EuroVis Workshop on Visual Analytics (2012)
K. Matkovic and G. Santucci (Editors)

Visual Exploration of Feature-Class Matrices for Classification Problems

W. Kienreich[1] and C. Seifert[1,2]

[1]Know-Center Graz, Austria
[2]Knowledge Management Institute, TU Graz, Austria

Abstract

When a classification algorithm does not work on a data set, it is a non-trivial problem to figure out what went wrong on a technical level. It is even more challenging to communicate findings to domain experts who can interpret the data set but do not understand the algorithms. We propose a method for the interactive visual exploration of the feature-class matrix used to represent data sets for classification purposes. This method combines a novel matrix reordering algorithm revealing patterns of interest with an interactive visualization application. It facilitates the investigation of feature-class matrices and the identification of reasons for failure or success of a classifier on the feature level. We discuss results obtained by applying the method to the Reuters text collection.

Categories and Subject Descriptors (according to ACM CCS): H.5.2 [Information Interfaces and Presentation]: User Interfaces—Graphical user interfaces

1. Introduction

Consider a classification application that turns out not to work as expected. Finding the reasons for the failure is a non-trivial problem. Machine learning experts use more or less structured processes including trial and error elements to find out why a classification application does not work sufficiently well. Some steps in the process are straightforward, others include more creative investigations. For instance, bias-variance problems can be found and solved by plotting learning curves. Highly skewed classes can be found using a histogram of the training data. However, solving the resulting problems may not be so simple. Problems on the feature level, e.g. dependency or redundancy of features, are hard to spot [QHY05]. In application scenarios, the challenge is to mediate between user assumptions, which are frequently based on valuable domain knowledge, and actual properties of the dataset.Why can two classes, which are associated with distinct concepts in the user's mind, not be separated by selected terms describing the concepts? In our work, we try to generate visual answers to such questions.

We propose an interactive visualization that allows to visually exploring the features for classification problems. We exploit the structure of the feature-class matrix composed of few columns, which correspond to classes, and many rows, which correspond to features. The visualization provides an overview on the feature-class matrix as well as zooming and filtering facilities. Details about specific matrix areas are available on demand. Data transformation is available primarily through various matrix reordering algorithms. Visual analysis is tightly coupled with data transformation and enables the interactive exploration of very large data sets. The presented visualization is work in progress. In this paper, we describe our approach and report on experiments with a well-known text classification dataset, the Reuters-21578.

2. Related Work

Matrix Reordering: The problem of sorting a matrix to identify meaningful patterns is known as matrix reordering, and was solved manually for small matrices already in 1967 by Bertin [Ber99, Ber01]. With the availability of graphical computer devices, interactive visualizations became possible [Sii99, SM05]. Interactions include sorting rows and columns, dragging rows and columns. However, these user interface considerations only apply to sufficiently small matrices which fit on the screen as a whole. Mäkinen showed that many reasonable problems of matrix reordering are NP-complete [MS00, MS05]. Thus, all practically applicable algorithmic solutions involve some kind of heuristics, a survey of reordering techniques can be found in [Hen08, Lii10].

DOI: 10.2312/PE/EuroVAST/EuroVA12/037-041

Matrix Visualizations: Henry and Fekete present MatrixExplorer [HF06], combining node-link diagrams and matrix visualization, for analyzing large social networks. MatrixExplorer is based on a requirements analysis for social sciences applications. Matrix reordering is applied to find potentially interesting clusters in the network. The synchronized node-link diagram then can be used to confirm or reject the hypothesis. Similarly, the zoomable adjacency matrix explorer (ZAME) [EDG*08] was specifically designed to represent large networks at different levels of zoom. Intelligent data aggregation and matrix reordering allows users to quickly get an overview of large adjacency matrices. Both the MatrixExplorer and ZAME have been designed for the visual analysis of graph adjacency matrices. Note that such matrices have both a large number of columns and rows, and that the entity type (nodes in a graph) is uniform.

