# A Compression Framework for Generating User Profiles

Xiaoxiao Shi
Department of Computer Science
University of Illinois at Chicago
xiaoxiao@cs.uic.com

Kevin Chang, Vijay K. Narayanan
Vanja Josifovski, Alexander J. Smola
Yahoo! Research
Santa Clara, CA 95051, USA
{klchang, vnarayan, vanjaj, smola}@yahoo-inc.com

## ABSTRACT

Predicting user preferences is a core task in many online applications from ad targeting to content recommendation. Many prediction methods rely on the being able to represent the user by a profile of features. In this paper we propose a mechanism for generating such profiles by extracting features that summarize their past online behavior. The method relies on finding a compressed representation of the behavior by selecting the dominant features contributing to the Kullback-Leibler divergence between the default distribution over user actions and the user specific properties. We show that the feature selection model of [1] can be extended to a hierarchical encoding of user behavior by means of using an intermediate clustering representation. Preliminary experiments suggest the efficacy of our method.

## 1. INTRODUCTION

In order to be able to reason about the user, search engines, computational advertising platforms, and recommendation systems require a suitable *representation* of the user. The profile is created based on the user's past behavior and such as to target the user with the suitable content or advertising. The targeting is usually performed by a model that predicts the user's interest. The key challenge in determining the right profile representation is to find predictive features. One alternative for this task is to use a generative methods, such as singular value decomposition of the (user, action) matrix; topical analysis by Latent Dirichlet Allocation [2]; clustering users based on their actions; or Probabilistic Latent Semantic Indexing factorization [4]. These methods describe the user in terms of some other representation (dense vectors, sparse sets of abstract topics, cluster ids). Often this suffices for the purpose of inference, since secondary algorithms do not require *interpretable* features.

In some cases, though, interpretability is important: advertisers *want* to know which users are being targeted by their ad campaigns. For instance some advertising contracts may specify that an ad be shown to males living in California, aged 18-25 years. We aim to obtain more fine-grained yet understandable representations. This problem is not limited to advertising. For instance, social scientists *want* to have understandable representations to support decisions.

We show that is possible to achieve both goals — to obtain interpretable features describing users in a human-understandable fashion *and* to obtain features which are good for predicting a user's actions. Our work builds on that of [1] who suggested a simple algorithm to determine meaningful features of users: choose features whose distribution differs most from the global baseline in terms of their Kullback-Leibler (KL) divergence. This retains an interpretable set of features, since it simply selects a subset of user actions 'pars pro toto'. While this rule was proposed in a slightly ad-hoc fashion, it is rather well founded: choosing the actions which contribute the most to the KL divergence between a given user and the sample average selects terms from the distribution which require the largest number of additional bits to encode. Note that the KL divergence $D(p\|q)$ quantifies the number of additional bits needed by encoding data drawn from $p$ with the code optimal for $q$.

However, a simple multinomial model as used by [1] is not necessarily ideal when it comes to representing the distribution over the actions of many users. Instead, we may choose more sophisticated formulations such as clustering, topics, or any of the other factorization approaches mentioned above. The simple KL divergence selection heuristic now becomes one of encoding the structure in a sparse fashion and subsequently one of encoding the user with regard to its latent representation. As a running example we use clustering for the latent representation. Hence, in order to represent a user's features we first pick a representation of a user's cluster in terms of sparse features and second we pick the user's actions relative to its cluster. This two-stage approach could be easily extended to many stages by feature extraction in hierarchical clustering. Likewise, in topic models [2] we could represent a user's action in terms of the actions most representative for the topics associated with the user and secondly with the actions which cannot be explained equally well by the topic model (and hence which have large KL divergence). Such multistage models effectively smooth out the actions of a user.

For instance, assume that one set of users would use the term 'car' and another set 'automobile'. While they clearly should belong to the same cluster, thus smoothing over minor differences in behavior, there clearly is additional information to be gained by recording the dichotomy between 'cars' and 'automobiles'.

