A similarity-based Chinese Restaurant Process for Social Event Detection

Athanasios Papaoikonomou
National Technical University of Athens
tpap@mail.ntua.gr

Konstantinos Tserpes
National Technical University of Athens
tserpes@mail.ntua.gr

Magdalini Kardara
National Technical University of Athens
nkardara@mail.ntua.gr

Theodora Varvarigou
National Technical University of Athens
dora@mail.ntua.gr

ABSTRACT

In this paper, we present our approach for the Social Event Detection task of Medieval 2013 [2]. The goal of the task was to group similar multimedia items into event clusters, based on their metadata (e.g. title, description, tags). Since the number of the event clusters in the test set was not known in advance, we formulated a non-parametric algorithm which resembles the Dirchlet process clustering. More specifically, we developed a similarity-based version of the Chinese Restaurant Process (CRP) which exploits the similarities among the media items. Our approach achieved a F1 score of 0.2364.

Keywords

event detection, dirichlet proccess clustering, latent topic discovery, chinese restaurant process

1. INTRODUCTION

The goal of the Social Event Detection task of Medieval 2013 was to discover event-related multimedia items and organize them in event-specific clusters. For this purpose a large training set of about 312,000 photos was given along with their textual metadata like title , description, location and tags.

One of the biggest challenges that we faced had to do with the calculation of the number of event clusters in the test set, since this was not known in advance. To tackle this problem, we used the Chinese Restaurant Process (CRP), which is a formulation of the Dirichlet process. It has attracted its name from its analogy to a Chinese Restaurant where n customers are seated to an infinite number of tables based on the following algorithm: The first customer sits at the first table. All the subsequent customers either sit at one of the previously occupied K tables with probability $\frac{n_k}{n-1+\alpha}$, where n_k is the number of customers already seated at table k, $k = 1, 2, ..K$ or sit at a new table with probability $\frac{\alpha}{n-1+\alpha}$, where α is a pre-defined parameter. The connection between the CRP and our task is pretty straightforward. The photos in the dataset stand for the customers being seated and the tables are the event clusters that group similar multimedia items. The traditional CRP takes into account only the *popularity* of a table in order to decide whether to assign an item to a specific table or not. In Section 3 we will present our modified version of the CRP, which we call similarity-based CRP as it exploits the similarity among the items in the dataset.

MediaEval 2013 Workshop ,October 18-19, 2013, Barcelona, Spain

2. RELATED WORK

One of the most prominent algorithms in latent topic discovery is the Latent Dirichlet Allocation (LDA) presented by Blei et al. in [1]. HDP-LDA is a non-parametric version of LDA, based on the Hierarchical Dirichlet Process clustering algorithm given in [3]. Borrowing ideas from the LDA algorithm (especially the HDP-LDA), we tried to build an event detection algorithm focused on metadata mining. LDA (and its variants) exploit word co-occurences to identify latent topics, but in the case of metadata there are attributes that cannot be modeled directly as words. For example, in the case of the date taken attribute, we do not expect two multimedia items to share the exact same value for the timestamp even if they refer to the same event. A different approach should be followed, as we do in Section 3.

3. APPROACH

In this section we present our algorithm for the Social Event Detection task. Our approach leans on the assumption that similar customers (photos) will tend to gather together and "sit at the same table" in the context of our modified Chinese Restaurant Process. Our analysis focused on the comparison of the available metadata of multimedia items which in our case were typical properties like title, description, tags and username, *spatial properties* like the longitude and latitude, and finally *temporal attributes* like the date that the media item was taken. The following subsections present in a stepwise manner the construction of our algorithm.

3.1 Similarity Computation

Attribute	$prob_a$
Location	56.35%
Username	10.80%
Title	09.34%
Date Taken	25.74%
Tag	48.20%

Table 1: Attribute statistics.The second column reports the percentage of the datapoints that share a common value and belong to the same event cluster.

