

Text Categorization for Multi-label Documents and many Categories

I. Sandu Popa⁽¹⁾, K. Zeitouni⁽¹⁾, G. Gardarin⁽¹⁾, D. Nakache⁽²⁾, E. Métais⁽²⁾

(1) PRiSM Laboratory, 45 avenue; des Etats-Unis, F-78035 Versailles, France

{*Iulian.Sandu-popa, Karine.Zeitouni, Georges.Gardarin*}@prism.uvsq.fr

(2) CEDRIC Laboratory, 292 Rue Saint Martin, 75141 Paris Cedex 03, France

datamining@wanadoo.fr, elsa@cnam.fr

Abstract

In this paper, we propose a new classification method that addresses classification in multiple categories of textual documents. We call it Matrix Regression (MR) due to its resemblance to regression in a high dimensional space. Experiences on a medical corpus of hospital records to be classified by ICD (International Classification of Diseases) code demonstrate the validity of the MR approach. We compared MR with three frequently used algorithms in text categorization that are k-Nearest Neighbors, Centroides and Support Vector Machine. The experimental results show that our method outperforms them in both precision and time of classification.

1. Introduction

In France and many countries, it is a legal obligation to supply ICD codes whenever a patient treatment reaches its end. Some hospitals employ whole-time members of the medical profession to fulfill this task. The purpose is triple: (1) basis for appropriation assignment, considering these codes represent the unit activity, (2) authorizations for nationwide and international epidemiological studies, thanks to a systematic and standardized coding of pathologies, and (3) factor for calculating indicators of the quality of treatments, such as the number of nosocomial infections, which justifies that the quality of coding itself has been proposed as a quality indicator. In the last years, machine learning approaches have demonstrated some success for the construction of automatic document categorizers. Machine learning addresses the question of how to build computer programs that improve human performance at some task through experience [1]. In the field of machine learning, two kinds of methods have been developed: supervised learning and unsupervised learning. Supervised learning algorithms operate by learning the objective function from a set of training examples and then applying the function to the target set. Unsupervised learning operates by trying to find relations, associations, or intrinsic affinities on the data.

In this paper, we propose a new method for classifying hospital records by ICD codes. The documents – hospital records – must be classified in the most relevant ICD categories. Our method is based on simple counting on a corpus of specialist classified records. Section 2 reports on the state of the art in text classification. Section 3 describes our new method. Section 4 reports on the experimentations that demonstrate the validity of the approach. Section 5 concludes the paper.

2. State of the Art

Many approaches have been proposed in the text categorization field, and many evaluations of them exist in the literature [2][8][19]. In this paper, we focus on a particular text categorization problem. Documents belong to several categories and there is a high number of categories, up to thousands and even tens of thousands. Yang [3] reported that “k-NN is the only learning method that has scaled to the full domain of MEDLINE categories, showing a

graceful behavior when the target space grows from the level of hundred categories to a level of tens of thousands”. Several characteristics of this method make it preferable, i.e., efficient to test, easy to scale up, and relatively robust as a learning method. However, SVM [6][13] was not tested in [3]. Joachims [18] evaluated SVM against the best classification algorithms (Bayes, Rocchio [9][11], C4.5 and k-NN). SVM performed best, but k-NN had also very good performances, being next to SVM – 95% in classification quality compared to SVM. Yang [5] fairly tested five categorization methods (SVM, k-NN, LLSF, NNet [7] and Naïve Bayes [20]) focusing on the robustness of these in dealing with a skewed category distribution and concluded that SVM and k-NN outperforms the other classifiers. The problem with k-NN is that it is a lazy algorithm which does not scale up with the dimension of the training collection. SVM is a non overlapped labels categorization algorithm, so it is not easy to use in a multi-label multi-category problem. Its complexity makes it also more difficult to implement. We use the *SVM light* [4] package developed by Joachims and compare the classification quality of the four algorithms.