Table Visualizations: Our matrix visualization extends the idea of the TableLens [RC94] for exploring large tables. The TableLens is a focus and context technique presenting a fish-eye view of the rows in the focus and aggregating the rows in the context. In the TableLens visualization, the general pattern of the cells is not visible anymore for cells in the focus area. Thus, we implement an additional overview that is synchronized with the table.

3. Text Classification and Term-Class Matrix

In single-label text classification text documents are assigned to one of the predefined classes by a classifier. A text classifier is trained by example document-class pairs. In order to allow the classifier to process the text documents, the text needs to be converted to feature vectors [Seb05]. Thus, after pre-processing, each training document has a feature vector and a class label assigned. The training documents for each class can be aggregated and visualized in the term-class matrix. The term-class matrix visualizes the importance of a term for each class. The term-class frequency tcf of term t for class c is the number of occurrences of term t in the documents belonging to class c. The normalized term-class frequency ntcf is the term-class frequency divided by the total number of terms for this class. It is an estimate of the probability that term t occurs in class c. The term-class matrix is then defined as follows:

	class_1	…	class_n
term_1	ntcf_{11}	…	ntcf_{1n}
⋮	⋮	⋱	⋮
term_m	ntcf_{m1}	…	ntcf_{mn}

4. Method

In this section, we outline a method for visualizing feature-class-matrices. We first define the patterns of interest which should be revealed by the method. We then describe how to detect the patterns and introduce a matrix sorting algorithm which reveals groups of patterns. We finally describe a visual interface which accounts for the large number of rows which characterizes the use case of text classification.

4.1. Visual Patterns

The term-class matrix is designed to show the relevance of features for a text classification task. In the following we describe visual patterns that can appear in the matrix visualization and their meaning with respect to the classification task. Figure 1 gives an overview of the patterns.

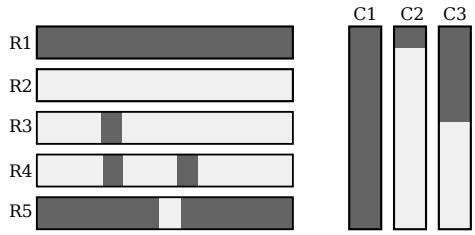

Figure 1: *Interesting row (left) and column patterns (right) in term-class matrix*

R1: The term occurs frequently and nearly uniformly distributed over all classes (so-called stopwords). The term is irrelevant for classification.

R2: The term occurs rarely and nearly uniformly distributed over all classes. The term is irrelevant for classification.

R3: The term is frequent in the documents for one class, and occurs rarely for the other classes. This term is highly informative for classification.

R4: The term is frequent in the documents for two classes, and occurs rarely for the other classes. This term is highly informative for classification, it can be used to distinguish two classes from all others.

R5: The term is frequent for all but one class. The term's absence is informative for classification.

C1: The class contains all terms with high frequency. It is likely to be hard for the classifier to separate this class from all other classes.

C2: The class has only a few frequent terms. If these terms do not overlap with the terms from the other classes (see row patterns), the class can be well-separated.

C3: The class has approximately half of all terms frequently.

These row- or column-wise patterns can be generalized over multiple rows and columns, i.e. over multiple features or classes. For instance, the pattern **R4** repeated over multiple neighbouring rows represent a set of features that are highly informative for a single class. To make such patterns visible a sensible reordering of the matrix is necessary.

4.2. Matrix Reordering

We implemented two well-know matrix reordering algorithms, threading along a column [SM05] and 2D

Figure 2: *Sorting the Reuters R8 dataset. Natural order in dataset (left), Threading along column (center), 2D Sort (right)*

sort [MS00]. *Threading along a column* sorts all rows such that the values of the respective column are sorted ascendingly. The *2D sort* iteratively rearranges rows and columns by the weighted sum of their entries. More specifically, the algorithm is as follows: (i) calculate weighted row sums, the weight is the column position of the cell, (ii) arrange matrix ascendingly according to these row sums (sort rows), (iii) calculate weighted column sums, the weight is the row position of the cell, (iv) arrange matrix ascendingly according to these column sums (sort columns), (v) repeat step (i) to (iv) until no reordering occurs. Results of applying these algorithms to the Reuters-21578 training dataset are shown in figure 2. The Reuters-21578 data set [Reu] is a standard data set for text classification tasks. It contains news articles manually categorized by Reuters in 1987. We used the train split of the single-label R8 subset containing 5485 documents, 14575 rows (terms) and 8 columns (classes).