We evaluated the importance of each attribute to the allocation of the media items in event clusters. More concretely, we operated on the training set and we measured the probability that two datapoints sharing the same value for a specific attribute, will also belong also to the same event cluster.The results are depicted in Table 1. In order to measure the similarity of two photos in the *test set*, we used the computed probabilities as scores. More specifically, the computation of the similarity between two datapoints i and j was performed using the following formula

$$v_{\mathrm{ij}} = \sum_{a \in attrs} prob_{\mathrm{a}} \cdot M_{\mathrm{a}}(i,j)$$

, where *attrs* is the set of the metadata, $prob_a$ the associated scores from Table 1 and $M_a(i,j)$ is the *matching* function. For the majority of the attributes (Location, Username, Title, Tags) the M function is simply the indicator function $I_{\mathrm{a}}(i,j) = \begin{cases} 1 & i,j \text{ share the same value for attribute a} \\ 0 & \text{otherwise} \end{cases}$

In the case of the "Date Taken" attribute we used an exponential decay weight function to model the temporal proximity of two photo items. More specifically the similarity score was computed as $M_{\mathrm{ij}} = exp(-\frac{|d_{\mathrm{i}} - d_{\mathrm{j}}|}{h})$, where d_{i}, d_{j} are the timestamps of the two datapoints. The denominator in the exponent h is called the *bandwidth* and controls the rate of decay. We set this value to 1 hour, so that timestamps with time difference less than one hour will receive high values (close to 1) , while larger deviations are penalized more heavily.

3.2 Table Profiles

The second issue that we faced was the scaling of the algorithm in large datasets. Even for medium size datasets, like the one given for the task, it becomes impractical to measure the similarities among all pairs of datapoints. To tackle this problem, we introduced the concept of *table profiles*. A table profile is simply the union of all the photo items that "sit" in that table, and it is equivalent to a *super-photo item* that encompasses all the characteristics of the photos belonging to the table. Using this trick, we reduced significantly the number of comparisons needed to allocate a new "customer" (media item) from n (the total number of customers at that point) to K (the number of occupied tables)

3.3 Similarity-based CRP algorithm

This subsection finally presents our modified CRP algorithm. When a new customer (photo) comes in we measure its similarity with each one of the K already occupied tables and then we make a stohastic decision: The newcomer will either sit in one of K tables with probability analogous to their similarity value, or she will pick a new table.In short, the algorithm works as follows :

1. The first customer (photo) sits at the first table (event cluster) and initializes the first table profile.
2. For each of the subsequent customers we compute their similarity value with each of the K table profiles. We denote these values as v_k, k = 1,2..K
3. The customer sits at table k with probability $\frac{v_k}{\sum_i v_i + \alpha}$ or sits at a new table with probability $\frac{\alpha}{\sum_i v_i + \alpha}$. In the former case, the attributes of the media item are "merged" into the k-th table profile, while in the latter case a new table profile is initialized by the media item.

The parameter α controls the distribution of the customers on the tables. Higher values of α signify higher dispersion and thus, a larger number of occupied tables, while lower values of α give more compact allocations. Finally, to better measure the quality of our results, we computed the *purity*[1] of the generated clusters as $purity(\Omega, \mathbb{C}) = \frac{1}{N} \cdot \sum_k max_{\mathrm{j}}(\omega_{\mathrm{k}} \cap c_{\mathrm{j}})$, where where $\Omega = \{\omega_1, \omega_2, \ldots, \omega_K\}$ is the set of the clusters generated by the algorithm and $\mathbb{C} = \{c_1, c_2, \ldots, c_J\}$ is the set of the actual clusters.

4. RESULTS

The results were very sensitive to the selection of α. We performed a line search in the region [1,100]. We observed that for $\alpha > 10$ the algorithm diverged giving a huge number of clusters. For example, for $\alpha = 20$ we got about $50k$ clusters in the training set, while the actual number was about $15k$. Of course, the high purity value that we measured in this setting is meaningless, since the results are not well interpretable. For $\alpha < 10$, the algorithm generated a few and low-purity clusters. For example, for $\alpha = 5$, the algorithm gave about $3k$ clusters in the training set. We decided to set the value of α equal to 10, since it was a good compromise between the number and the purity of the generated event clusters in the training set. On the test set we achieved a F1(Main Score) of 0.2364 and NMI of 0.6644.

5. REFERENCES

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003.

[2] T. Reuter, S. Papadopoulos, V. Mezaris, P. Cimiano, C. de Vries, and S. Geva. Social Event Detection at MediaEval 2013: Challenges, datasets, and evaluation. In *MediaEval 2013 Workshop*, Barcelona, Spain, October 18-19 2013.

[3] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101, 2004.

[1] http://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-clustering-1.html