## 2. ENCODING FRAMEWORK

### Sufficient Statistics

In the following we denote by $i$ users and by $x_i$ their actions. More specifically, $x_{ij}$ encodes action $j$ by user $i$. It is our goal to extract some features $\phi(x)$ from $x$ such that $\phi(x)$ leads a) to good predictions downstream and b) to understandable representation. First note that whenever $\phi(x)$ is a sufficient statistic of $x$ it follows that

$$p(y|x, \phi(x)) = p(y|\phi(x)). \tag{1}$$

In other words, given a good representation $\phi(x)$ of $x$ we can dispose of $x$ itself. This is particularly desirable since we can assume that $x$ is rather noisy, having been drawn as a sample from the distribution over all actions a particular user might take. Hence, if we find a more compact set of (sufficient) statistics which describe $x$ sans noise, we should be able to obtain equal or better estimation performance after applying $\phi(x)$.

### Kullback Leibler Divergence

Recall that the information contained in a stream of samples from some distribution $p$ is given by its entropy, that is by

$$H[p] := \mathbf{E}_{z\sim p}[-\log_2 p(z)] \tag{2}$$

One may show (see e.g. [3]) that the optimal code for encoding $z$ requires $-\log_w p(z)$ bits on average. In this view the Kullback Leibler divergence

$$D(p\|q) := \mathbf{E}_{z\sim p}[\log_2 p(z) - \log_2 q(z)] \tag{3}$$

quantifies the excess number of bits we spend on average by encoding $z \sim p$ with a code which is optimal for $z \sim q$. This provides a natural weight for symbols $z$ via $p(z)\log\frac{p(z)}{q(z)}$, the expected contribution of $z$ to the excess in encoding. In other words, if we want to reduce the excess in encoding $q$ we should focus on setting aside those terms $z$ with the largest contribution to the KL-divergence. This is precisely what the algorithm of [1] amounts to. Symbols $z'$ with small deviations in the encoding contribution are more likely to be due to sampling noise, hence we benefit from omitting them.

### Multinomial Distribution Model

The following example clarifies this rather abstract model. Assume that we have a multinomial distribution $p$ over symbols $z$, parameterized by probabilities $\theta_z$, i.e. $p(z) = \theta_z$. Such a distribution can be estimated, e.g. for a set of users, by the background model over all their actions: denote by $m$ the total number of actions of all users and let $n_z := \sum_{i,j}\{x_{ij} = z\}$ be the number of times any user takes action $z$. Then we can estimate

$$\theta_z = \frac{n_z}{m} \text{ or } \theta_z = \frac{n_z + \alpha}{m + N\alpha} \tag{4}$$

where the second equation is obtained by smoothing with a Dirichlet prior. Observing actions $x_{ij}$ for user $i$ we define the corresponding quantities $n_z^i := \sum_j \{x_{ij} = z\}$ and $m^i = \sum_z n_z^i$. This yields smoothed estimates for the action distribution for the user via $\theta_z^i = \frac{n_z^i + \alpha'}{m^i + N\alpha'}$ for a Dirichlet smoother $\alpha'$. When using the background distribution as quantified by $\theta$ rather than the user specific distribution $\theta^i$ we pay $d(p\|q|z) := \theta_z^i \log \theta_z^i/\theta_z$ additional bits for symbol $z$. Hence to minimize the amount of inefficiency in coding, we should store $\theta_z^i$ for all $z$ with $d(p\|q|z) \geq c$ for some threshold $c$. It yields understandable features, obviously provided that $z$ amounts to human-understandable actions or tokens.

