A Framework for Mixed-Initiative Clustering

Yifen Huang HYIFEN@CS.CMU.EDU
Language Technologies Institute, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213 USA

Tom M. Mitchell TOM.MITCHELL@CS.CMU.EDU
Machine Learning Department, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213 USA

Abstract

Mixed-initiative clustering is a machine learning task that integrates a machine's clustering capability and a user's guidance in order to obtain the user's desired result. This task is different from traditional autonomous clustering tasks by introducing user criterion, a user's understanding of data and purpose of sorting. We propose a framework for mixed-initiative clustering to solve problems such as how to model a user's criterion and how to utilize the criterion to improve the clustering performance. Our approach consists of representing properties of the clustering model to a user and letting the user gives feedback on these properties. We also demonstrate the feasibility of this framework using an application called "activity extraction from personal workstation contents." Our work includes building structured activity descriptions to represent a machine's speculation and a new clustering algorithm, the SpeClustering model, that enables extended user feedback. Furthermore, we identified the SpeClustering model as an instantiation of mixed-initiative clustering.

1. Introduction

In most machine learning literature, clustering is investigated as an analysis tool to find interesting data associations. In this paper, I propose to explore a new clustering task where its data can also be organized by a user, but the task is too tedious to be performed manually. The goal of this clustering task is not only finding interesting data associations, but also achieving a user's desired result. In other words, the task tries to help the user configure data according to their own criterion.

A user's criterion is constituted by a user's understanding of data and purpose of this sorting task. We use a simple picture example to illustrate the concept of user criterion. Figure 1 contains a set of pictures and the task is to sort these pictures into two groups. The user Mary's criterion about this sorting task is animal species so her desired clustering result is to put cats into one group and dogs into another group. Figure 1(a) shows her desired result. The user Joe's criterion is to separate real animals to animated ones. Figure 1(b) shows his desired result. Even the data is the same, according to different user criteria, the desired results will not be the same. In fact, there is no optimal clustering result for every user but there is a desired result for each user. Furthermore, Figure 1(c) shows a possible autonomous clustering result generated by a computer on this same set of pictures. The clustering result is different from either Mary's or Joe's desired results. We call the difference between machine's result and user's result "clustering mismatch".

Since a user has a golden criterion on how the data can be organized, it is intuitive to involve the user into the clustering process. The issue is how to incorporate a computer and a user so they can achieve the user's desired result jointly. According to the simple picture example, we can identify two important problems that distinguishes this new clustering task from a traditional autonomous clustering task. The first problem is how to model a criterion in a clustering algorithm. The second problem is how to rectify a mismatched clustering result (and model) according to the user's golden criterion.

We propose a framework to solve these two problems. Figure 2 depicts this framework in graphics. There are

Presented at *North East Student Colloquium on Artificial Intelligence (NESCAI)*, 2007.

two agents, a computer agent and a user agent. Individually, a computer agent has capabilities of analyzing data; a user agent possesses the golden criterion of clustering but may not be aware of their own criterion in an explicit way and cannot express the criterion in a computer understandable format. We first define two languages for these two agents to communicate with each other so the computer agent can learn the criterion from the user and the user can utilize the computational power on analyzing dada. We then define a learning process to incorporate both agents' efforts. We call this framework "mixed-initiative clustering" because both agents contribute to the clustering process interactively.

The remaining paper is organized as follows. We will describe our framework in details in the next section. Section 3 provides an overview over mixed-initiative systems. Section 4 introduces the application on workstation activity extraction. Section 5 briefly describes our approaches and results on the application. At the end of the paper, we will conclude and discuss future research directions.

2. Framework

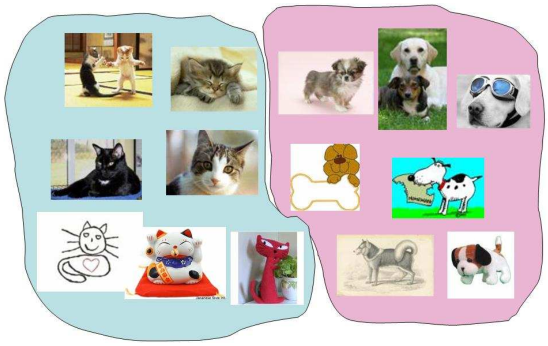
A mixed-initiative clustering framework combines both agents' advantages in order to achieve the desired clustering result more efficiently. We define two languages used for communication between two agents and a learning process of mixed-initiative clustering.