We implemented a novel sorting algorithm, called pattern sort. The motivation of pattern sort is to make the described row patterns visible across multiple adjacent rows. Each pattern is defined by the number and location of peaks in the term frequency distribution within a row. Pattern sort therefore sorts rows first by the number of peaks and then by an encoding of the location of peaks. The methods for peak detection and peak location encoding may depend on data set and use case. In our work, we have explored approaches suitable for the Reuters-21578 training dataset.
Peak detection: Several statistical approaches are suitable for detecting peaks in the term frequency distribution of a row. Intuitively, values which significantly differ from the mean frequency of a term constitute peaks. We therefore identify a cell as a peak if its value is larger than half the standard deviation of the normalized term frequency for its row. Different data sets and use cases may profit from adapting this measure. For instance, in sparse data sets a simple non-zero treshold may be suitable to identify peaks for location encoding purposes.
Peak location encoding: In our work, we have encoded peak location as a bit vector directly derived from the column location of detected peaks. For each row, an integer sort index is created from the number and location of identified peaks. The lower n bits (where n denotes the number of columns) of the index encode peak positions as a bit vector. The bits above n encode the number of peaks. More advanced peak location encodings could be employed to better characterize the similarity between peak location patterns.

In our experiments with the Reuters-21578 training dataset, we identified the number of peaks by counting the values larger than half the standard deviation in a row. We encoded the position of peaks using a bit vector based on the column position of non-zero values. Figure 3 shows the matrix with different sorting criteria: left: sort by number of peaks (without considering peak location); center: sort by number and position of peaks (values larger than half the standard deviation); right: sort by number of peaks (values larger than half the standard deviation) and position of peaks (non-zero values).

4.3. Visualization

The visualization is centred around a matrix rendering view (compare figure 4). Matrix cells are represented by a pixel area of constant width (corresponding to horizontal space available for columns) and varying height (corresponding to vertical space available for rows and magnification factor). On high levels of magnification, each matrix row covers one or more rows of pixels and cells appear as filled rectangles. On low levels of magnification, one row of pixels aggregates values from several matrix rows. In this case, we facilitate smooth rendering and transition by vertically aggregating matrix cell values.
Column labels are rendered centred atop the horizontal range allocated for each column, the label size is adapted to the magnification factor. Row labels are rendered dynamically left of the horizontal range allocated to each row. If one row of pixels corresponds to more than one row in the matrix, we

Figure 3: *Visualizing the Reuters R8 dataset with pattern sorting. Rows reordered by number of peaks (left), number and position of peaks (center), number of peaks and position of non-zero cells (right)*

render a pixel line with a length corresponding to the average length of the aggregated labels. This approach provides a smooth transition from readable labels to structural outline showing only label length. Matrix cell values (individual and aggregated) are currently represented by pixel values on all three RGB channels, yielding a view where fully filled cells are black and cells with zero values appear white. Other RGB mappings will be explored, for instance to compare data sets. A scaling operator enables non-linear mappings which account for non-linear value distributions in data sets. Navigation is facilitated by an overview which is rendered with the same algorithm used for the detail view, but with the magnification factor adjusted to show the whole dataset at once. The currently selected area in the matrix can be modified by mouse drag operations and current magnification factor can also be changed using mouse interactions. The area visible in the detail view is highlighted in this overview (small white area, figure 4). The use of a separate overview leaves mouse operations on the detail view free for further operations like table lens or selection operations. A search facility is available; all rows whose label contains a specified search term are highlighted in the overview and in the row label display of the detail view (red lines, figure 4).

