# Supervised and Unsupervised Clustering for Instance Response Variable Prediction*

Peter Lubell-Doughtie, 6095445

University of Amsterdam

**Abstract.** We present an incremental approach to cluster assignment which predicts the response variables for a set of instances based upon those assigned to instances that are near it in latent topic space. The method uses a novel combination of supervised clustering and unsupervised clustering to reduce the number of labeled instances needed for accurate classification. In experiments on the political blog corpus we find that predicting response variables based on unsupervised clustering and supervised labeling of a limited number of instances results in performance competitive with that produced by supervised labeling for all instances.

## 1 Introduction

The user's effectiveness in navigating the exceptional amount of available information is largely dependent on how that information is organized. The more intuitive, extensive, and customizable the organization method, the easier it is to find the information of interest within the information set as a whole. The challenge of information navigation and organization is two fold: (i) the desired information is small relative to the total information, a "needle in a haystack" and (ii) the vast majority of information is unlabeled.

In modern information retrieval and knowledge management systems the most common methods used to address these challenges are:

1. keyword search, where the organization is dependent on the similarity between keyword terms and document terms (and meta-terms, especially in the case of non-text)
2. graph based authority measures, normally various forms of spectral ranking [9] algorithms based upon a citation graph measuring centrality and relatedness (e.g. HITS [5] and PageRank [7])
3. categorization or clustering, in which we assign topics to documents or associate latent topics with documents

Hybrid systems are often built which use weighted combinations of these methods, e.g. relational topic models [4]. In this paper we will address clustering

* This report completes the project requirement for *Machine Learning: Principles and Methods* 2010, taught by professor Maarten van Someren.

methods, which fall into the third category. From a machine learning perspective clustering can be viewed on a spectrum from unsupervised through semi-supervised to supervised learning.

When we have no labeled documents the task is fully unsupervised and a form of clustering. As we introduce more labelled documents the task is a hybrid of clustering and classification. With labeled documents we can break our task down into a first step of clustering, and a second step of classification given the clusters in which labeled documents appear. When all documents are labelled we can build a model using supervised methods and use this to predict the class of new documents.

We design a method of cluster based labeling such that as the number of labeled documents increases the accuracy of the labeling of unlabeled documents also increases. This makes it especially applicable to active learning scenarios in which the user can provide additional information to the system or the system can request that the user provide additional information. We compare our cluster labeling method with a classification method that does not take clusters into account. We find that our method has comparable performance and, with a large number of topics, outperforms. Although we have discussed the task as document classification and clustering, the method is readily extendable to arbitrary instances provided we can associate a latent topic model with them.

## 2 Related Work

There has been a significant amount of research in both active learning and clustering. Active learning is the task of determining which instance to request for labeling from a pool of unlabeled instances. Clustering is the task of finding a latent structure which groups instances according to a similarity metric that is useful for the domain at hand — instances within the same cluster should generally be more similar to each other than to instances in other clusters. In the case of text documents, topic models can be inferred from a document's words and clusters defined through a document's most representative topic. We will review these in turn with a focus on research related to our approach.

### 2.1 Active Learning

In their active learning approach, Tong and Koller view the set of labelled instances as defining a version space which is a set of hyperplanes separating the data in some induced feature space [8]. They then use support vector machines (SVM) to find the largest hypersphere in version space whose surface in not touching any of the hyperplanes of the labeled instances. Given this version space, the optimal strategy is to choose new instances for labeling so as to maximally reduce the size of the version space. Tong and Koller prove that choosing query points which half the version space accomplishes this. They present heuristic methods that approximate halving the version space by using the instances'

distances from their separating hyperplanes and the estimated changes to the version space caused by labeling various instances.

Bordes et al. present a similar method in [3]. They note that naively choosing the most misclassified example, called gradient selection, will give poor performance on noisy data sets and suggest to choose the instance closest to the current decision boundary, or the instance from a sample that is closest to the boundary, respectively called active and auto-active sampling. They compare these with a baseline that chooses random instances and show that, on noisy data, gradient selection underperforms random selection while active and auto-active selection outperform all other methods [3].