We first discuss the communication from the computer to the user. Since the hypothesized function, $\hat{f} : X \to Y$, is too complex to represent to a user, for example, "the probability of this word given this cluster is 0.0005" doesn't help the user to understand $\hat{f}$ at all, we represent a set of hypothesized properties, $H_{c \to u}$ in Figure 2, of the function instead.

Definition $L_{c\to u}$ is the computer's language to the user that defines admissible hypothesized property types in a mixed-initiative clustering task. A hypothesized property is admissible if it can be interpreted by a clustering model probabilistically and be understood by a user. $H_{c\to u}$ is a set of computer-generated hypothesized properties of the hypothesized function, where each hypothesized property in $H_{c\to u}$ belongs to one admissible property type defined in $L_{c\to u}$.

To clarify the notation, the language $L_{c\to u}$ is similar to a grammar and $H_{c\to u}$ is similar to a set of legit sentences according to the grammar.

Example Let $X^l = \langle X_1^l, \cdots, X_M^l \rangle$ indicate the l^{th} instance. $\widehat{Y}^l$ is the clustering prediction using the



(a) Mary's criterion is animal species.

(b) Joe's criterion is to separate real animals to animated ones.

(c) A possible clustering result generated by a computer on this same set of pictures.

Figure 1. An example (sorting a set of pictures into two categories) of different user criteria and clustering mismatch.

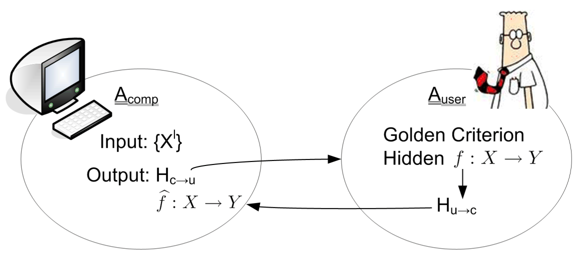


Figure 2. The communication between two agents plays an important role in mixed-initiative clustering.

trained clustering model, $\widehat{f}(X^l) = \widehat{Y}^l$. The user's criterion, $f(X^l) = Y^l$, is a hidden function. Here are some examples of admissible hypothesized property types which $L_{c\to u}$ may include:

- **hypothesized instance-label property** The prediction of an instance is correct. $\widehat{f}(X^l) = f(X^l)$.
- **hypothesized pair-wise must-link property** If the predictions of two instances are the same, they belong to the same cluster according to the user criterion as well. $\widehat{f}(X^l) = \widehat{f}(X^{l'}) \Rightarrow f(X^l) = f(X^{l'})$
- **hypothesized cluster-existence property** If a cluster is generated by the clustering model, this cluster exists in the user's criterion, too.
- **hypothesized feature-label property** If a feature X_m is a key feature (based on a feature selection measurement) for the computer agent's cluster $\widehat{f}(X) = y$, X_m is also a key feature for the user with regards to cluster $f(X) = y$, too. From the user's point of view, X_m is a key feature for cluster y if the existence of X_m in an instance gives high evidence that the instance belongs to y.

Secondly, We discuss the communication from the user to the computer.

Definition $L_{u\to c}$ is the user's language to the computer that contains all admissible user feedback types for each admissible hypothesized property type in $L_{c\to u}$. A user feedback type is admissible if it matches the following two requirements: (1) its property type can be presented in an interface and a user can give this type of feedback on these properties through the interface, (2) feedback on a hypothesized property can be represented as another property. $H_{u\to c}$ collects a user's feedback regarding a subset of hypothesized properties in $H_{c\to u}$.

Example Let h be a hypothesized property in $H_{c\to u}$. Here are some possible user feedback types for $L_{u\to c}$:

- **approval feedback** A_{user} agrees with h. The approved property is annotated as h^+.
- **disapproval feedback** A_{user} disagrees with h. The disapproved property is annotated as h^-.
- **modification** A_{user} edits h. We annotate the modified property as h^m.
- **insertion** A_{user} adds a property, h^i.

With languages between two agents defined, we can now define the learning process of mixed-initiative clustering.

