A Data Mining Approach to Choosing Categorical Attributes for Ranked Lists[*]

Koninika Pal
University of Kaiserslautern
Kaiserslautern, Germany
pal@cs.uni-kl.de

Sebastian Michel
University of Kaiserslautern
Kaiserslautern, Germany
smichel@cs.uni-kl.de

ABSTRACT

This work proposes and evaluates a novel approach to determine interesting category for ranked lists using ν-SVM. We identify three characteristics (features), entropy, unlikability, and peculiarity and show how to train a classifier on these features using a set of Wikipedia tables. The learned model is evaluated by relevance assessments obtained through a user study, reflecting the correctness of our approach.

1. INTRODUCTION

Understanding and exploring information is becoming more and more complex due to the dramatic growth of data. Scale, dynamics, and (schema) heterogeneity advocate for solely automated means, by which users can sit back and explore data already put in meaningful and interesting categories. In this work, we specifically look at ranked lists, a concise form of data summarization, that can be found in nearly all domains as virtually everything can be ranked, if not by measurable means then by crowdsourcing rankings. Given a ranked list, for instance, the list of tallest building in the world, we consider the task to decide which dimension is worth being used as a constraint to specialize the query. In database terminology, we are interested in determining OLAP-cube dimensions for the drill-down operation, but consider the case of data beyond a well understood database schema. Specifically, our approach is using statistical measures that can be computed from any table, no knowledge about the semantics of the schema or human input is required. Getting back to the above example, a list of tallest buildings by continent or country appears interesting, while a list of the tallest buildings that are 31 stories tall might be less important to be investigated, if at all. The key idea behind this work is to assume that it is feasible to train a classifier based on training data obtained from Web tables, such as tables in Wikipedia, assuming that the presence or absence of a table can act as in indicator of general interest or disinterest of humans in such a table.

1.1 Problem Statement, Setup, and Key Idea

Consider a set of rankings-style tables $\mathcal{R}$, where $r \in \mathcal{R}$ is a ranking table with its attributes $\mathcal{A}$. A subset of $\mathcal{A}$ is of numeric type, denoted as $\mathcal{N}$. We divide $\mathcal{A}$ in three kinds of attributes: (i) the subject of ranking denoted as a_s which represents the set of entities, (ii) the criterion of the ranking, denoted as a_{cr}, based on which the ranking order of subject entities are made and (iii) the remaining attribute considered as categorical attribute denoted as a_c. Hence, the ranking table r is written as $r = (a_s, a_{cr}, a_{c1}, a_{c2}...)$.

Each attribute $a \in \mathcal{A}$ is associated with a set $\mathcal{V}_a$ of possible values. Following the previous example, assigning a constraint on an attribute, for instance, $a_{c_1}=$ 'Canada' where 'Canada' $\in \mathcal{V}_{a_{c_1}}$ and c_1 denotes the country a building is placed in, specifies the ranking of the tallest buildings in Canada. That is, $(a_{c_1}$, 'Canada'$)$ becomes the category for r_{new}.

Let $\mathcal{I}$ be the complete set of interesting categorical attribute. Our objective is to create a classifier $\mathcal{C}$ that can tell whether using a non-numeric categorical attribute as a constraint to a table would lead to an interesting "new" table or not, i.e.,

$$\mathcal{C}(a_c) = \begin{cases} \text{interesting}, & \text{if } a_c \in \mathcal{I} \text{ and } a_c \notin \mathcal{N} \\ \text{not interesting}, & \text{otherwise} \end{cases} \quad (1)$$

In this work, we opt for a classification-based approach using ν-SVM as our classifier. Hence, we first need to determine training data $\mathcal{T}$ of interesting categorical attributes, and we do so by harnessing a set of Wikipedia tables $\mathcal{R}$. **Based on our assumption, categorical attributes for a specific subject are considered interesting iff we find at least one ranking in Wikipedia that is created by imposing a constraints over that categorical attributes.** Formally, $(a_s, a_c) \in \mathcal{T}$ iff $\exists r, r_{new} \in \mathcal{R}$. Note that the interestingness of a category is bound to the entity type (i.e., class of the subjects), which satisfied by the condition: $r.a_s = r_{new}.a_s$. Then, we learn the model using SVM on the training data and verify how accurate we can predict interesting category for ranking by evaluating it on held-out test data and by relevance assessments obtained from a user study.