3. The Proposed Method – Matrix Regression

We propose a new supervised learning method for text categorization. We call this method Matrix Regression (MR) due to its similarity to regression models. The training phase builds a terms-categories matrix and the scoring phase uses this matrix to classify new documents. The method we propose is simple and intuitive.

3.1 The Training Phase

The intuition behind our training phase is to evaluate how important is a term to a category in a training collection. We use a weighted matrix represent such term-category association (see Figure 1). To build the model, we consider the ordered set of all the concepts (terms) that appear in the documents of the training collection. This set is obtained as the union of all concept vectors of the training documents. Each document has a vector of concepts associated. We denote with T the set of terms of the training documents:

$$T = \{t_1, t_2, \dots, t_T\}$$

where t_i is a term. $|T|$ is the total number of terms in the training collection.

We build then the ordered set of all the categories that appear in the training collection. This set is the union of the category sets of the pre-classed documents. We denote with C this set of categories of the training collection:

$$C = \{c_1, c_2, \dots, c_C\}$$

where c_i is a category. $|C|$ is the total number of categories in the training collection. In the ideal case, C is equal to the total number of possible categories. As we will see in section 4, the training collection does not always contain documents that cover all possible categories.

We attach to these two sets a matrix $W = (w_{ij})_{T \times C}$, where the lines correspond to terms and the columns correspond to categories.

$$W = \left(\begin{array}{c} \overbrace{w_{11} w_{12} \dots w_{1C}}^{\text{Categories}} \\ w_{21} w_{22} \dots w_{2C} \\ \dots \\ w_{T1} w_{T2} \dots w_{TC} \end{array} \right) \left. \vphantom{\begin{array}{c} \overbrace{w_{11} w_{12} \dots w_{1C}}^{\text{Categories}} \\ w_{21} w_{22} \dots w_{2C} \\ \dots \\ w_{T1} w_{T2} \dots w_{TC} \end{array}} \right\} \text{Terms}$$

Figure 1: Matrix Regression Model

Initially, $w_{ij} = 0, \forall i, j$. Then, for each document in the training collection, we obtain the list of categories with which the document is labeled. For each term of the current document vector and for each category we identify the associated counter w_{ij} in the matrix. We increment the counter with the *tf-idf* (term frequency inverse document frequency) weight of the term.

After building the model, it is stored on the disk. Each time we need to classify new documents, the model is read and the scoring phase is applied to these documents. The model built in this phase is incrementally maintained. If new documents are added to the training collection, the algorithm checks for new concepts and new categories in the new documents. New lines and/or columns are added correspondingly. Then the counters are updated for the new affected lines/columns and for the modified existing terms.

3.2 The Scoring Phase

The second phase in the supervised learning process is the scoring phase. First, we read the saved model represented by the W matrix and the two sets: T and C . This is made once at the beginning of the process. Each new document to classify is represented by the associated term vector denoted T_d . The vector $F = (0)_{1 \times T}$, called the filtering vector, is built. It corresponds to the set of terms T . Let $T' = T_d \cap T$. For each element of T' , the corresponding value of the element of F is set to 1. The role of this vector is to filter the lines of the training matrix W , lines that correspond to the terms of the new document to classify. The filtered lines are then used to propose the categories (labels) of the new document. The result is $W' = F \times W, W' = (w'_{ij})_{1 \times C}$, which is a vector of size C , where the elements correspond to weights associated to each possible category. The “non-interesting” lines of the matrix W have been put to zero by the product with F and the “interesting” ones have been added. We recall that the matrix W is the term-category association given by a weight established in the training phase.

The last step is to filter the categories with a threshold value. Different strategies exist for category filtering [14]. We propose to assign to the new document the categories that have the associated weight greater than the threshold value. The pseudo code of the scoring phase of the algorithm is presented in the Figure 2.