5. Examples

Figure 3 (right) shows the matrix visualization using the proposed reordering algorithm on the Reuters R8 data set (with the secondary sort criteria counting each non-zero cell value as a peak). Some observations can immediately be made: Pattern R1, which denotes ill-suited terms, occurs exclusively in the lowest part of the visualization. Pattern R3, which denotes well-suited terms, occurs exclusively in the upper part of the visualization. Significant numbers of pattern R4, which denotes term overlap between two classes, occur for example for classes acq(uisition) and earn(ings). From the small range denoting R1 Pattern terms, we could guess that classes interest and grain are under-represented.

Figure 4: *Application window, overview and detail views.*

6. Summary and Conclusion

We proposed a visualization for the interactive exploration of feature-class matrices. The visualization combines matrix reordering algorithms which group patterns of interest with matrix rendering algorithms which account for the specific structure of the feature-class matrix. We presented results obtained by applying the visualization to a standard text classification dataset. Future work includes the extension of data transformation to other feature weighting schemes (such as class-averaged TF-IDF and BM-25), the implementation and evaluation of further peak detection methods and the user evaluation of the visual interface. A demonstration version of the visualization is available online at http://www.know-center.at/matvis.

Acknowledgement

The Know-Center is funded within the Austrian COMET Program under the auspices of the Austrian Ministry of Transport, Innovation and Technology, the Austrian Ministry of Economics and Labor and by the State of Styria. COMET is managed by the Austrian Research Promotion Agency FFG.

References

[Ber99] BERTIN J.: *Graphics and graphic information processing*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999, ch. 1D, 2D, 3D, pp. 62–65. 1

[Ber01] BERTIN J.: Matrix theory of graphics. *Information Design Journal 10* (2001), 5–19. 1

[EDG*08] ELMQVIST N., DO T.-N., GOODELL H., HENRY N., FEKETE J.-D.: Zame: Interactive large-scale graph visualization. 215 –222. 2

[Hen08] HENRY N.: *Exploring Social Networks with Matrix-based Representations*. PhD thesis, University South Paris and University of Sydney, July 2008. 1

[HF06] HENRY N., FEKETE J.-D.: Matrixexplorer: a dual-representation system to explore social networks. *IEEE Transactions on Visualization and Computer Graphics 12* (2006), 677–684. 2

[Lii10] LIIV I.: Seriation and matrix reordering methods: An historical overview. *Wiley Periodicals 3*, 2 (2010). 1

[MS00] MÄKINEN E., SIIRTOLA H.: Reordering the reorderable matrix as an algorithmic problem. In *Diagrams '00: Proceedings of the First International Conference on Theory and Application of Diagrams* (London, UK, 2000), Springer-Verlag, pp. 453–467. 1, 2

[MS05] MÄKINEN E., SIIRTOLA H.: The barycenter heuristic and the reorderable matrix. *Informatica (Slovenia) 29*, 3 (2005), 357–364. 1

[QHY05] QU G., HARIRI S., YOUSIF M.: A new dependency and correlation analysis for features. *Knowledge and Data Engineering, IEEE Transactions on 17*, 9 (sept. 2005), 1199 – 1207. 1

[RC94] RAO R., CARD S. K.: The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence* (New York, NY, USA, 1994), CHI '94, ACM, pp. 318–322. 2

[Reu] Reuters-21578 text collection. http://csmining.org/index.php/r52-and-r8-of-reuters-21578.html. 3

[Seb05] SEBASTIANI F.: Text categorization. In *Text Mining and its Applications*, Zanasi A., (Ed.). WIT Press, Southampton, UK, 2005, pp. 109–129. 2

[Sii99] SIIRTOLA H.: Interaction with the reorderable matrix. *Information Visualisation, International Conference on 0* (1999), 272. 1

[SM05] SIIRTOLA H., MÄKINEN E.: Constructing and reconstructing the reorderable matrix. *Information Visualization 4*, 1 (2005), 32–48. 1, 2