Often the data we are presented with has multiple views, or feature spaces. For example, web pages and academic papers can be viewed as their text contents or as their citation graphs. We define a contention point as an instance on which each view predicts different labels. Similarly to [3], in [6] Muslea et al. define an "aggressive" method appropriate for little and no noise situations and a "conservative" method for high or unknown noise situations. To choose the contention point for labeling in the aggressive strategy we choose the instance for which the least confident hypothesis makes the most confidant prediction. In the conservative strategy the instance on which the confidence of the hypothesis' predictions disagree the least is chosen, this biases us towards ignoring outliers. If we consider a hypothesis' predictions as defining a decision boundary, the conservative method is analogous to choosing the instance with minimum margin where margin in this case is the sum of the distance from the instance to each decision boundary. (This differs from SVM in that we now have multiple decision boundaries, whereas before we had a single boundary).

All of the above methods rely upon simple decision boundaries — a minimal structuring of the data. The method we develop is based upon generating a stronger structure for the data through topic modeling and then using this structure to apply labeling techniques similar to the above. The methods above can be used to determine how best to label clusters by helping us to decide which cluster instances should be selected for labeling.

### 2.2 Topic Models for Labeling

Supervised latent Dirichlet allocation (sLDA), developed by Blei and McAuliffe, extends latent Dirichlet allocation (LDA) [2] by adding accommodations for a response variable. Figure 1 presents a graphical model of sLDA taken from [1]. For each document we observe the response variable $Y_d$ which is generated by a generalized linear model (GLM) with parameters $\eta$, $\delta$, and the latent topics $Z_{d,n}$ for the $n = 1 \dots N$ words in document $d$. Note that if we exclude the response variable and its parameters we are left with the usual graphical model for LDA.

Using a GLM to model the response variable $y$ allows us to model the variance in $y$. By linking topics to both the response variable and document words ($W_{d,n}$ in the graphical model) topics split their explanatory power over both of these random variables. After training we can use our model to predict the response variable for a new document based on the variationally approximated posterior of

latent variables, $\boldsymbol{\theta}$, and topic to word assignment frequencies, $\mathbf{Z}_n$ [1]. Combining sLDA and LDA provides a powerful method for semi-supervised learning and label prediction; this is the approach we take.

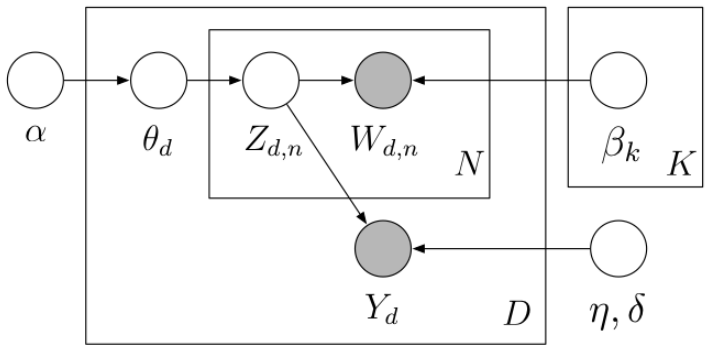


Fig. 1: Graphical model of sLDA. $\alpha$ is a Dirichlet parameter for the topic mixing proportion $\theta_d$ which generates the topics $Z_{d,n}$ through a multinomial distribution. $Z_{d,n}$, $\eta$, and $\delta$ generate the response variable $Y_d$ through a GLM. Topics $Z_{d,n}$ and parameter $\beta_k$ generate each of the $N$ words $W_{d,n}$ for document $d$. Shaded nodes represent observed variables. Taken from Blei and McAuliffe [1].

## 3 Method

We are given a corpus $\mathcal{X} = \{d_1, \ldots, d_n\}$ formed by a set of documents $d_i$ where for each $d_i$ we have:

$$d_i = \big\{\langle w_{i,1}, \ldots, w_{i,m}\rangle, \{c_{i,1}, \ldots, c_{i,l}\}\big\} \tag{1}$$

which is a tuple containing an ordered multi-set of words (or terms) and a set of categories where $l \ll m$. We assume all $d_i \in \mathcal{X}$ have at least one category and we say $d_{i,c} = \emptyset$ when a document has no categories, i.e. it is unlabeled. We further assume that our corpus is a subset of a larger corpus, $\mathcal{X}^*$, which contains documents that we do not have the category listings for. Formally, we assume $\mathcal{X} \subset \mathcal{X}^*$, and let $\overline{\mathcal{X}} = \mathcal{X}^* \setminus \mathcal{X}$, where $d_i \in \overline{\mathcal{X}} \rightarrow d_{i,c} = \emptyset$. Our goal is to determine the categories of the documents in $\overline{\mathcal{X}}$ given the documents in $\mathcal{X}$. Ideally we desire a method that produces better than random performance in the case $\mathcal{X} = \emptyset$ and better performance as $n$, the number of documents in $\mathcal{X}$ — i.e. cardinality of $\mathcal{X}$ — increases.