1. Read the model: W, T, C
2. Input: new document with the associated term set T_d
3. Build the vector $F = (0)_{1 \times T}$ and the set $T' = T_d \cap T$
For each element of T' modify to 1 the corresponding value in F
4. Obtain the vector $W' = F \times W, W' = (w'_{ij})_{1 \times C}$
5. Use the vector and a threshold value to propose as categories the ones that have the associated counter above the threshold

Figure 2: Matrix Regression – scoring phase

The principles behind this algorithm are easy to explain. The basic idea is to measure the strength the association between the concepts and the categories. The pre-classed documents, represented by the term vectors, are formed of a list of concepts and a list of categories. In each document, there are several concept-category associations. The matrix role in the training phase is to separate the associations that are initially mixed in each document. The strength of the association between a concept and a category is measured by a weight in this matrix. If the weight value is relatively high, this indicates a possible association between the

concept and the category. If not, the category will be suppressed by the threshold value in the scoring phase.

4. Experimental Results

We used our method for a real-life application and compared the results with those of our implemented version of k-NN and Centroid [10] from one hand, and with those of *SVM light* package developed by Joachims from the other hand. The application requires classifying documents from the medical field. For every patient in a hospital, a clinic, etc. a hospital report (HR) is made by the doctor. The HR includes several paragraphs: the name and age of the patient, the antecedents, the history of the patient's diseases, the evolution of the patient, the conclusions, etc. The last paragraph contains a list of ICD codes. These codes represent the diseases of the patient and they are given in conformity with the *Statistic Classification of Diseases and Connected Health Problems ICD-10* [15]. This gives a standardized codification of diseases; each disease has an ICD code associated.

There are approximately 10000 ICD codes, among which 1000 are often used. The ICD-10 codes are organized in a five level hierarchy that is portioned in 21 chapters that cover all the human diseases. The chapters contain 266 intervals; the intervals contain 1036 categories, which contain 9994 sub-categories. It is difficult for an expert to know all the ICD codes and to complete the corresponding paragraph when he makes a HR. The supervised learning can be used to help codifying these documents. This constitutes one primary objective of the project RNTS¹ Rhea that is in progress.

We use as training collection some data that is gathered from different French hospitals. The collection unites 30 thousands HRs that are already classified. All the paragraphs of a HR contain text. The text is in form of medical language, so it principally contains medical terms. There is an average of five labels for each document, with some documents labeled with up to 32 categories.

Performance measures

We implemented and tested the three algorithms (MR, k-NN and Centroid) and used *SVM light* package for testing SVM. We disposed of a database with 30919 HRs. In the vectorization phase, we only considered the "conclusion" field, as we think it is the most important. For increasing the precision of the results, all fields must be considered, possibly with a weighting on each. The process of vectorization being very slow, we preferred to simplify it to the maximum. The resulted vocabulary size has 4218 features (terms) with the documents belonging to 1996 categories.

After the vectorization, we passed to the classification phase. We used as the training collection the first 20 thousands HRs. To evaluate the built model, we used the last 10 thousands HRs². The input data was the same for all algorithms.

At the testing phase, each method takes as input a list of documents to classify. For each document, the algorithms generate as output a list of categories – the classes to which the document possibly belongs. To measure the performances of the four methods, we used the next three measures: the recall, the precision and the F-measure. These measures, defined as below, are well known in the literature and have been used in multi-label multi-category classification. The definitions of these measures, adapted to our problem, are given below. We have the following notations:

¹ In the framework of the French National RNTS (Réseau National des Technologies de la Santé)

² We organized the data so that each ICD code presented in the test documents is represented by at least five documents in the training collection. We also mention that different splits of the corpus gave similar results.