2. WORKING MODEL

Creation of Training Data: Algorithm 1 retrieves interesting and non-interesting categorical attributes for a specific ranking subject (i.e., (a_s, a_c)). The constraints of a ranking table is parsed from the title/caption of the table or the title of the Wikipedia page by using propositions from the English dictionary, presented by the function in line 6 in Algorithm 1.

Learning Interesting Categories: In this work, we use the *soft-margin classifier* ν-SVM [4] to learn the interesting characteristics of categorical attributes. $\nu-$SVM suits best for our purpose as it can detect outliers while learning. According to our intuition, conciseness and diversity of a categorical value of ranking entity is important to capture

[*]This work has been supported by the German Research Foundation (DFG) under grant MI 1794/1-1.

Algorithm 1 Generating Training Samples

```
1:  procedure GENERATESAMPLES($wikitables$)
2:      $contraintsmap \leftarrow$ empty map(constraints, subjectList)
3:      $interesting \leftarrow$ empty list(subject, attribute)
4:      $nonInteresting \leftarrow$ empty list(subject, attribute)
5:      for $\mathcal{T} \in wikitables$ do
6:          $\mathcal{T}.a_s, \mathcal{T}.cons \leftarrow parse\_cons(\mathcal{T}.title)$
7:          $contraintsmap[(\mathcal{T}).cons] \leftarrow (\mathcal{T}.cons, \mathcal{T}.a_s)$
8:      for $\mathcal{T} \in wikitables$ do
9:          $[\mathcal{T}.a_c, \mathcal{V}_{A_c}] \leftarrow parse(\mathcal{T}.table)$
10:         for $a_c \in \mathcal{T}.A_c \setminus \mathcal{N}$ do
11:             for $x \in \mathcal{V}.a_c$ do
12:                 $subjectList \leftarrow contraintsmap.contains[x]$
13:                 if $\mathcal{T}.a_s \in subjectList$ then
14:                     $interesting \leftarrow (\mathcal{T}.a_s, a_c)$
15:                     break;
16:                 else
17:                     $noninteresting \leftarrow (\mathcal{T}.a_s, a_c)$
18:     return $interesting, noninteresting$
```

human interests. Hence, we use the following three measures as features to feed our learning model.

Shannon Entropy reflects the uncertainty of information content of a discrete random variable. Here, we treat a categorical attribute $a_c \in \mathcal{A} \backslash \mathcal{N}$ as a random variable, where V_{a_c} is the set of possible values that a_c can hold. Shannon entropy is calculated by $H(a_c) = -\sum_{x \in \mathcal{V}_{a_c}} P(x) \log_2 P(x)$, where $P(x) = count(x)/|\mathcal{T}|$, $|\mathcal{T}|$ is the size of the ranking table. The normalized entropy is calculated as $\hat{H}(a_c) = \frac{-\sum_x P(x) \log_2 P(x)}{\log_2 |\mathcal{T}|}, x \in \mathcal{V}_{a_c}$.

Unlikeability is a diversity measures for categorical attribute that measures how often the observation of random variables differs from one another [2], calculated as $U(X) = 1 - \sum_{x \in \mathcal{V}_{a_c}} P(x)^2$.

Peculiarity is another diversity measure for categorical value used here to measure the peculiarity which is defined by the probability that a randomly chosen categorical value has not been seen previously [2]. It is defined by $D(X) = 1 - \sum_{x \in \mathcal{V}_{a_c}} \frac{count(x)(count(x)-1)}{|\mathcal{T}|(|\mathcal{T}|-1)}$

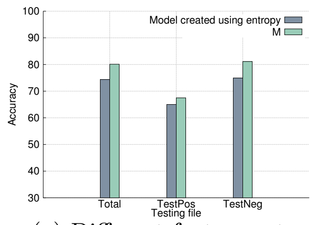
All three measures are normalized to $[0, 1]$, the values towards 1 indicate few or only one distinct categorical value of the ranking entity and a value towards 0 represents a large (maximum) diversity.

