Correlating entities based on DSL specifications of entity group relations

Erik Brännström
Chalmers University of Technology

1 Introduction

Duego was founded in 2010 with the idea of creating a social networking site whose target group is those who wish to meet new people. This stands in contrast to the existing networks that either manage people who the user already knows (e.g. Facebook) or where the user tries to find a life partner (i.e. dating sites). Online advertisement is leveraged as a way to promote the site to new users. This is done both using targeted ads that are adapted based on the target demographics on social networks, as well as more conventional ads on websites. Today, managing these campaigns is a manual process that is outsourced to another company.

The wish of Duego is to automate this process using software. Such a system would be required to analyze the existing campaigns with regard to ads and target groups along with campaign metrics such as clicks per impression, conversion ratio, etcetera, and use this data to suggest, or even automatically add, new campaign ads. The data set that will be used in production consists of hundreds of different campaigns, each with a large number of ads. Along with the attributes of both campaigns and ads, this means that the complete advertisement data set consists of hundreds of thousands of attributes to be analyzed.

This problem statement is described from the view of Duego, however it can be generalized into a broader problem in the sphere of software engineering, namely knowledge discovery in databases (KDD) or more specifically data mining. Data mining is the process of delegating and automating the task of identifying useful knowledge in a large set of data to a computer, either fully or partially. The somewhat philosophical discussion of what knowledge really is and how to define what is useful knowledge will not, however, be covered in this paper.

2 Scope

Useful terminology is defined by IBM [2006] in the field of autonomic computing. An autonomic manager is a component that collects data from a system and, based on this data, performs actions with the purpose of improving the system. This control loop is divided into four sub-tasks called monitor (collect system information), analyze (correlate and model data), plan (design behaviour re-

quired to reach goal) and execute (run the planned actions), sometimes refered to as MAPE.

Using this framework, only the analysis and planning tasks are considered part of this paper, where as monitoring and execution are out of scope. The latter two are however relevant in the verification step, but will not be covered as a research topic. In this context, this means for example that the feature of integrating this system with marketing services to automatically add new ads and campaigns will not be a part of the final system.

Furthermore, the data set includes attributes whose values are free text and images. Text mining and image recognition is beyond the scope of this project. Instead these attributes will be manually categorized, so that the value space is discrete and finite.

3 Foundations

A number of high-level descriptions of frameworks for knowledge discovery in databases exist [Fayyad et al., 1996, Frawley et al., 1992] and they exhibit a number of commonalities. These include the importance of having a knowledgeble human operator guiding the process in terms of supplying domain knowledge to the system formulating the goal of the knowledge discovery; feeding discovered knowledge back into the system; and the identification and application of a discovery method, or more specifically the data mining algorithm(s).

In data mining, the input to a system can be described using the terms *concepts*, *instances* and *attributes*, where concept is the actual result of the mining, i.e. what we want to be learned; an instance is one single example of data to be mined and can be compared to a row in a database; and attribute is a property of an instance, which in the database analogy would be a column [Witten et al., 2011].

The Weka Project has defined an input format to be used for their open source data mining software called ARFF (Attribute-Relation File Format) [Garner, 1995, Witten et al., 2011]. This format is used by the Weka software package, but is well-defined and can be used as the input format for custom systems as well. Witten et al. [2011] also describe how to use implementations of Weka in custom software projects as well as how to extend the system with new functionality.

In a highly influential paper, Quinlan [1986] decribe how decision trees can be created from a training set and how well it handles the problem of unknown attributes values and noisy data. A related paper, Quinlan [1987a], deal with how generated decision trees can be simplified in order to more easily be applied. Four different methods are evaluated, one of which is the reformulation of the tree as a set of production rules. This specific topic is further analyzed in Quinlan [1987b] where such production rules are shown to be more compact and also in many cases improve the classification of unseen data. An added positive effect is that production rules from separate classifications can be merged more efficiently than their original decision trees.