We can view this as a semi-supervised learning problem[1] where we have a set of labeled data $\mathcal{X}$ and desire labels for the unlabeled set $\overline{\mathcal{X}}$. We assume there

[1] Excepting the case $\mathcal{X} = \emptyset$, where it is an unsupervised learning problem.

are a set of latent topics such that the probability documents share categories is inversly proportional to their distance in topic space.

### 3.1 Placing Documents in Topic Space

To create a latent topic space for documents we use a latent clustering algorithm. In this implementation we will use latent Dirichlet Allocation. We define a clustering operator, $\mathfrak{C}$, such that:

$$\mathfrak{C}(d_i) = [t_{i,1}, \ldots, t_{i,s}] = \mathbf{t}_i \tag{2}$$

where $s$ is the number of latent topics and $\mathbf{t}_i$ is a vector of topic probabilities for document $d_i$ which form a probability distribution such that:

$$\forall t_j \in \mathbf{t}_i [0 \leq t_j \leq 1]$$
$$\sum_{t_j \in \mathbf{t}_i} t_j = 1$$

We can apply $\mathfrak{C}$ to a corpus to obtain a matrix of topic probabilities $\mathbf{T}$:

$$\mathfrak{C}(\mathcal{X}) = \mathbf{T} = \begin{bmatrix} \mathbf{t}_1 \\ \vdots \\ \mathbf{t}_n \end{bmatrix} \tag{3}$$

where $n$ is again the number of documents in $\mathcal{X}$.

### 3.2 Clustering Topic Space

From the topic probabilities we can further derive specific categories for a document by naively assigning each document to membership in the cluster denoted by its topic of highest probability. We make a distinction between *topics*, which are latent probability assignments, and *clusters*, which are binary membership functions, i.e. they form a partition of the corpus into sets of documents. Using clusters we partition our corpus $\mathcal{X}$ into a set of clusters, such that for each cluster, $\mathcal{X}_j$, $d_i \in \mathcal{X}_j$ iff $j = \text{argmax}_k\, t_{i,k}$. Although we lose a significant amount of information through this approach it greatly reduces the complexity of labeling a new document. We now need to only consider what the label of the cluster the document would be in is, and not the labels of all the clusters and their varying likelihoods.

### 3.3 Labeling Clusters

The range of methods for labeling clusters can be divided into (i) global methods that rely on all the documents in the cluster, (ii) global methods that weight documents based on cluster centrality, and (iii) local methods based upon only the central document, or documents above some centrality threshold. Consider

a corpus $\mathcal{X}$ that we have partitioned into a set of clusters $\{\mathcal{X}_i\}$. Now, consider a cluster $\mathcal{X}_j \in \{\mathcal{X}_i\}$, which consists of documents for which we have topic vectors and term frequency vectors. The topic vectors are based on all documents in the corpus and to retrieve more discriminative topics we can execute $\mathfrak{C}(\mathcal{X}_j)$ which returns a set of topics relative to *only* the documents in the cluster.

We then consider corpus topic space, cluster topic space, or term space as metric spaces based upon the document vectors in these spaces. We calculate the center of the subspace formed by the documents in $\mathcal{X}_j$ and we label the center of this space $\bar{\mathbf{t}}_j$. We then calculate the centrality of a document $i$ as a scalar using the inner product:

$$\texttt{centrality} = \bar{\mathbf{t}}_j \cdot \mathbf{t}_i. \tag{4}$$

We can express all the above labeling methods by weighting the contributions of documents given their inner product with the cluster subspace's center.