Thus, it is clear that a categorical attribute of an interesting ranking should have more tendency toward having a feature-value near the mid-range of $[0, 1]$, i.e., around 0.5. In contrast, uninteresting ranking would have a tendency toward having a feature value closer to 0 or close to 1.

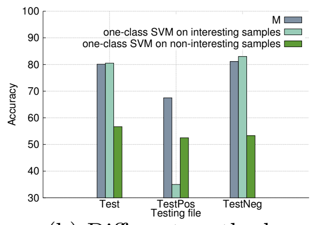
Clearly, the range of values is responsible for the classification of interestingness measures. Hence, we use the Radial Basis Function (RBF) kernel to map our data space to the dot product space needed for SVM .

3. EXPERIMENTS

Setup: We use the LIBSVM [1] tool to learn the models. 2,744 non-interesting and 158 interesting samples are extracted using Algorithm 1 from 2,045 ranking tables out of Wikipedia. 25% of samples are chosen randomly from each class, and also merged for testing purpose denotes as **TestPos** and **TestNeg**, and **Total**. The remaining non-interesting samples are divided into 10 smaller chunks and merged with the remaining interesting samples to create 10 training files, each containing 323 samples. We learn the respective model for all these files and chose the best performing one, denoted as **M** with accuracy of 80.11% for **Total**. According to our testing samples, the accuracy is a fraction of the correctly classified samples. For our training data, a feasible solution for SVM is found where $0 \leq \nu \leq 0.73$ and

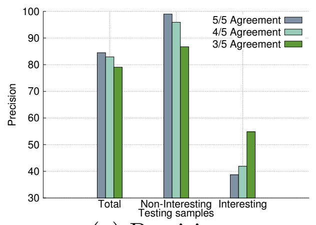


(a) Different feature set

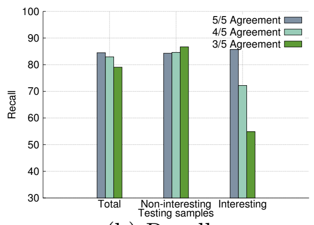


(b) Different method

Figure 1: Comparison among learning models



(a) Precision



(b) Recall

Figure 2: User study

the optimal $\nu = 0.51$. Here, we also present the evaluation of a user study on test set of 130 randomly selected samples of (a_s, a_c), i.e., classification tasks. We gathered five user relevance assessments per classification task. The accuracy of model **M** is evaluated by using ground truth that is built for different agreement levels of users.

Validation of Models on Test Data: In Figure 1(a), we can see that the accuracy increases at least 5% for all testing files except **TestPos** (2.5%) while using all three features together in learning. As our training data is unbalanced, a common option is to use the simpler one-class SVM model. However, Figure 1(b) shows that the model **M** is a better classifier than the model possible to create from an one-class SVM. **M** is also more robust as it classifies samples from each class better than the other two models.

Results Based on User Study: Figure 2 is showing precision and recall achieved by **M** with varying agreement level of user. We see that for a user agreement of 5/5 the highest accuracy is reached. That is, considering the tasks with 5/5 user agreement as ground truth of the test data, our model correctly identify more than 80% of the test cases (Figure 2). We use Fleiss Kappa to determine the reliability of user agreements. The 5/5 user agreement has a Kappa score 0.28, which denotes fair agreement according to a commonly cited interpretation of Kappa values [3]. Also, the 95% confidence interval for Kappa has range between 0.24–0.32 for the collected user data. These values significantly differ from 0 and, thus, prove the statistical significance of 5/5 agreement level for our user-study.

4. CONCLUSION

From the experimental evaluation, we can conclude that our model of classifying category to capture the interesting ranking performs very well. We also saw that all three features together create a better classification model than the commonly used entropy measure.

5. REFERENCES

[1] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27, 2011.

[2] G. D. Kader and M. Perry. Variability for categorical variables. *Journal of Statistics Education*, 15(2).

[3] J. R. Landis and G. G. Koch. The measurement of observer agreement for categorical data. Biometrics, 33, 1977.

[4] B. Schölkopf et al. New Support Vector Algorithms. *Neural Computation*, 12(5), 2000.