Standard decision trees give a best-effort boolean classification of the input data, however sometimes it might be more appropriate to give the probability that the input belongs to each of the available classes. This can be described using probability estimation trees (PETs). Provost and Domingos [2003] discuss

the problems of estimating these probabilities from ordinary decision trees and goes on to show how to increase the accuracy of the estimates by performing tree pruning more conservately and by applying the Laplace correction. They also show that probability-bagging, meaning the combination of results from multiple classifiers instead of just the one, greatly improves the estimates. The suggested algorithm, called C4.4, is part of a comparative study of PETs by Chu et al. [2011] and is shown to have an impressive accuracy, though other algorithms may still be more appropriate. A likely candidate algorithm for this thesis is the Naive Bayes tree, which is a standard decision tree with the difference that the leaves are Naive Bayes classifiers, providing probabilities that an instance belong to each of the available classes.

A thorough literature study on domain-specific languages is covered by van Deursen et al. [2000] and covers high-level topics, such as potential benefits and risks of using DSLs and exemplifying with a number of languages from different areas. The article also mentions the phases of development in the creation of a new DSL, a topic which is described in more detail by Mernik et al. [2005]. The latter article also provides insight into the relevant choices that need be made in the creation process, and both articles help define a useful terminology for discussing the topic of domain-specific languages.

4 Related works

Chen et al. [1996] give an overview of the field of data mining where they mention the aspect of multi-level data mining, which states that correlations may not commonly exist on the lowest level of granularity, but instead by forming groups of related items. An example given would be that a specific brand of milk does not necessarily imply the purchase of a specific brand of bread, however purchasing milk of any kind may still be correlated to the purchase of bread irrespective of brand.

A related area in data mining is association rule learning. Agrawal et al. [1993] describe how, given a large set of transactions containing any number of different items, rules can be identified between these items. In the case of commerce, such rules would describe how the purchase of one product would, with a probability above a certain threshold, also infer the purchase of another product. This method could be applicable for the problem in this paper, even though it is more of a classification problem, and some of the necessary extensions of the method are mentioned below.

Srikant et al. [1997] expands the above research by applying contraints on the items in the resulting association rules. These constraints can either be expressed using specific items or with taxonomy rules, e.g. if an item is a descendant or ancestor of another item. For this problem, the relation between target groups and ads can, based on existing metrics, be restructured into a transactional table where each transaction is an impression (in the case of the impression per click metrics) so that each entry would have an attribute stating whether or not the impression yielded a click. The item constraint would then be used to only formulate association rules that lead to clicks.

Considering that the original market basket problem assumes binary attributes and that relational data can contain both quantative and categorical data, [Srikant and Agrawal, 1996] gives a useful description on how such values

can be mapped onto the binary (or boolean) problem space.

Hahsler et al. [2007] presents a package for the R software environment used in statistical computing, which implements a base for transaction databases as well as integration with two of the most common mining algorithms, Apriori and Eclat. One of the useful implementation choices is the implementation of the sparse matrix data structure, which greatly reduces memory load.

5 Suggested thesis topics

- Define data mining algorithm for generating decision rules of binary classes with probabilities
- Compare the rules in a decision tree with those of an association rule mining algorithm in regard to performance and quality.
- Case-study of the KDD process in online marketing.
- Literature study on the topic of KDD/data mining/decision trees.
- Study the process of selecting input data in regard to the expected outcome, comparing the differences between small, targeted input sets and broad, general dittos. If a lot of data yield better results, can they be cached and updated incrementally?
- Modify an existing algorithm to use the metrics already related to the input data to make the calculations more effecient.
- Discuss the usefulness of heuristics to increase efficiency, while only negligibly decreasing accuracy.

6 Project plan

The proposed workflow of this thesis is inspired by the agile practices that perhaps most notably have become common in the software development industry. The most important methods that should be applied to this work are iterative project development and frequent "customer collaboration", in this case thesis suporvisors from both Chalmers University of Technology and Duego Technologies AB.

The project is expected to run for about 20 weeks. By dividing this relatively long period of time into iterations of four weeks, the project will be easier to manage from my point of view as well as provide ample opportunity for the supervisors to have their feedback incorporated.