### 3.4 Clustered LDA Algorithm

The algorithm we use combines LDA and sLDA to predict document labels from supervised labels generated from only a subset of the unlabeled documents. This algorithm has two primary use cases:

1. reducing computational cost
2. improving accuracy by pooling instances with related features

The first case holds when the cost of running sLDA prediction on the whole test set is greater than that of running LDA, cluster assignment, and sLDA only once for each cluster center. The second case is domain and optimization parameter dependent; it is more likely to be applicable with noisy data sets. In this case the method will improve the accuracy if the predicted response variable of the cluster's most central document is accurate and the correct response variables of intra-cluster documents are correlated.

We present the general cluster based classification algorithm below as Algorithm 1. We begin by creating a supervised and unsupervised model from our clustering algorithms given disjoint labeled data $\mathcal{X}$ and unlabeled data $\overline{\mathcal{X}}$, i.e. $\mathcal{X} \cap \overline{\mathcal{X}} = \emptyset$. Then, from lines 6 to 11, we build a mapping, `docForTopic`, from topics to representative documents.

On line 7 we use our unsupervised model to assign to $\mathbf{t}_d$ the topic that is most representative of document $d$. Then, for each topic $t$, we find the set of documents where this topic has the highest probability and assign the document with the highest score, as defined by the function `topicDocScore`, to be the representative document for this topic. As in Eq. 4, we define `topicDocScore` as the inner product were $\bar{\mathbf{t}}_j$ holds the normalized word counts within cluster $j$ and $\mathbf{t}$ holds the normalized document word counts. These documents will be responsible for the response variables assigned to all documents in their clusters.

Note that problems will arise if there is a topic that is not the most probable amongst any of the documents. We handle this case on lines 13 through 17, where any topics in our unsupervised model that don't have a document assigned to

**Algorithm 1** General form for method using unsupervised clustering combined with supervised clustering to choose response based on cluster center labels.

**Require:** clustering algorithm $\mathfrak{C}$, supervised clustering algorithm $\mathfrak{S}$
**Require:** labeled data $X$, unlabeled data $\overline{X}$
1: // Create our supervised model $\mathbf{\Psi}$
2: $\mathbf{\Psi} \leftarrow \mathfrak{S}(\mathcal{X})$
3: // Create our unsupervised model $\mathbf{\Pi}$
4: $\mathbf{\Pi} \leftarrow \mathfrak{C}(\overline{\mathcal{X}})$
5: // From $\mathbf{\Pi}$ build most likely documents for topics
6: $\texttt{docForTopic} \leftarrow \emptyset$
7: $\mathbf{t} \leftarrow \texttt{maxTopics}(\mathbf{\Pi}, \overline{\mathcal{X}})$
8: **for** $t \in \mathbf{t}$ **do**
9: $\quad \overline{\mathcal{X}}_t \leftarrow$ documents where $t$ has highest probability
10: $\quad \texttt{docForTopic}(t) \leftarrow \text{argmax}_{d \in \overline{\mathcal{X}}_t} \texttt{topicDocScore}(\mathbf{t}_d, d, \mathbf{\Pi})$
11: **end for**
12: // back-off in case we have any unrepresented topics
13: **for** $t \in \mathbf{\Pi_t}$ **do**
14: $\quad$ **if** $\texttt{docForTopic}(t)$ is null **then**
15: $\quad\quad \texttt{docForTopic}(t) \leftarrow \text{argmax}_{d \in \overline{\mathcal{X}}} \texttt{topicDocScore}(\mathbf{t}_d, d, \mathbf{\Pi})$
16: $\quad$ **end if**
17: **end for**
18: $\mathbf{y} \leftarrow \texttt{predict}(\mathbf{\Psi}, \texttt{docForTopic})$
19: **for** $d \in \overline{\mathcal{X}}$ **do**
20: $\quad d^* \leftarrow \texttt{docForTopic}(t_d)$
21: $\quad y_d \leftarrow y_{d^*}$
22: **end for**
23: **return** $\mathbf{y}$

them are assigned the document with the highest score from all documents in the corpus. We can view this as a sort of "back-off" model which uses a larger pool of data when the restricted pool is not informative enough.

On line 18 we use the sLDA `predict` function to assign each representative document in `docForTopic` a response variable $y$ based upon our supervised model. Then on lines 19 through 22 we use these response variables to assign the response variables for all other documents in the cluster. We stress that this algorithm can be used with any specific clustering algorithms given one is supervised and another is unsupervised.

