

# Categorization Using Semi-Supervised Clustering

Jianying Hu

Moninder Singh

Aleksandra Mojsilovic

*IBM T.J. Watson Research Center, 1101 Kitchawan Road, Route 134 Yorktown Heights, NY 10598*  
{jyhu, moninder, aleksand}@us.ibm.com

## Abstract

*Many applications require matching objects to a pre-defined, yet highly dynamic set of categories accompanied by category descriptions. We present a novel approach to solving this class of categorization problems by formulating it in a semi-supervised clustering framework. Text-based matching is performed to generate “soft” seeds, which are then used to guide clustering in the basic feature space. We introduce a new variation of the k-means algorithm, called Soft Seeded k-means, which can effectively incorporate seeds that are of varying degrees of confidence, while allowing for incomplete coverage of the pre-defined categories. The algorithm is applied to real-world data from a business analytics application, and we demonstrate that it leads to superior performance compared to previous approaches.*

## 1. Introduction

With the explosion of data volumes and Internet usage, many everyday applications rely on some form of automated categorization to avoid information overload and improve user experience. While matching of objects to predefined categories can often be carried out via supervised classification, there are many cases where classification is difficult due to inability to generate training examples for all categories. This is particularly true in situations where the pre-defined categories are highly dynamic due to constantly evolving applications and user needs. This makes it impossible to rely on manual labeling or sample selection, as they would have to be conducted whenever the taxonomy changes. In this paper we present a semi-supervised clustering methodology to address these challenges, and apply it in a real-life categorization task.

We consider categorization tasks where the following applies: 1) there is a dataset with underlying *basic features* attached to every point; 2) there is a set of categories, each accompanied with *category descriptions*; and 3) some or all of the data points also have textual

descriptions, generated independently of the category descriptions (e.g. outdated labels, or labels assigned without any predefined taxonomy in mind). We refer to these as *data descriptions*.

Note that this set of assumptions captures well the characteristics of many real world applications. For example, in a stock photo database, images are organized according to predefined categories, such as “Portraits” or “Macro”, with corresponding descriptions. Selected image features can be computed for all photos in the collection, forming the basic feature set, while some photographers enter the description of their photos, which then represent optional data descriptions. Another example, which we will use to develop and validate our methodology, is in the area of business analytics, and relates to categorization problems often encountered in project management tools. Such tools are used to track projects and compute business metrics according to a set of predefined categories aligned with products/services sold by the company. However, because of the dynamic business environments and changing customer needs, the solution portfolios are constantly evolving and frequently redefined, limiting the ability of project managers to categorize projects accurately. Hence, there is a need for an automated methodology to assist with project categorization.

We consider the latter application and formulate the categorization problem as a semi-supervised clustering one. Text-based matching between category and data descriptions is used to generate “soft” seeds, and guide clustering in the basic feature space using a new variation of the k-means algorithm called *Soft Seeded k-means*. We introduce a new metric for model selection in situations where the seeds do not necessarily cover all categories, and a novel *seed re-assignment penalty* measure to effectively make use of the text matching results with varying degrees of confidence.

Past research has proposed various methods to incorporate “constraint violation penalty” in clustering with pairwise constraints [7, 2], but not for cases using seeds with varying confidence levels in a k-means setting.

While semi-supervised clustering has been the focus of several recent projects [7, 1, 2], most of the prior studies were carried out using constraints that were artificially generated from true labels. Few applications with practical pair-wise constraints have been described in the past [7, 5]. To our knowledge, the current paper is the first to demonstrate the benefit of semi-supervised clustering using seeding in a real world application.

## 2. Project Categorization Using Soft-Seeded K-means

In the project categorization problem, each category is defined by a category name, along with category descriptions specifying the scope of the category. For example, a category could be “Server Product Services for Microsoft”, and the description may include “MS Application Development and Integration Services”, “MS Evaluation and Planning for On Demand” etc. For each project, the basic features consist of a skill allocation vector computed based on the actual hours billed by practitioners of various skills [4]. In addition, each project has an *optional* description field containing free text descriptions specifying the nature of the project, typically entered by the project manager at the beginning of the project.

In the first step of our approach, available project descriptions are matched against the category descriptions. For a subset of the data, this step produces a category label, along with a confidence score for each sample point.