### Mixture of Multinomials Model

A single multinomial distribution is not a very accurate characterization of user actions. One of the simplest modifications is to use a mixture of multinomials rather than a single multinomial. That is, we assume that observations follow the distribution

$$p(x) = \sum_y p(x|y)p(y) \tag{5}$$

where both $p(x|y)$ and $p(y)$ are multinomials (conveniently with an associated Dirichlet smoother). As a result the encoding problem now decomposes into two parts: encoding the *distribution* $p(x|y)$ and encoding the difference between $x_i$ and the estimated cluster $y_i$. We address this problem by applying the encoding strategy for multinomial distributions twice – once to encode the cluster distribution above the global baseline profile, and another time to encode the user distribution above the cluster profile. That is, for the cluster we select all $z$ which satisfy

$$d(p(\cdot|y_i)\|q|z) \geq c \tag{6}$$

with respect to the common baseline distribution $q$. Moreover, to encode $x_i$ we select all $z$ which satisfy

$$d(p^i\|p(\cdot|y_i)|z) \geq c \tag{7}$$

Note that (7) is slightly imprecise: since we did not encode $p(\cdot|y_i)$ entirely, it behooves us to compare the smoothed estimate of the user distribution $p^i$ *not* with $p(\cdot|y_i)$ but rather only with the *encoded* subset of tokens from $p(\cdot|y_i)$.

### General Framework

Given a model of the form

$$p(x) = \int p(x|\alpha)p(\alpha|\beta)p(\beta|\gamma)\ldots d\alpha d\beta d\gamma\ldots \tag{8}$$

we may resort to a hierarchical encoding by successively selecting $z$ which contribute most to the KL-divergence terms

$$D(p(x|\gamma)\|p(x)) \text{ or } D(p(x|\beta)\|p(x|\gamma)) \text{ or } \ldots D(p^i\|p(x|\alpha)).$$

## 3. TOKENS FOR USER ACTIONS

We assign tokens to each possible user event that we log. The tokens can be of varying granularity depending on the nature of the end application that uses this user profile. For instance, a user searching for "toyota prius" could be tokenized in a granular manner as "query for toyota prius" or in a coarser manner as "query related to autos".

We constructed a dataset from the events of a sample of 1 million users at Yahoo!, over the 55 day period from Jan 1, 2010 to Feb 24, 2010. We used events from 6 types of user activities: web page views (pv), views of display ads (adv), clicks on display ads (adc), search queries (sch), clicks on search results (slc) and clicks on search ads (olc). We tag each of these events with categories in a taxonomy of user interests, based on the content of the webpage, ad, or search query, through a combination of editorially labeled dictionaries and automated machine learned categorizers. We represent each event as the token"c<event-type>-<category>".

We then represent a user's history as a sequence of tokens of actions. For example, if the user had 1) a page view on autos.yahoo.com, followed by 2) a search query for "mutual fund", followed by 3) a click on a result of this search query, the sequence of tokens would be: "cpv-autos", "csch-finance", "cslc-finance". In all, there were a total of 5911 different unique tokens, constructed from the 6 event types and 1205 different categories for these events. We estimated the multinomial user model from this sequence of tokens.

# 4. EXPERIMENTS

We clustered the users in this dataset and estimated the average profile of users belonging to each cluster, and performed some preliminary experiments using our user profiles for the modeling task of ad conversion modeling.

## 4.1 Analysis of Clustering

Figure 1 shows the distributions of KL divergence between the user profile and the global profile and that between the user profile and the cluster profile. The distribution of the KL divergence between user and the cluster profiles is distinctively skewed towards lower values compared to the distribution of KL divergence between the user and the global profiles, demonstrating that clustering can effectively encode a large fraction of users. For example, more than 50% of the users have KL divergence in the range $[0, 0.7]$ between the user profile and the associated cluster profile, while less than 25% of the users have KL divergence in this range between the user profile and the global profile. Hence, a cluster profile is a more accurate representation of a user profile than the global profile. Figure 2 shows that clustering can still preserve groups of unique users which have high KL divergence from the global background, and does not wash out the profiles of users with distinctive profiles compared to the global profile.

Figures 1 and 2 show that clustering can effectively group together similar users by their behaviors, and also preserve groups of users with unique profiles compared to the global profile. Another interesting experiment is to understand the relevant features identified by the clustering encoding. Table 1 shows the user features in two different clusters that contribute most to the KL-divergence terms as described in (8). These features are the ones that most distinguish the cluster from the global profile.