Definition A *mixed-initiative clustering* task consists of unlabeled data $\{X^l = \langle X_1^l, \cdots, X_M^l \rangle, l = 1, \cdots, D\}$ and two agents, A_{user} and A_{comp}. It is a machine learning task that learns (a) the hypothesized clustering function $\widehat{f} : X \to Y$, where Y indicates the cluster index of X, and (b) the hypothesized properties, $H_{c\to u}$, that represent both the clustering function and the hidden user criterion. The mixed-initiative clustering learns the clustering function and the properties through communications, $H_{c\to u}$ and $H_{u\to c}$, between A_{user} and A_{comp} using the languages $L_{c\to u}$ and $L_{u\to c}$. Below is the detailed process:

1. Initialize $H_{u\to c} = \{\}$.
2. A_{comp} builds an initial clustering model: $\widehat{f} : X \to Y$.
3. Based on $\{X^l\}$ and $\widehat{f}$, A_{comp} generates a set of hypothesized properties $H_{c\to u}$ to be presented to A_{user}. Each hypothesized property, h, in $H_{c\to u}$ shouldn't contradict $H_{u\to c}$.
4. A_{user} gives feedback, $\{h^*\}$, on a subset of $H_{c\to u}$. $H_{u\to c} \leftarrow H_{u\to c} \cup \{h^*\}$.
5. Based on X and $H_{u\to c}$, A_{comp} re-trains the clustering model, $\widehat{f} : X \to Y$.
6. Repeat step 3 to step 6 until user's satisfaction.

The communication between two agents plays an important role in mixed-initiative clustering. A computer analyzes input data and generates hypothesized properties, $H_{c\to u}$, which consists of machine's speculation on the hidden user criterion. A user gives feedback on hypothesized properties according to their hidden golden criterion. The feedback, $H_{u\to c}$, represents true properties of the hidden user criterion, $f : X \to Y$. For each re-training of $\widehat{f} : X \to Y$ according to $H_{u\to c}$, we hope $H_{c\to u}$ will become more and more representative of the hidden user criterion as well.

3. Related Work

A mixed-initiative system is a system where machine's automation and user's guidance collaborate in order to achieve the user's goal.

Some studies of mixed-initiative systems begin from the realm of human-computer interaction. (Horvitz, 1999) describes the principles of mixed-initiative user interface and provides a utility function to decide the invocation of machine's automation according to user's attention cost and automation benefits. Planning can also be designed as a mixed-initiative system, for example, a user can navigates through machine's planning scripts (Amant, 1997).

Other studies of mixed-initiative systems begin from the realm of machine learning. (Kerne et al., 2005) organizes information elements in a spatial hypertext space which a user can adjust by choosing cluster anchors and setting relative importance of features through a pie chart. Socially guided machine learning studies how to represent robot's confident level as a learner through robot's gaze behaviors (Thomaz & Breazeal, 2006), what is a person's behavior as a teacher in an interactive robot training process and how a machine can modify its reinforcement learning algorithm to work with its teacher more effectively(Thomaz et al., 2006).

Semi-supervised learning techniques are highly related to mixed-initiative systems, (Huang & Mitchell, 2006) contains related work in this area.

4. Activity Extraction from Workstation Contents

The goal of activity extraction from workstation contents is to build an activity-based environment. We use an example to describe the activity-based environment[1]: A user, Karen, is in charge of weekly newsletter editing for her department. She checks the websites of three subcontractors and pastes information from those sites into her Word document. Then she pulls up recent emails from Sharon because Sharon is the coordinator of the department meetings. Karen summarizes important information from Sharon's emails into the newsletter. Once the intelligent assistant learns her newsletter editing activity, it can prepare websites Karen need to visits, emails she will summarize and the newsletter template for her. Knowing what task/project the user is working on is a key component of an intelligent assistant so that it can integrate information from various software applications for the user.

The content of a user's workstation ranges from official, highly user-involved work to personal-defined, less salient involvements, so we use the terminology **activity** for any routines a user performs on her own workstation. A user has the freedom to define her activities flexibly, for example, an official project can be an activity and checking snow conditions of a nearby ski resort can be another activity, too.

[1]This example is a re-write of "A Day in the Life of CALO: 2008" by Michael Chorost.