In the second step, this set of labeled data along with the confidence scores is used as soft seeds in a semi-supervised clustering algorithm. We refer to the labeled set as “soft” seeds for two reasons: 1) The labels are not viewed as hard constraints on cluster membership, but rather soft ones with varying degrees of strength tied to the confidence score. 2) The seeds do not necessarily provide complete coverage of all categories. Note that in the latter case, semi-supervised clustering cannot be used to achieve completely automatic categorization (nor can any other learning scheme). However it still provides great value by reducing the amount of manual labeling required, since instead of having to label individual data points, one only needs to map the “un-covered” clusters to the “un-covered” categories.

We chose to use a k-means type algorithm because it has been successfully used in a variety of application domains. While the original algorithm was designed to operate in a completely unsupervised manner, several semi-supervised variations have been proposed recently to take into consideration “side information” in the form of both class labels and pair-wise constraints [7, 1, 2].

The new variation we have developed, Soft Seeded k-means, is most closely related to the Seeded k-means and Constrained k-means algorithms proposed by Basu *et. al.* [1]. The differences are that in Seeded k-means and Constrained k-means the seeds are either used only for initialization, or treated as hard constraints. Furthermore, both algorithms assume that there is at least one seed for each cluster. Our algorithm treats the seeds as “soft” constraints through the introduction of a seed re-assignment penalty, and allows incomplete coverage of clusters by seeds.

### 2.1. Seed Generation

Seeds are generated on the basis of the similarity between the project descriptions and the category descriptions, measured using the standard TF\*IDF (Term Frequency - Inverse Document Frequency) method [6]. For each category, the category descriptions are first processed for punctuation removal, stop word removal and stemming, abbreviation expansion (e.g. VOIP is expanded to Voice over IP, IPT is expanded to IP Telephony, etc.) as well as domain specific term mapping (e.g. AS400 is mapped to iSeries, System 390 is mapped to zSeries, etc.). The resulting descriptions are then tokenized into word-tokens using white space as delimiters. Finally, for each project description, the TF\*IDF similarity score is computed between that project description and each of the category descriptions. The similarity score provides a measure of the confidence in the mapping, with identical strings having a score of 1 and totally different strings having a score of 0.

A seed set can then be generated based on a chosen threshold on the confidence score  $T$ . For each project, if the maximum confidence score is greater than  $T$ , then the category corresponding to that score is assigned to the project instance; otherwise, the project is not assigned to any category. The set of projects labeled in this way (along with the corresponding scores) provide the “soft seeds” for clustering. Clearly, the choice of  $T$  determines the trade-off between the coverage and quality of the seed set: the higher the threshold, the “cleaner” the seed set, but the lower the coverage. As shown in the experiments later, when the seed labels are used as hard constraints the clustering outcome is highly sensitive to the selection of  $T$ . This sensitivity can be greatly reduced through the introduction of soft seeds.

### 2.2. Soft-Seeded K-means

The Soft Seeded k-means algorithm is summarized in Table 1. Here  $\mathcal{D}(x, y)$  is a pairwise distance mea-

**Table 1. Soft Seeded K-means Algorithm**

<p><b>Input:</b></p> <p><b>Data set:</b> <math>\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}, \mathbf{x}_i \in \mathbb{R}^d</math></p> <p><b>Seed label vector:</b> <math>\mathbf{L} = \{l_1, l_2, \dots, l_n\}, l_i = 0</math> if <math>\mathbf{x}_i</math> is unlabeled, and <math>l_i = j, j \in [1, \dots, k]</math> if <math>\mathbf{x}_i</math> is a seed assigned to cluster <math>j</math>.</p> <p><b>Seed score vector:</b> <math>\mathbf{S} = \{s_1, s_2, \dots, s_n\}, s_i \in [0, 1]</math>, where <math>s_i = 0</math> if <math>\mathbf{x}_i</math> is not a seed.</p> <p><math>k</math>: total number of clusters.</p> <p><math>q</math>: number of clusters covered by seeds. Without loss of generality, assume the first <math>q</math> clusters are covered.</p> <p><b>Method:</b></p> <ol style="list-style-type: none"> <li>1. Iterate <math>m = 1 : M</math> <ol style="list-style-type: none"> <li>(a) Initialize centroids <math>\xi_1, \dots, \xi_q</math> using labeled data points.</li> <li>(b) Initialize centroids of the remaining clusters through random sampling of unlabeled data points.</li> <li>(c) For each data point <math>\mathbf{x}_i</math>, assign it to the cluster <math>j^*</math> that minimizes: <math>\mathcal{D}(\mathbf{x}_i, \xi_j) + \mathcal{P}(j, l_i, s_i)</math>.</li> <li>(d) For each cluster <math>C_i</math>, update its centroid using current assignments</li> <li>(e) Iterate between (c) and (d) until convergence (i.e., no change of membership take place)</li> <li>(f) Store resulting partition and its <i>model fitness score</i> <math>\mathcal{F}_m</math>.</li> </ol> </li> <li>2. Return the partition with the highest score <math>\mathcal{F}_m</math>.</li> </ol>
---