It is clear that clustering can group together users with the similar latent interests represented by the tokens. For example, Cluster 1 consists of users with many technology related tokens, while Cluster 2 consists of users with many finance related and mobile phones related tokens.

Table 2 shows the most distinguishing tokens of 2 users selected at random from Cluster 1 and another user from Cluster 2. The highlighted tokens in bold letters show the features in the user profile that are also present in their respective clusters. Comparing the profiles of users 1 and 2, it is clear that while they share events related to technology, they are still different from each other in that user 1 has more events related to finance and sports, while user 2 has events related to beauty, personal care and babies. User 3 on the other hand has a number of events related to finance and cellular phones, and is also further interested in retail products. Thus, we observe that a) clustering effectively groups users with similar events in their profiles b) clustering-based encoding can judiciously capture the most

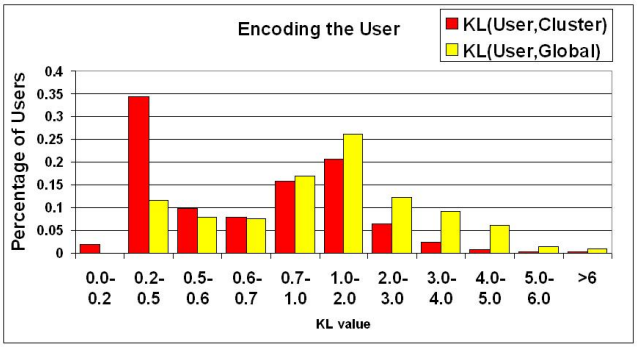


**Figure 1: Contribution to the KL divergence between user profile and global profile (yellow) and between user profile and cluster profile (red). Clustering is a better model for encoding the data (most users can be accurately captured by the model small KL divergence.**

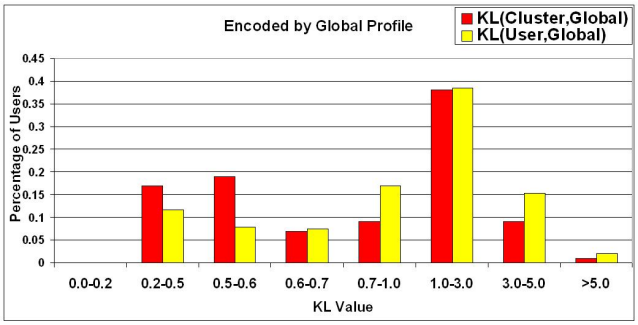


**Figure 2: Revealing unique user groups. Clustering can still preserve the groups of unique users with large KL divergence ($1.0 < KL < 3.0$)**

significant tokens among similar users; c) different clusters reveal different significant tokens. These three observations show the potential power of incorporating intermediate clustering representations to encode user profiles.

## 4.2 Preliminary conversion modeling results

We constructed a linear model to predict whether or not a user would convert on a specific ad, where we have one model per ad. The baseline model considered raw user events, while the experimental model considered raw user events as well as the user profile, constructed as user's cluster profile, and the 50 most divergent tokens between the user and the global profile, and the user and cluster profile. We evaluated the 191 models by computing the area under the ROC curve (AUC), on a test set that ocurred after the train set in time.

In order to compare the user profile-based models and the baseline across the 155 different ads/models, we tallied the "wins" as the number of ads for which the user profile-based models AUC exceeded the baseline models.

| # conv | # ads | % win | profile AUC | base AUC |
|---|---|---|---|---|
| $\geq 10$ | 155 | 62% | 0.57 | 0.56 |
| $[10, 100]$ | 112 | 68% | 0.55 | 0.53 |
| $\geq 100$ | 43 | 47% | 0.60 | 0.60 |

As an example to aid interpretation of the table, there were 112 ads with between 10 and 100 conversions in the