5. Approaches

Our approaches to inferring users' activities from workstation contents include the ActivityExtractor system (Mitchell et al., 2006) which focuses on generating user understandable activity descriptions and the SpeClustering model (Huang & Mitchell, 2006) which focuses on model adaptation according to extended user feedback.

5.1. ActivityExtractor

ActivityExtractor consists of three components which (a) cluster emails based on analysis of their text content, (b) perform social network analysis on each cluster to potentially subdivide it into subclusters, and (c) construct a structured representation of each cluster (activity), associating calendar meetings and person names with the activity. The system also outputs an email classifier which can be used to incrementally classify future emails into the corresponding activity. Refer to (Mitchell et al., 2006) for details of algorithms in ActivityExtractor. For general text clustering tasks, keywords constitute a reasonable description for each cluster.

An activity description produced automatically by ActivityExtractor is shown in Figure 1. This description is the verbatim output of ActivityExtractor, except that we have removed person names from the list of keywords to improve readability, and we have removed phone numbers from the meeting descriptions for privacy reasons. The user recognized this cluster description because the activity description represented here corresponds to one of the research projects of this workstation's user.

The quality of activity descriptions reflects not only the performance of the ActivityExtractor system but also how close these machine generated descriptions are to the hidden user criterion. On a qualitative evaluation, a user found each cluster was related to one or more of of his ongoing activities (e.g., a committee, family email, etc.), but that each cluster also contained 20-50% of extraneous emails not affiliated with the main activity of the cluster. Despite this non-homogeneity in the clusters, the post-processed descriptions of each cluster often produced quite reasonable descriptions of its dominant activity.

Keywords and persons are the most recognizable parts in structured descriptions and we can formalize them as hypothesized feature-label properties. However,

- **Activity Name**: CALO ActivityExtractor
- **Keywords (omitting person names)**: ActivityExtractor, TFC, IRIS, clustering, heads, emails, collected, clusters, SRI, ...
- **Person Names**: Adam Cheyer (0.49), Ray Perrault (0.36), Hung Bui (0.32), Melissa Beers (0.30), James Arnold (0.28), Jack Park (0.26), Sophie Wang (0.25), Tomas Uribe (0.25), jeanne ledbetter (0.24), Leslie Pack Kaelbling (0.24), . . .
- **Meetings**: CALO TFC telecon (0.59), CALO phone call (0.55), SRI Meeting - Chicago (0.48), SRI TFC Telecon and quarterly rpt (0.47), SRI visit. Bill and Ray. Call Melissa Beers when arriving (0.47), CALO annual mtg at SRI (0.45), . . .
- **Primary Senders**: tom.mitchell@cmu.edu (75), sophie.wang@cs.cmu.edu (20), adam.cheyer@sri.com (16), perrault@ai.sri.com (14), . . .
- **Primary Recipients**: tom.mitchell@cmu.edu (94), adam.cheyer@sri.com (40), william.cohen@cs.cmu.edu (35), perrault@ai.sri.com (19), . . .
- **Emails**: [email125], [email72], . . . (245 emails in total)
- **User Activity Fraction**: 245/2822=0.086 of total emails
- **User Involvement**: user authored 30% of email (default 31%)

Figure 3. An example of an activity description created automatically from clustering results. Real-valued numbers in parentheses indicate confidences that the corresponding person or meeting is associated with this activity. Integers in parentheses indicate the number of occurrences of the entity within the 245 emails associated with this cluster.

a multinomial Naive Bayes clustering model (Nigam et al., 1998) cannot interpret these properties probabilistically so they are not admissible property types for the model. In other words, even the user gives feedback on these properties, the multinomial model cannot adjust its parameters accordingly without resorting to heuristics. We then develop the SpeClustering model where they hypothesized feature-label property type is admissible in the mixed-initiative clustering framework.

5.2. SpeClustering and Extended User Feedback

In order to represent a subset of key features, we use an idea that is similar to topic models (Blei et al., 2003). We assume there is a specific topic for each cluster and there are some general topics. A document is generated by a mixture of a specific topic and general topics. Intuitively, a specific topic will contain a subset of key features for a cluster and general topics will contain other features that don't affect cluster prediction.