$|correct\ ICDs| = n$, the number of known correct categories for a pre classed document;
 $|proposed\ ICDs| = m$, the number of categories proposed for a document by a classification method;

$|correct\ ICDs \cap proposed\ ICDs| = k$, the number of correctly found categories for a document by the classification method. With these notations, we give the definitions in (1). Note that we directly defined F_1 -measure, which we used and is derived from the general formula $F_\beta(r, p) = \frac{(\beta^2 + 1)rp}{\beta^2 p + r}$, with $\beta = 1$, meaning that the recall and the precision are equally weighted.

$$Recall = r = \frac{k}{n}, Precision = p = \frac{k}{m}, F_1 - measure = \frac{2rp}{r + p} \quad (1)$$

Comparison between MR, k-NN, SVM and Centroid

We tested k-NN with a value for k between 10 and 120, with a step of 10. We then fixed the value of k to 70, because this value was given the most representative results. For the graphics in the following figures, for k-NN, only the threshold value is variable. For MR, SVM and Centroid, there is a single parameter: the threshold value that filters the categories. For a given threshold value (specific for each method), we execute the programs that classify several thousands HRs. Each program computes after the classification the average values for recall, precision and F1-measure.

Figure 3 traces the precision/recall curve (left graphic) and gives the maximal F1-measure (right graphic) for the four methods. The performances of MR are better than those of k-NN. For a recall of 0.431, the precision is 0.359 for MR and only 0.194 for k-NN. The line describing the MR is above of the k-NN line, so in general, for the same recall we will have a greater precision with MR. The Centroid-based algorithm gives very poor results. This is expected because this algorithm is not appropriate for the multi-label classification problems. The reason we tested this method is the apparent similarity with MR.

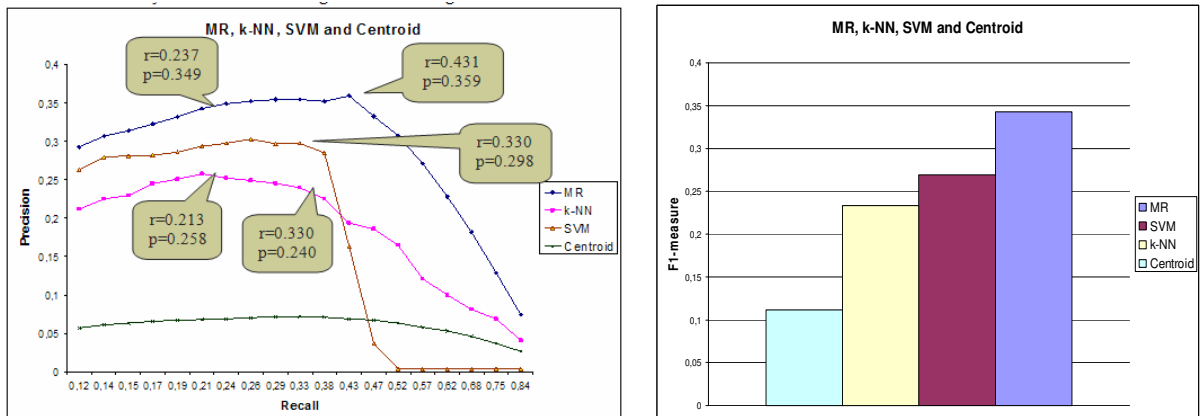


Figure 3: MR, k-NN, SVM and Centroid – precision/recall and F1-measure, maximal values

SVM represents a reference point in the area of text categorization; therefore, we could not omit a comparison with this method. Due to its complexity, we opted for an already implemented version of the algorithm, which usually includes many optimizations. We choose the package *SVM light* developed by Joachims [16]. The program resolves binary classification problems. Joachims proposes an adaptation for multi-class classification in *SVM struct*, but only for predicting one of k mutually exclusive classes. In our experiments, we

adapted *SVM light* for multi-label multi-category problems. The results show that SVM slightly improves the classification results of k-NN, but still remains behind MR.

5. Conclusions and Future Work

In this paper, we presented a new supervised learning method that we called Matrix Regression. We applied this algorithm to classify automatically Hospital Reports. We directly compared its performances with k-NN that is reported in [4] and [2] to be a very good classification algorithm. The results demonstrate the validity of our approach.