## 4 Experiments

We apply our algorithm to the `poliblog` corpus under various settings. We divide the corpus into training and test sets and generate an sLDA model from the training set. As a baseline we use the sLDA predictor on the test set as a whole, as opposed to our experiments, which use clustered prediction, as defined by Algorithm 1, on the test set.

### 4.1 Data

We use the `poliblog` corpus of 773 political blogs where each blog has been given an accompanying rating of either $-100$, liberal, or 100, conservative [1]. We use the ratings as response variables and take a portion of the blogs with their ratings, denoting the labeled set $\mathcal{X}$, and then take the remainder without their ratings denoting the unlabeled set, $\overline{\mathcal{X}}$. The goal is to predict the rating of the unlabeled blogs given the labeled blogs.

### 4.2 Evaluation

We use the mean squared error (MSE) to evaluate the results. Although this can be seen as a binary assignment problem, the changing size of the test sets makes the MSE more amenable to comparison than other loss functions, e.g. 0-1 loss. We note that the clustering prediction algorithm is a fundamentally weaker algorithm than fully supervised prediction in that it makes the assumption that ratings are strongly dependent on a document's location in feature space. Given this, we might not expect the clustering method to perform well.

### 4.3 Parameter Settings

We use sLDA and LDA with parameters $\alpha = \eta = 0.1$ for the Dirichlet and GLM distributions. We run LDA for 50 iterations and sLDA for 10 iterations of expectation and 4 iterations of maximization. The number of topics is set to 10, 15, and 25 in different runs.

## 5 Results and Discussion

Varying the number of topics used in prediction results in different results as more topics allow finer grained clusters and a greater number of documents to be labeled by the sLDA algorithm. Below we refer to the ratings as predicted by running sLDA prediction on all unlabeled documents as "predicted" and we refer to the ratings as predicted using clustering with LDA and sLDA, as described in Algorithm 1, as "clustered". In Table 1 we present the MSE for the predicted and clustered methods using 10, 15, and 25 topics. The number of training instances is varied from 1 to 751 in increments of 25. The results are the same until greater than 301 training instances are used, therefore results for less than 301 training instances are excluded from the calculations.

Table 1 presents the mean and minimum MSE and the variance in MSE taken over all sizes of training instances tested greater than or equal to 301. The MSE has been scaled to the range [0,1], with 0 indicating no error. With 25 topics using the clustered method we have a mean MSE of 0.5717, which is below the baseline predicted MSE of 0.5951. The MSE of the clustered method has a variance of 0.4592, significantly higher than the 0.2717 variance of the baseline. This is somewhat expected as the performance of the clustering is likely unstable with respect to the number of training examples used.

Table 1: Mean and minimum mean squared error (MSE), and the variance in the MSE, for the baseline predicted method and the clustered method. MSE has been scaled to $[0, 1]$, lower MSE is better, no differences are statistically significant.

| Number Topics | Experiment | Mean MSE | MSE Variance | Min MSE |
|---|---|---|---|---|
| 10 | Predicted | **0.6509** | 0.3289 | 0.09278 |
| | Clustered | 0.6765 | 0.4283 | 0.1277 |
| 15 | Predicted | **0.6071** | 0.3452 | 0.2128 |
| | Clustered | 0.6291 | 0.4118 | 0.4118 |
| 25 | Predicted | 0.5951 | 0.2717 | 0.1702 |
| | Clustered | **0.5717** | 0.4592 | 0 |

Figure 2 presents a plot of the MSE for both the predicted and clustered method given 25 topics (additional plots for the other topic sizes are presented in the Appendix). We see that, generally, the clustered method is only able to outperform the supervised baseline when there are a larger number of training instances. We see a transition at around 550 training instances, after which clustered outperforms. Interestingly, when the number of training instances is very large, above 601, which is greater than 77 % of the total data size, the clustering method significantly outperforms.

It is not clear exactly what causes this but it is likely a combination of the advantages clustering has given a small amount of data to cluster and the disadvantages of sLDA given a small amount of data to classify. In a sense, with a large training and small test set, sLDA has less room for error because the effect of a mistake in classification holds greater weight on the total error. Further, as the test set gets smaller the clusters will become more representative in virtue of their need to cover a smaller number of data points.