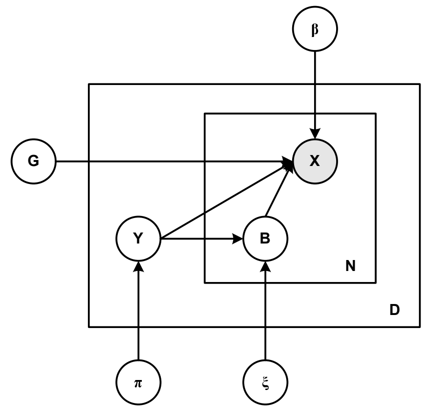


Figure 4. Graphical representation of SpeClustering model. Y is a variable representing the cluster associated with a document, X represents an observed word in a document, and B is a boolean variable that indicates whether word X is generated conditioned on the cluster Y or whether it is generated according to a cluster-independent general distribution of words G.

To construct this SpeClustering model, we extend the standard multinomial model in two ways. The first modification is to add a G topic variable that is intended to capture general topics not related to the cluster. The second modification is to introduce a hidden boolean variable, B, associated with each word X in each document. If $B = 1$, the observation X is generated by the cluster-specific topic Y, and if $B = 0$, the observation X is generated by a general topic G. Throughout this paper we simplify the model by assuming there is only one general topic instead of multiple topics, so the value of G is fixed at $G = g$. Figure 4 shows the graphical model representation of the model. Here the outer rectangle (or plate) is duplicated for each of the D documents, and the inner plate is duplicated for each of the N observations X and associated variables B. Note the general topic G is constant across all documents and words, whereas the cluster topic Y is different for each document.

We apply bag-of-word representation for document X^l, so it can be represented as a sequence of observations $\{X^l_{(i)}; i \in \{1, 2, ..., N(l)\}\}$ or as a feature vector $\langle X^l_1, \cdots, X^l_M\rangle$. The Speclustering model θ has four sets of parameters:

$$\pi_c = P(Y = c)$$

$$\xi_c = P(B = 1|Y = c)$$

$$\beta_{cm} = P(X = X_m|Y = c)$$

$$\beta_{gm} = P(X = X_m | G = g)$$

where $c \in \{1, 2, ..., |Y|\}$, $g \in \{1\}$ for the simplified case. Please refer to (Huang & Mitchell, 2006) for details of model training.

The SpeClustering model provides the probability interpretations for hypothesized cluster-existence properties (parameter π_c), instance-label properties (Eq. 1), and feature-label properties (Eq. 2) so we can include all these hypothesized property types in $L_{c\to u}$.

$$\begin{aligned} \phi^l(c) &\equiv P(Y^l = c|X^l; \theta) \\ &\propto \pi_c \prod_{m=1}^{M} [\xi_c \beta_{cm} + (1 - \xi_c)\beta_{gm}]^{X^l_m} \qquad (1) \end{aligned}$$

$$\begin{aligned} \psi_m(c) &\equiv P(B_m = 1|Y = c, X = X_m; \theta) \\ &= \frac{\xi_c \beta_{cm}}{\xi_c \beta_{cm} + (1 - \xi_c)\beta_{gm}} \qquad (2) \end{aligned}$$

With admissible hypothesized property types defined, we can define admissible feedback types for each hypothesized property type. The allowed user feedback are:

- CR: disapproval feedback on the cluster-existence property
- PP: approval and disapproval feedback on the instance-label property
- WX: insertion, approval and disapproval feedback on the feature(word)-label property
- HX: approval and disapproval feedback on the feature(person)-label property

Each feedback on a hypothesized property gives the model the true value or the constraint of the probability that associated with the hypothesized property. The SpeClustering model can thus re-train its parameters by adjusting probabilities during the EM process according to user feedback.

Figure 5 summarizes this process of integrating user feedback into the SpeClustering model. It is an example of mixed-initiative clustering defined in Section 2 with thoroughly defined $L_{c\to u}$, $L_{u\to c}$ and a feedback adaptation algorithm.

5.3. Experiments

5.3.1. Dataset and Measurement

To test the SpeClustering algorithm, we used an email dataset (*EmailYH*) which contains 623 emails. It had previously been sorted into 11 folders according to the user's activities. It contains 6684 unique words and 135 individual people after pre-processing[2].