Our method is more precise than k-NN in terms of classification quality, while it also addresses the computational scalability problem of the latter. One line of our future work is to improve MR complexity in terms of memory space. Indeed, the matrix size becomes very large as the terms and/or categories increase. Using the property of the matrix sparseness, i.e., with many counters valued to zero or negligible in comparison with the others, compression techniques should be developed and evaluated for occupancy optimization. Another approach could be the application of feature selection [17] upstream on the term-document matrix, but this decreases the training phase performances.

6. References

1. Mitchell, T., *Machine Learning*. Boston, MA, McGraw-Hill. 1998.
2. Yang, Y. and X. Liu, *A re-examination of text categorization methods*, in *SIGIR-99*. 1999.
3. Yang, Y. *An evaluation of statistical approaches for text categorization*. in *Information Retrieval, I(1)*. 1999.
4. Joachims, T., *Text categorization with support vector machines: Learning with many relevant features*. 1998.
5. Yang, Y., *A scalability analysis of classifiers in text categorization*, in *Proceedings of SIGIR-03, 26th ACM International Conference on Research and Development in Information Retrieval*, ACM Press, New York, US: Toronto, CA. 2003.
6. Cortes, C. and V. Vapnik, *Support Vector Networks*. 1995. p. 273-297.
7. W., E., J.O. Pedersen, and A.S. Weigend, *A Neural Network Approach to Topic Spotting*. 1993. p. 22-34.
8. Lewis, D.D. and M. Ringuette, *A Comparison of two learning algorithms for text categorization*. 1994. p. 81-93.
9. Rocchio, J.J., *Relevance feedback in information retrieval*, In *the SMART retrieval system: Experiments in automatic document processing*, Englewood Cliffs, NJ. 1971. p. 313-323.
10. Han, E.-H. and G. Karypis, *Centroid-Based Document Classification: Analysis & Experimental Results*, in *Tech. Rep. 00-017, Computer Science, University of Minnesota*. 2000.
11. Joachims, T., *A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization*, in *Technical Report CMU-CS-96-118, Carnegie Mellon University*. 1996.
12. Koller, D. and M. Sahami, *Hierarchically classifying documents using very few words*, in *In Proceedings of the 14th International Conference on Machine Learning, Nashville, Tennessee*. July 1997.
13. Kwok, J.T.-Y., *Automated Text Categorization Using Support Vector Machine*. 1998. p. 347-351.
14. Kou, H., *Intelligent Web Wrapper Generation Using Text Mining Techniques*. 2003, PhD Thesis, University of Versailles Saint-Quentin-en-Yvelines.
15. *Official ICD10 Swiss Website*. [cited; Available from: <http://www.icd10.ch/index.asp>
16. *SMV light package*. [cited; Available from: <http://svmlight.joachims.org/>].
17. Yang, Y. and J.O. Pedersen, *A Comparative Study on Feature Selection in Text Categories*, in *Proceedings of the 14th International Conference on Machine Learning*. 1997.
18. Joachims, T.: "Text categorization with support vector machines: learning with many relevant features". In C. Nédellec and C. Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pages 137–142, Chemnitz, DE, 1998. Springer Verlag, Heidelberg, DE. Published in the "Lecture Notes in Computer Science" series, number 1398.
19. Boaz Lerner, Neil D. Lawrence : "A Comparison of State-of-the-Art Classification Techniques with Application to Cytogenetics". Computer Laboratory, University of Cambridge, Cambridge, UK. Neural Comput & Applic (2001). 2001 Springer-Verlag London Limited.
20. K-M. Schneider : "On Word Frequency and Negative Evidence in Naive Bayes Text Classification", in *proceedings of ESTAL2004, Advances in Natural Language Processing*, Alicante, Spain, LNAI Lecture Notes in Artificial Intelligence, Springer Verlag 2004, Volume 3230, pp 474-485.