sure defined over  $\mathbb{R}^d$ , and  $\mathcal{P}(j, l_i, s_i)$  is the seed re-assignment penalty measure defined as following:

$$\mathcal{P}(j, l_i, s_i) = \begin{cases} 0 & \text{if } j = 0 \text{ or } j = l_i \\ \frac{\gamma}{1 + e^{(-\alpha(s_i - \beta))}} & \text{otherwise} \end{cases} \quad (1)$$

In other words, the penalty is 0 if a data point is either not a seed, or it is a seed assigned to the same cluster as indicated by the seed label. Otherwise, the soft seeding constraint is violated and the penalty incurred is a sigmoid function of the seed score  $s_i$ . The sigmoid implements a soft stepping function with values in the range  $(0, \gamma)$ . The middle point of the soft step is defined by  $\beta$  and the slope of the step is controlled by  $\alpha$ . In general,  $\gamma$  should equal the maximum pairwise distance, and a reasonable value for  $\beta$  is the mean of the seed confidence scores.

As in any k-means style algorithm, the inner loop of the algorithm only converges to a local minimum, and thus depends heavily on centroid initialization. For clusters covered by seeds, the initial centroids are estimated using the seeds. For the remaining clusters no such side knowledge exists, and we are faced with the same initialization challenge as in unsupervised k-means clustering. It has been shown that one of the most

effective methods of getting better initialization in this case is to perform multiple runs with different random initializations, followed by model selection [3]. However, an important question is how to define the model fitness score.

In unsupervised k-means, the model fitness score is simply the inverse of the overall distortion. However, in semi-supervised clustering, such a function is not suitable. Intuitively, in this setting the best partition is not necessarily the one with minimum overall distortion, but rather one that conforms best to the seed constraints while maintaining low overall distortion. Thus, in our algorithm the fitness score is defined as:

$$\mathcal{F}_m = \frac{1}{\mathcal{D}_m^l * |\mathcal{D}_m^l - \mathcal{D}_m^u|}, \quad (2)$$

where  $\mathcal{D}_m^l$  and  $\mathcal{D}_m^u$  are the overall distortion of the labeled and unlabeled data points, respectively. The first term favors small distortion over the labeled data points (indicating better conformity to seed constraints), while the second ensures that the distortion is small over the un-labeled data points as well. Empirical evidence shows that this fitness score leads to significantly better performance than one based on the overall distortion. A reasonable refinement would be to incorporate weights to  $\mathcal{D}_m^l$  and  $\mathcal{D}_m^u$  which can be tuned on the training set. This and other possible enhancements are to be explored in our future work.

### 3. Experiments

Experiments were conducted using real data collected from the Integrated Technology Service division of IBM and validated by domain experts. The data set contains 302 projects from the Server Services Product Line, belonging to 8 predefined categories. Each project is associated with a 67 dimensional skill allocation vector. The pairwise vector distance  $\mathcal{D}(\mathbf{x}, \mathbf{y})$  is computed using  $L_1$  norm. For results reported in this paper, the parameters used to compute the seed re-assignment penalty (Eqn. 1) are set to:  $\alpha = 10$ ,  $\beta = 0.4$ , and  $\gamma = 2.0$ . The number of random initializations  $M$  is set to 10.

We generated 8 different sets of seeds by setting the threshold at levels equally spaced between 0.8 and 0.1, covering the range of all non-zero text matching scores on this data set. Table 2 shows the coverage and accuracy of the different sets of seeds. Note that in this case, even when the threshold is set at the lowest level, the seeds still do not provide complete coverage of the clusters.