**Table 1: Important events in the cluster specific profile and KL divergence from the baseline. The instances show that user events are well grouped and human understandable. Cluster 1 shows a strong affinity for events related to technology, while Cluster 2 groups events in finance and cellular phone related tokens.**

| Cluster ID | Important User Behaviors | KL Divergence |
|---|---|---|
| Cluster 1 (Technology Group) | cpv-Technology/Internet Services | 0.3883 |
| | cpv-Technology/Internet Services/Online Community | 0.3854 |
| | cpv-Technology | 0.3840 |
| | cpv-Technology/Internet Services/Online Community/Email | 0.2829 |
| | cpv-Technology/Internet Services/Online Community/Portals | 0.2806 |
| | cpv-Technology/Internet Services/Online Community/Photos | 0.0122 |
| Cluster 2 (Finance Group) | cadv-Finance | 0.0365 |
| | cadv-Finance/Credit Services | 0.0274 |
| | cadv-Finance/Insurance | 0.0145 |
| | cadv-Finance/Insurance/Automobile | 0.0119 |
| | cadv-Telecommunications/Cellular and Wireless Services | 0.0081 |
| | cadv-Technology/Consumer Electronics/Comms/Mobile/Cellular Telephones | 0.0078 |

**Table 2: Profiles of 2 users randomly selected from Cluster 1 and one user from Cluster 2, with their event tokens and frequencies. The boldface tokens are common to the average profile of the relevant cluster.**

| User Index | Behavior set |
|---|---|
| User 1 Cluster 1 | **cadv-Technology/Internet Services/Online Community**:134<br>**cpv-Technology/Internet Services/Online Community**:190<br>**cpv-Technology/Internet Services**:192<br>**cpv-Technology**:192<br>**cpv-Technology/Internet Services/Online Community/Email**:190<br>**cadv-Technology**:150<br>**cadv-Technology/Internet Services/Online Community/Email**:132<br>**cadv-Technology/Internet Services**:142<br>cadv-Finance/Insurance:52<br>cadv-Finance:52<br>cpv-Sports/Soccer:21<br>cpv-Sports/Auto Racing:32 |
| User 2 Cluster 1 | **cpv-Technology**:212<br>**cpv-Technology/Internet Services/Online Community**:212<br>**cpv-Technology/Internet Services**:212<br>**cpv-Technology/Internet Services/Online Community/Email**:212<br>cadv-Consumer Packaged Goods:18<br>cadv-Consumer Packaged Goods/Beauty and Personal Care:14<br>cadv-Life Stages/Parenting and Children/Baby:10<br>cadv-Life Stages:10 |
| User 3 Cluster 2 | **cadv-Finance/Credit Services**:72<br>**cadv-Finance**:190<br>**cadv-Finance/Investment/Discount Brokerages**:44<br>**cadv-Technology/Consumer Electronics/Communication/Mobile/Cellular Telephones**:34<br>**cadv-Technology/Consumer Electronics/Communication/Mobile**:104<br>cadv-Small Business and B2B:44<br>cadv-Life Stages:32<br>cadv-Retail:24 |

test set used to compute the AUC. User profile models had a greater AUC than the baseline for 68% of these ads, and had an average AUC of 0.55 vs 0.53 for the baseline. We note that the user profile-based models perform better when there is less data. Models with little data tend to be more sensitive to noise, and therefore beneift from cluster-based user profiles that provide a smoother set of features.

# 5. CONCLUSION

We have proposed a method of extracting features from user profiles, based upon first clustering the users and then encoding the user profile as a combination of the cluster profile and the most distinguishing features between the user and this cluster profile. Experimental results show that the two stage approach of first clustering the user profiles, and then encoding the users with the cluster profile and the difference from this cluster profile, is a more effective method to encode a user profile, as opposed to a single step encoding of the users using the distinguishing features compared to the global profile alone. Preliminary modeling results show that the profiles can improve conversion modeling, especially when there are few positive examples in the data.

# 6. REFERENCES


[1] M. Shmueli-Scheuer, H. Roitman, D. Carmel, Y. Mass, and D. Konopnicki. Extracting User Profiles from Large Scale Data. In *PMDAC*, 2010.

[2] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *JMLR*, 3:993–1022, 2003.

[3] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley and Sons, NY, 1991.

[4] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1/2):177–196, 2001.