[2]The pre-processing for words includes stemming, stop word removal and removal of words that appear only once in the dataset. The pre-processing for people contains reference-reconciliation over email senders and recipients, and removal of people that are involved in only one email.

Algorithm: Mixed-Initiative SpeClustering
Input: Unlabeled corpus $\mathbb{C}$
Output: A clustering model and a list of activity descriptions, $H_{c\to u}$.
Languages: $L_{c\to u}$ consists of the hypothesized instance-label property type, feature-label property type, and cluster-existence property type. $L_{u\to c}$ consists of approval/disapproval feedback types for instance-label properties, approval/disapproval/insertion feedback types for feature-label properties, and approval/disapproval feedback type for cluster-existence properties.

Method:

1. Initialize $H_{u\to c} = \{\}$.
2. A_{comp} builds an initial model $\widehat{f}$.
3. Based on $\mathbb{C}$ and $\widehat{f}$, A_{comp} generates activity descriptions, $H_{c\to u}$, where the feature-label properties consist of top K features of each cluster.
4. A_{user} gives feedback, $\{h^*\}$, on a subset of $H_{c\to u}$. $H_{u\to c} \leftarrow H_{u\to c} \cup \{h^*\}$.
5. A_{comp} performs $\widehat{f}^{new}$ =SpeClustering-with-Feedback($\mathbb{C}, \widehat{f}, H_{u\to c}$).
6. $\widehat{f} = \widehat{f}^{new}$. Repeat step 3 to 6 until user's satisfaction.

Algorithm: SpeClustering-with-Feedback
Input: Unlabeled corpus $\mathbb{C}$. $\widehat{f}$ as the current model. $H_{u\to c}$ as the collection of user's feedback.
Output: $\widehat{f}^{new}$ as the model after adaption according to user's feedback.

Method:

1. $\widehat{f}^t = \widehat{f}$.
2. Estimate probabilities $\mathcal{P}^t$ of Eq 1 and Eq 2 given $\mathbb{C}$ and $\widehat{f}^t$.
3. Adjust $\mathcal{P}^t$ according to $H_{u\to c}$ to obtain $\mathcal{P}^t_{adj}$.
4. Re-estimate model parameters using $\mathcal{P}^t_{adj}$ to obtain $\widehat{f}^{t+1}$.
5. $\widehat{f}^t = \widehat{f}^{t+1}$; repeat step 2 to 5 until the model converges.
6. $\widehat{f}^{new} = \widehat{f}^t$.

Figure 5. Summarization of the mixed-initiative SpeClustering clustering.

run	doc #	CR	PP	WX	HX
Email1	623	3	99	37	30
Email2	623	3	73	35	31
Email3	623	4	92	48	26
Email4	623	7	32	28	15
Email5	623	4	91	43	28

Table 1. Entry numbers of different feedback types for 5 runs.

We use folder-reconstruction accuracy to estimate cluster quality. To calculate the folder-reconstruction accuracy, we search through all possible alignments of cluster indices to folder indices in order to find the alignment resulting in optimal accuracy, then report the accuracy under this optimal alignment.

5.3.2. User Interface and Feedback

The user gave feedback using the interface shown in Fig 6. The top left panel shows a list of documents that are clustered into the selected cluster label, the top right panel shows 5 key-persons of the cluster and the bottom right panel shows 20 keywords of the cluster. The keywords and key-persons of the cluster are selected using a Chi-squared measurement (Yang & Pedersen, 1997). When a user clicks on a document in the document list, the content of the document shows in the bottom left panel. The user can give various types of feedback described in Section 5.2 and the interface displays feedback the user has entered so far. The user can also go back and forth to correct conflict assumptions she has made to achieve consistent cluster interpretations.

An interesting observation we found is that displaying keywords and key-persons tremendously helps the user make judgements about a cluster. In fact, to decide the meaning of a large cluster based only on examining the documents is extremely difficult. A user would tend to decide based on the first several documents she goes through even when the cluster contains more than hundreds of documents, and the biased decision often causes conflicts with later clusters. The user usually chooses to remove a cluster, if the keywords and key-persons don't show any consistency and are not meaningful to the user, or if documents in the cluster are a hodgepodge from several categories. If the keywords or key-persons make sense to the user, the user gives feedback about document-cluster associations according to these meanings. We don't put constraints on how the user gives the feedback, so the user can make decisions freely based on how she perceives the clustering results, and gives feedback using her own interpretation of the results.