The initial iterations will most likely require a heavy focus on research and studies of the related fields. The following iterations will include more work on the practical side of the projects. Once the research has progressed to an extent that the problem space as well as the approach of the solution are well defined, more effort can be put into the development of the final application that is to be used by Duego. A preliminary list of milestones are listed below.

1. Finish extended proposal (June 19)

2. Research problem is to be well-defined and theoretical solution described (July 17)
3. Demo a prototype of the final system (August 14)
4. Research should be finalized and draft of final paper submitted to supervisors (September 11)
5. Submit final paper to examinator and working system to Duego (October 9)

7 Verification

The result of the thesis work would be verified by applying the results to the problem suggested by Duego in a real-world setting. The suggested advertisement and target groups from the system will be added to their online marketing portfolio which will allow us to analyze the effectiveness of the software. This can be done continuously over the course of the development in addition to more traditional software unit and system testing.

8 Thesis outline

- Abstract (0.5 pages)
- Introduction (1 page)
- Related work (1 page)
- ... (Depends on the thesis topic)
- Discussion (1-2 pages)
- Conclusion (1 page)

References

Rakesh Agrawal, Tomasz Imielinski, and Arun Swami. Mining Association Rules between Sets of Items in Large Databases. In *SIGMOD Record*, volume 22, page 207, 1993.

Ming-Syan Chen, Jiawei Han, and Philip S. Yu. Data Mining: An Overview from a Database Perspective. In *IEEE Transactions on Knowledge and Data Engineering*, volume 8, pages 866–883, 1996.

N. Chu, L. Ma, P. Liu, Y. Hu, and M. Zhou. A comparative analysis of methods for probability estimation tree. *WSEAS Transactions on Computers*, 10(3): 71–80, 2011.

Usama Fayyad, Gregory Piatetsky-shapiro, and Padhraic Smyth. Knowledge discovery and data mining: Towards a unifying framework. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, August 2-4, 1996*, pages 82–88. AAAI Press, 1996.

William J. Frawley, Gregory Piatetsky-Shapiro, and Christopher J. Matheus. Knowledge discovery in databases: An overview. *AI Magazine*, 13(3):57–70, 1992.

Stephen R. Garner. Weka: The waikato environment for knowledge analysis. In *Proceedings of the New Zealand computer science research students conference*, pages 57–64. Citeseer, 1995.

Michael Hahsler, Bettina Grün, and Kurt Hornik. Introduction to arules: Mining association rules and frequent item sets. *SIGKDD Explorations*, 2:4, 2007.

IBM. An architectural blueprint for autonomic computing. *Quality*, 36(June): 34, 2006.

Marjan Mernik, Jan Heering, and Anthony M. Sloane. When and How to Develop Domain-Specific Languages. 37(4):316–344, 2005.

F. Provost and P. Domingos. Tree induction for probability-based ranking. *Machine Learning*, 52(3):199–215, 2003.

J. R. Quinlan. Induction of decision trees. *Mach. Learn.*, 1(1):81–106, March 1986. ISSN 0885-6125.

J.R. Quinlan. Simplifying decision trees. *International journal of man-machine studies*, 27(3):221–234, 1987a.

J.R. Quinlan. Generating production rules from decision trees. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, volume 30107, pages 304–307. Citeseer, 1987b.

Ramakrishnan Srikant and Rakesh Agrawal. Mining quantitative association rules in large relational tables. *SIGMOD Rec.*, 25(2):1–12, 1996.

Ramakrishnan Srikant, Quoc Vu, and Rakesh Agrawal. Mining association rules with item constraints. In *Proceedings of the Third International Conference of Knowledge Discovery and Data Mining*, pages 67–73. AAAI Press, 1997.

Arie van Deursen, Paul Klint, and Joost Visser. Domain-specific languages: An annotated bibliography. 35(6):26–36, 2000.

Ian H. Witten, Eibe Frank, and Mark A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, 3rd edition, 2011.