### 5.1 Varying the Number of Topics

In Table 1 we additionally present the results for 10 and 15 topics. With smaller numbers of topics the predicted method outperforms the clustered method but not significantly. We see that as the number of topics increases the performance of both methods increases and the performance of the clustered method versus the predicted method improves. This is expected as more clusters imply greater discriminating power, for both the clustered and predicted methods, which translates into better use of labels and lower error. The variance is unstable but generally decreases for predicted and increases for clustered. The increase in variance for a greater number of topics could be because more clusters means a higher chance of being assigned to a different cluster as the size of the data set changes.

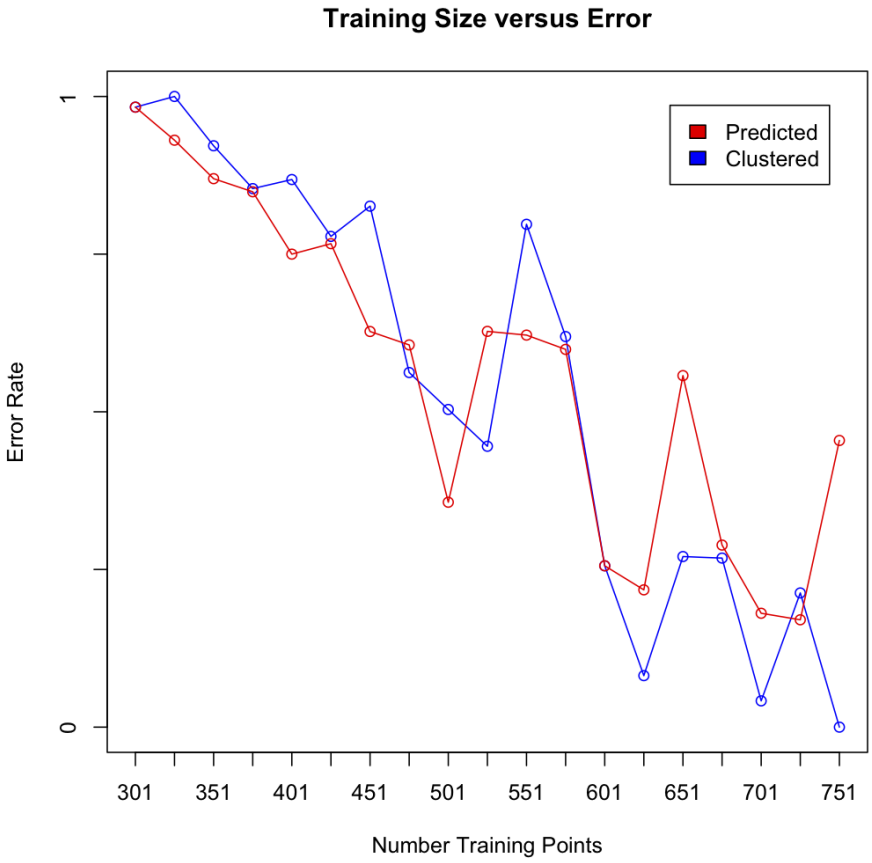


Fig. 2: Comparison of error for generated ratings for predicted and clustered methods. 25 Topics are used in sLDA and LDA.

## 6 Conclusion

We have presented a "semi-supervised" algorithm for labeling instances which leverages preexisting labels and clusters in the instance feature space. In applying this algorithm to a corpus of political blogs that are judged as either conservative or liberal our algorithm is able to outperform the judging prediction task when compared against a baseline predictor using pure sLDA. Although these results are not statistically significant they suggest our method may be a fruitful avenue for additional research given extensions and improvements are developed.

A simple extension involves mixing and matching various supervised and unsupervised clustering algorithms. A particularly nice feature of the method we have presented is that it is independent of the clustering algorithms and

can be combined with arbitrary or multiple clustering algorithms depending on the proclivities of the data set. A principled method of correctly choosing the appropriate clustering algorithms would test a set of algorithms on a held out data set and choose the combination with the lowest error on the held out set.

Further interesting research involves testing the various alternatives to relying upon the label of only the center document. When we use more documents as representative of a cluster we will incur larger computational costs, it remains to be seen whether this will be accompanied by worthwhile reductions in testing error. Much work is still to be done combining multiple features of the data sets. Specifically, in the case of documents with an concomitant citation graph, relational topic modeling would be a more appropriate model for the data.