5.3.3. Results

Due to nondeterministic results in any autonomous clustering task, we first obtain 5 different runs of initial clustering results and present each result to the user. Here we assume the cluster number is known to simplify the task. We ask the user to give feedback on these 5 runs individually. Table 1 shows how many entries of different feedback types the user enters for each clustering run. The user spends about 15 mins to finish one run.

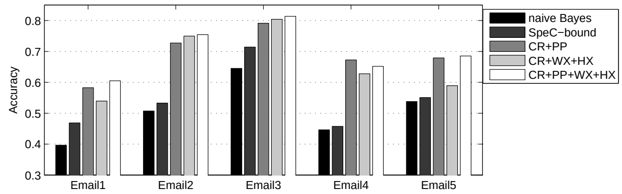


Figure 7. Performance of using three different combinations of feedback types (CR+PP, CR+WX+HX and full feedback CR+PP+WX+HX) on the *EmailYH* dataset. SpeC-bound is a modified SpeClustering model without using any user feedback. User feedback gives huge improvements in all runs.

Figure 7 shows the results using different combinations of feedback types. User's feedback gives huge improvements in all runs (19.55% average accuracy improvements from naive Bayes results to SpeClustering with full feedback). SpeClustering with full feedback performs best in 4 out of 5 runs. In the remaining one run, CR+PP feedback performs best. The number of WX+HX feedback are fewer than PP feedback in these runs. However, CR+WX+HX performs better than CR+PP in 2 runs, which shows that feedback on the hypothesized feature-label properties gives comparable information as feedback on the hypothesized instance-label properties. More compellingly, it is also much easier to get CR+WX+HX feedback than CR+PP in terms of time efficiency. In (Raghavan et al., 2005), they measure the time spend on labeling a document or a feature, and they find a person only need 1/5th of time to label a feature compared to the time to label a document.

In summary, we proposed a method to collect user feedback of a clustering task. We present an interface that enables a user to browse through clustering results and provide several types of feedback. The process requires the user's judgement about which cluster is associated with a real activity they are working on. We also integrate user feedback with the SpeClustering model. The SpeClustering model provides a natural way to adjust parameters according to extended user feedback types. The experimental results show that the SpeClustering model gains significant improvement in a personal email dataset.

6. Conclusion and Future Work

Mixed-initiative clustering combines the advantage of the machine's computational power to analyze huge amount of data, with the advantages of a human's understanding of categories of interest. In this paper, we defined the framework for mixed-initiative clustering, examined the feasibility of modeling user criterion using cluster descriptions, and proposed a new clustering algorithm, SpeClustering, in mixed-initiative clustering formalism and applied the algorithm to the activity extraction task successfully.

We want to further study mixed-initiative clustering in three directions. The first direction is adding new hypothesized property types for mixed-initiative clustering. The

