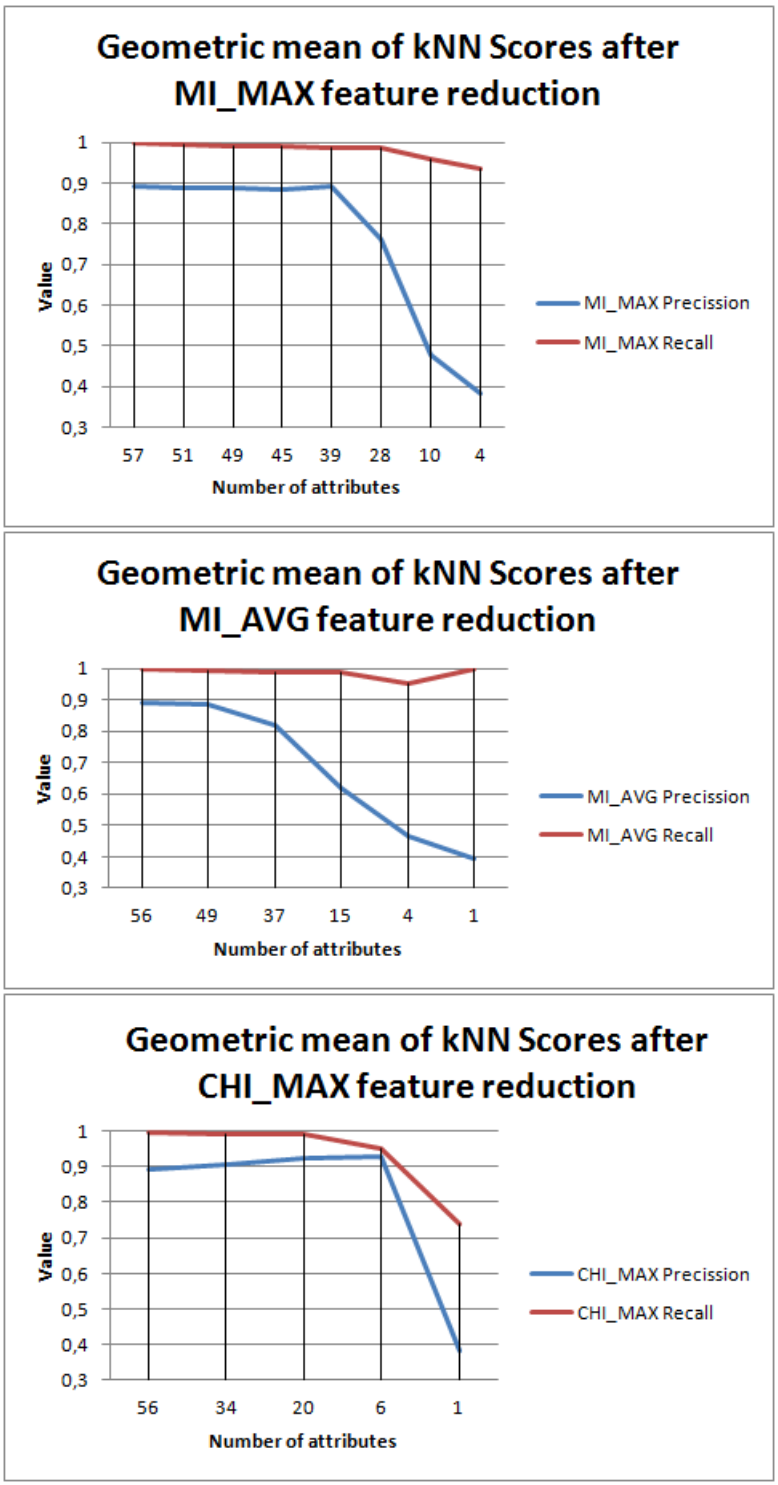
- `-h [TRUE,FALSE]` - switch for reading headers of a source file
- `-cname className` - class attribute header
- `-k value`
- `-CV value` - number of iterations/folds of cross valdiation
- `-DFT dftThreshold` - treshold for DFT algorithm
- `-MI_MAX` - switches on MAX score calculation for MI algorithm
- `-MI_AVG` - switches on AVG score calculation for MI algorithm
- `-MI miThreshold` - treshold for MI algorithm
- `-l verbosityLevel` - value [1,2,3,4,5]
- `-CHI_MAX`- switches on MAX score calculation for CHI algorithm
- `-CHI_AVG`- switches on AVG score calculation for CHI algorithm
- `-CHI chiThreshold` - treshold for CHI algorithm

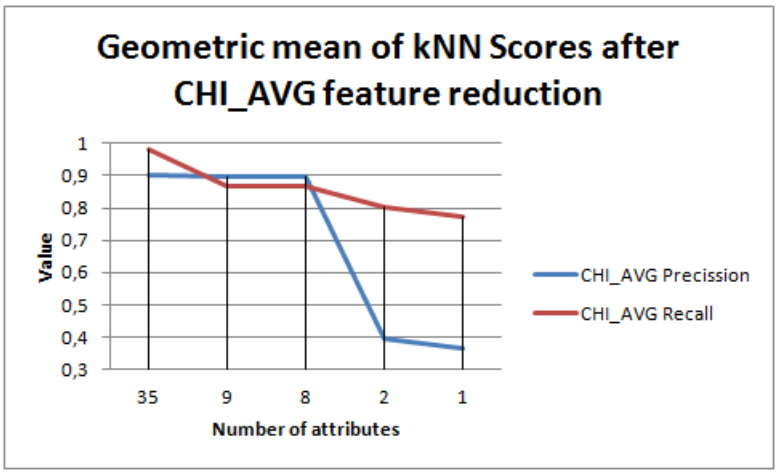
Chapter 4

Experiments

In order to validate implemented algorithms against the data set the *7-fold Cross-validation* method was used (`-CV 7`). kNN implementation has one control parameter - the number of neighbors voting. For evaluation purposes this k-value was set to 5. The influence of each feature reduction algorithm on the kNN classification method was measured with precission and recall values. Scores were calculated as a geometric mean of values obtained from each round of the cross validation. These quality measures were compared with the results of classification without feature reduction. Number of attributes was controlled with picked threshold values.




Geometric mean of kNN Scores after MI_MAX feature reduction
Value
1
0,9
0,8
0,7
0,6
0,5
0,4
0,3
57 51 49 45 39 28 10 4
Number of attributes
MI_MAX Precission
MI_MAX Recall
Geometric mean of kNN Scores after MI_AVG feature reduction
Value
1
0,9
0,8
0,7
0,6
0,5
0,4
0,3
56 49 37 15 4 1
Number of attributes
MI_AVG Precission
MI_AVG Recall
Geometric mean of kNN Scores after CHI_MAX feature reduction
Value
1
0,9
0,8
0,7
0,6
0,5
0,4
0,3
56 34 20 6 1
Number of attributes
CHI_MAX Precission
CHI_MAX Recall


Geometric mean of kNN Scores after CHI_AVG feature reduction
Value
1
0,9
0,8
0,7
0,6
0,5
0,4
0,3
35
9
8
2
1
Number of attributes
CHI_AVG Precission
CHI_AVG Recall

Chapter 5

Conclusions

The results are interesting. The application of DFT, CHI_MAX and MI_MAX on the given data set gave slight precission gain when less than a half of attributes were cut. The DFT method achieved high precission and recall scores even with 3 attributes left - it is a great result for a such simple algorithm. CHI_MAX and CHI_AVG methods marked comparable - good overall results. Mutual Inforamtion (MI) methods showed worse results for aggresive feature reduction.