Following the seed generation step, we ran the proposed Soft-Seeded k-means algorithm (SSk-mean) and

**Table 2. Seed Coverage and Accuracy at Different Confidence Levels**

Conf.	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
% of total	9	10	31	32	33	40	43	59
Coverage	1	2	4	5	5	7	7	7
Accuracy	100	100	100	99	98	96	89	78

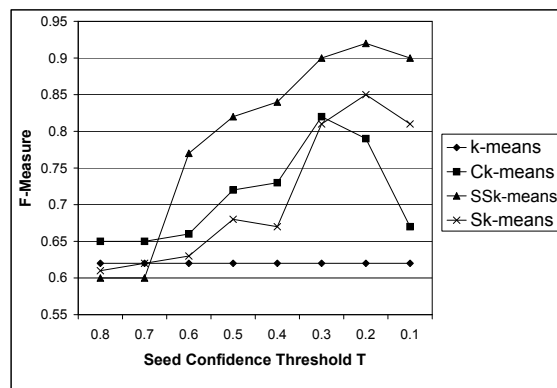
compared against standard k-means (k-means), Seeded k-means (Sk-means) and Constrained k-means (Ck-means) [1], for all seed sets. To evaluate the outcome of each clustering algorithm, we used the pairwise F-Measure [2]. Each algorithm was run 10 times at each seed setting, and the average F-Measure is reported. For all algorithms the number of random initializations was set to 10. In the case of Seeded and Constrained k-means, since the original algorithm assumed complete coverage, to make it applicable here we used random initialization for the uncovered clusters, followed by standard model selection using overall distortion, which appears to be consistent with the setting used for the experiments reported in [1].

Results are shown in Fig. 1. As seen in the plot, the proposed Soft Seeded K-means algorithm significantly outperforms both Seeded k-means and Constrained k-means, for all but the first two seed sets. For the first two seed sets, the seeds provide such a limited coverage (only up to 2 out of the 8 clusters) that none of the semi-supervised algorithms could benefit from them.

In all semi-supervised clustering algorithms compared here one can observe, to different degrees, the trade-off between the benefit of including more seeds, thus incorporating more side knowledge, and the drawback of allowing more noise in the seeds. When the seed confidence levels are relatively high, resulting in high seed accuracies, the inclusion of more seeds improves the clustering performance. When the seed accuracy becomes much lower, the performance starts to decrease again. Constrained k-means outperforms Seeded k-means at high confidence levels, however its performance drops sharply as the seed accuracy gets below 95%. On the other hand, the new Soft Seeded k-means algorithm is able to achieve better performance at all levels. These results indicate that the soft seeding mechanism allows for the combination of the best characteristics of the previous algorithms – it is robust against seed noise while at the same time makes most effective use of the constraints introduced by the seeds.

## 4. Conclusion

We have presented a novel solution to a class of categorization problems using a semi-supervised cluster-



**Figure 1. Performance Evaluation Results**

ing approach. We introduced a novel variation of the k-means algorithm called Soft Seeded k-means, which makes effective use of the side information provided by seeds with a wide range of confidence levels, even when they do not provide complete coverage of the pre-defined categories. This large degree of flexibility, which is critical in making semi-supervised clustering work in practice, is achieved through the introductions of a novel seed re-assignment penalty measure and model selection measure. Experiments using real world data demonstrate significant improvements over previous algorithms. While the approach is described in the context of the service product categorization problem, it can be easily applied to problems of similar nature in other domains.

## References

- [1] S. Basu, A. Benerjee, and R. Mooney. Semi-supervised clustering by seeding. In *Proc. 19th ICML*, Sydney, Australia, July 2002.
- [2] M. Bilenko, S. Basu, S. Basu, and R. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *Proc. 21st ICML*, Banff, Canada, 2004.
- [3] U. Fayyad, C. Reina, and P. Bradley. Initialization of iterative refinement clustering algorithms. In *Proc. 4th ACM KDD*, Menlo Park, CA, 1998.
- [4] J. Hu, B. Ray, and M. Singh. Statistical methods for automated generation of service engagement staffing plans. *IBM J. of Research and Development*, March 2007.
- [5] Z. Lu and T. Lean. Semi-supervised learning with penalized probabilistic clustering. In *Proc. 18th NIPS*, 2004.
- [6] G. Salton and C. Buckley. *Term Weighting Approaches in Automatic Text Retrieval (Technical Report 87-881)*. Dept. of CS, Cornell Univ., Ithaca, NY, USA, 1987.
- [7] K. Wagstaff, C. C., R. S., and S. S. Constrained k-means clustering with background knowledge. In *Proc. 18th ICML*, Massachusetts, US, July 2001.