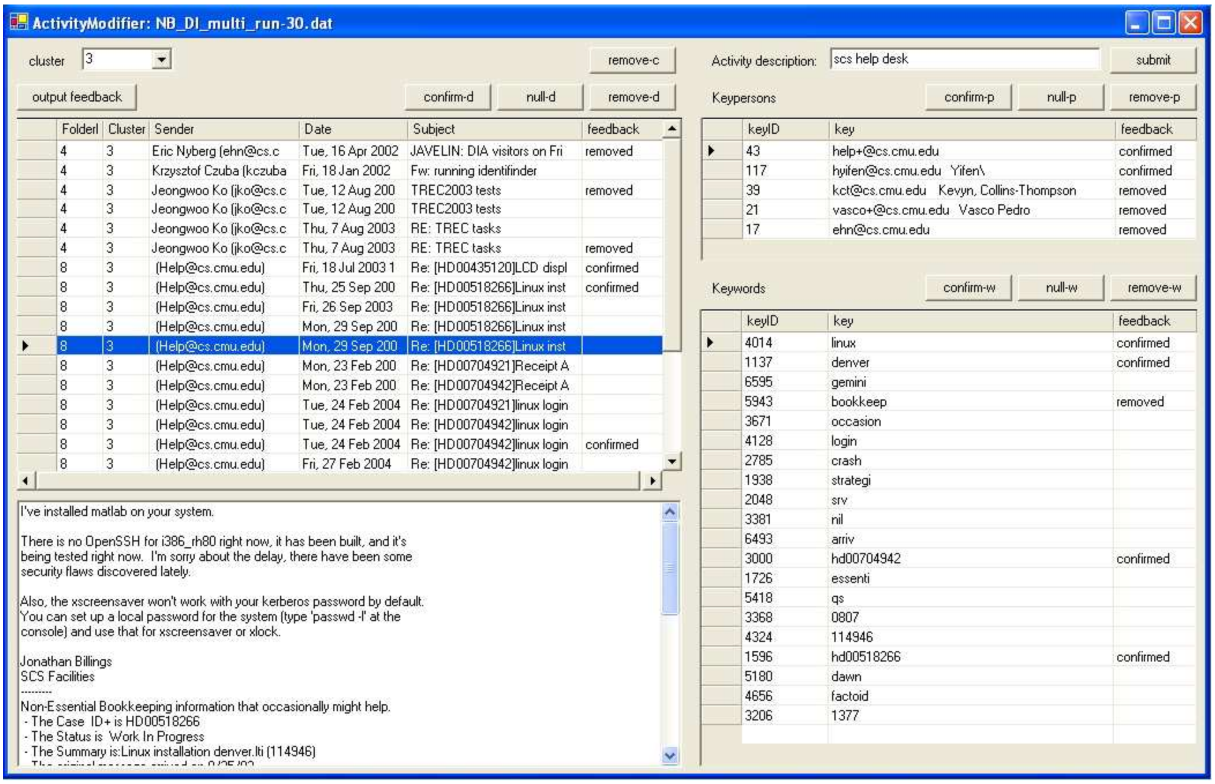


Figure 6. The user interface for feedback gathering. It displays a list of documents, keywords and key-persons for a selected cluster. A reviewer can decide (1) to keep the cluster or not, (2) confirm or remove keywords or key-persons (3) confirm or remove documents, (4) give a short description about the cluster. The reviewer can also go back and forth between clusters to make her feedback consistent.

second direction is combining active learning with mixed-initiative clustering. The third direction is dynamic mixed-initiative clustering that takes into account changes over time in a long-term clustering task. We also want to find more applications for mixed-initiative clustering.

References

Amant, R. S. (1997). Navigation and planning in a mixed-initiative user interface. *AAAI-97*.

Barrett, A., & Starbird, T. (2006). Drudgery relieving commands for mixed initiative planning. *the 5th International Workshop on Planning and Scheduling for Space*.

Blei, D., Ng, A., & Jordan, M. (2003). Latent dirichlet allocation. *Journal of machine Learning Research 3*.

Horvitz, E. (1999). Principles of mixed-initiative user interfaces. *CHI-99*.

Huang, Y., Govindaraju, D., Mitchell, T., Carvalho, V. R., & Cohen, W. (2004). Inferring ongoing activities of workstation users by clustering email. *First Conference on Email and Spam*.

Huang, Y., & Mitchell, T. (2006). Text clustering with extended user feedback. *SIGIR*.

Kerne, A., Koh, E., Sundaram, V., & Mistrot, J. M. (2005). Generative semantic clustering in spatial hypertext. *DocEng '05: Proceedings of the 2005 ACM symposium on Document engineering*.

Mitchell, T. M., Wang, S. H., Huang, Y., & Cheyer, A. (2006). Extracting knowledge about users' activities from raw workstation contents. *AAAI-06*.

Nigam, K., McCallum, A. K., Thrun, S., & Mitchell, T. M. (1998). Learning to classify text from labeled and unlabeled documents. *AAAI-98*.

Raghavan, H., Madani, O., & Jones, R. (2005). Interactive feature selection. *IJCAI-05*.

Thomaz, A. L., & Breazeal, C. (2006). Transparency and socially guided machine learning. *the 5th International Conference on Developmental Learning*.

Thomaz, A. L., Hoffman, G., & Breazeal, C. (2006). Reinforcement learning with human teachers: Understanding how people want to teach robots. *the 15th IEEE International Symposium on Robot and Human Interactive Communication*.

Yang, Y., & Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. *ICML-97*.