## References

1. David M. Blei and Jon D. McAuliffe, *Supervised topic models*, NIPS, 2007.
2. David M. Blei, Andrew Y. Ng, and Michael I. Jordan, *Latent dirichlet allocation*, J. Mach. Learn. Res. **3** (2003), 993–1022.
3. Antoine Bordes, Seyda Ertekin, Jason Weston, and Lon Bottou, *Fast kernel classifiers with online and active learning*, Journal of Machine Learning Research **6** (2005), 1579–1619.
4. Jonathan Chang and David M. Blei, *Hierarchical relational models for document networks*, Annals of Applied Statistics (2010).
5. Jon M. Kleinberg, *Authoritative sources in a hyperlinked environment*, J. ACM **46** (1999), 604–632.
6. Ion Muslea, Steven Minton, and Craig A. Knoblock, *Active learning with multiple views.*, J. Artif. Intell. Res. (JAIR) **27** (2006), 203–233.
7. L. Page, S. Brin, R. Motwani, and T. Winograd, *The pagerank citation ranking: Bringing order to the web*, Proceedings of the 7th International World Wide Web Conference (Brisbane, Australia), 1998, pp. 161–172.
8. Simon Tong and Daphne Koller, *Support vector machine active learning with applications to text classification*, J. Mach. Learn. Res. **2** (2002), 45–66.
9. Sebastiano Vigna, *Spectral ranking*, CoRR **abs/0912.0238** (2009).

## Appendix


Fig. 3: Comparison of error for generated ratings for predicted and clustered methods. 10 Topics are used in sLDA and LDA.

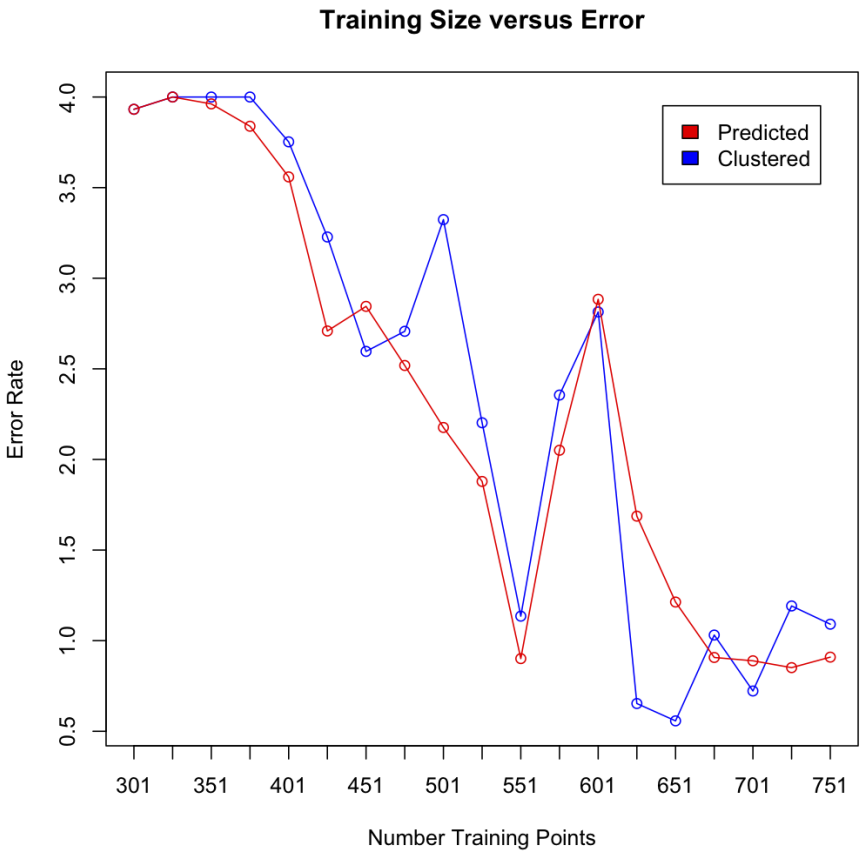


Fig. 4: Comparison of error for generated ratings for predicted and clustered methods. 15 Topics are used in sLDA and LDA.