

Clustering with Model-Level Constraints

David Gondek ^{*} Shivakumar Vaithyanathan [†] Ashutosh Garg [‡]

Abstract

In this paper we describe a systematic approach to uncovering multiple clusterings underlying a dataset. In contrast to previous approaches, the proposed method uses information about structures that are *not desired* and consequently is very useful in an exploratory datamining setting. Specifically, the problem is formulated as constrained model-based clustering where the constraints are placed at a model-level. Two variants of an EM algorithm, for this constrained model, are derived. The performance of both variants is compared against a state-of-the-art information bottleneck algorithm on both synthetic and real datasets.

1 Introduction

Clustering is a form of unsupervised learning which groups instances based on their similarity. Most clustering algorithms either optimize some objective [15] or make use of a similarity/distance function between instances [13]. In real applications there may be multiple clusterings underlying the data and the user may typically not know a priori which clustering is of interest. A commonly identified liability of clustering is the large degree to which the solution returned is dictated by assumptions inherent in the choice of similarity of objective function [12]. To address this, a number of *semi-supervised clustering* approaches have been proposed. They all share a dependency on user-supplied knowledge about the structure of the desired solution. They differ, however, on how this knowledge is employed. Approaches vary from enforcing constraints directly within the algorithm [22], seeding the initialization of clusters [1], learning distance functions based on user input as in [23] and [12], or even a combination of these approaches such as in [3, 2] which learns a distance function *and* enforces constraints within the algorithm.

The problem with these approaches is that the user may be unable to define a useful similarity function or

specify prior knowledge of a desired solution. For example, consider the “Yahoo! problem” of [6]: you are given 100,000 text documents and asked to group them into categories. You are not given any information in advance about what categories to use or which documents are related. The goal is then to create a categorization which can be browsed and accessed efficiently. In [6], they propose an iterative solution which supplies clusterings to the user and receives feedback according to three forms. The user may specify “This document doesn’t belong in here,” “Move this document to that cluster,” and “These two documents shouldn’t (or should) be in the same cluster.” We note that all three forms of feedback require the user to have information about a desired clustering. Furthermore, with 100,000 documents, a user may be able to inspect only a small percentage of the documents. Feedback obtained may be appropriate for the sample but misleading for the overall dataset. This will have the effect of biasing the search and suppressing other interesting clusterings of which the user is not aware.

We propose as an alternate approach to provide the user with a series of high-quality non-redundant clusterings. In our case, feedback would not require positive knowledge about which instances should be assigned to which cluster. Instead, the user’s interaction would be to specify, “Find another clustering.” With this in mind, we develop a mechanism to systematically search through the clusterings underlying the data. A formal framework to incorporate this non-redundancy as constraints in model-based clustering, as well as associated algorithms for obtaining high-quality solutions, is the focus of this paper.

2 Problem Definition

Unsupervised clustering algorithms obtain the *natural* grouping underlying the data¹. In some cases, prior knowledge about a desired clustering can be incorporated using a variety of constrained clustering techniques [22] [14] [23]. Briefly, these techniques require the analyst to provide explicit knowledge of the target clustering. E.g. in [22]: *must-link* constraints

^{*}Department of Computer Science, Brown University, Providence, RI 02912, dcg@cs.brown.edu

[†]IBM Almaden Research Center, 650 Harry Rd., San Jose, CA 95120, shiv@almaden.ibm.com

[‡]Google, Inc., 1600 Amphitheater Parkway, Mountain View, CA 94043, ashutosh@google.com

¹The natural clustering is dependent upon the choice of objective function.

require two instances to be assigned the same cluster and *cannot-link* constraints require two instances to be assigned to different clusters.

In contrast, we consider the less-studied setting in which the knowledge available pertains to particular solutions which are *not desired*. Our goal is to use this knowledge to discover new clusterings in a systematic fashion. In this paper this knowledge can be expressed in the following two ways:

- [a.] **Known clusterings** One or more clusterings are given.
- [b.] **Negative features** A set of “negative” features is specified and clusterings associated with these features are undesired.

In case [a.] above, the goal is to find newer clusterings which do not resemble the known clusterings. In case [b.], the goal is to find clusterings not associated with the negative features.

Table 1: Example: 5-dimensional Dataset

i	remaining features			negative features	
	$y_{.1}^+$	$y_{.2}^+$	$y_{.3}^+$	$y_{.1}^-$	$y_{.2}^-$
1	1	1	1	1	0
2	1	1	1	1	0
3	1	0	0	0	1
4	1	0	0	0	1
5	0	1	1	1	0
6	0	1	1	1	0
7	0	1	0	0	1
8	0	0	0	0	1

To understand the implications of negative features, consider the example in Table 1. Each instance has been separated into “negative” and “remaining” features. Clustering only on the negative features produces clusters $\{1, 2, 5, 6\}$ and $\{3, 4, 7, 8\}$. We desire clusterings that are different from this grouping of the data. The naive solution of simply ignoring $y_{.1}^-$ and $y_{.2}^-$ is not sufficient since $y_{.1}^-$ and $y_{.2}^-$ are correlated with $y_{.2}^+$ and $y_{.3}^+$. Consequently the naive solution will still result in finding $\{1, 2, 5, 6\}$ and $\{3, 4, 7, 8\}$ ². However, a more intelligent search may find $\{1, 2, 3, 4\}$ and $\{5, 6, 7, 8\}$ which is a different and meaningful grouping of the data. What complicates matters is that in general there may be many natural clusterings and negative features may have more than one associated clustering. The task at hand is then to find a meaningful grouping of the data

²Note that even repeated runs of clustering techniques which depend on random initializations, such as k-means [15], will typically return this dominant clustering.

which is different from all clusterings associated with the negative features.

Techniques have been derived from the Information Bottleneck framework[20] for related problems. The Information Bottleneck with Side Information[5] may be applied to clustering and we will use it as a baseline in our experiments. In [9], the Conditional Information Bottleneck was proposed as a technique to find a non-redundant clustering in a dataset given one known clustering. It is not, however, designed for the setting in which negative features may be associated with several known clusterings.

3 Constrained Model-Based Clustering

Mathematically stated, we have a n -instance dataset $\mathcal{Y} = \{\mathbf{y}_i : i = 1 \dots n\}$. Each instance, \mathbf{y}_i is assumed to have negative \mathbf{y}_i^- and remaining \mathbf{y}_i^+ features where m^- is the number of negative features and m^+ is the number of remaining features. y_{ij}^- and y_{ij}^+ represent the j th negative/remaining feature of the i th instance. We use $y_{.j}^-$ and $y_{.j}^+$ to represent the j th negative/remaining feature for a generic instance. All features are assumed to be *binary*, i.e., $\mathbf{y}_i^- \in Y^-$ where Y^- is the set of all such vectors $\{0, 1\}^{m^-}$ and $\mathbf{y}_i^+ \in Y^+$ where Y^+ is the set of all such vectors $\{0, 1\}^{m^+}$. The negative features \mathbf{y}_i^- are assumed to take one of the following forms:

Known clusterings One or more clusterings are given. In this case \mathbf{y}_i^- is represented by a vector where the entries for the clusters to which i is assigned are set to 1 and all other entries are 0.

Negative features A set of m^- negative features, $\{\nu(1) \dots, \nu(m^-)\}$, are specified. In this case \mathbf{y}_i^- is represented as the vector, $\mathbf{y}_i^- = [y_{i\nu(1)}, \dots, y_{i\nu(m^-)}]$.

3.1 Model Recall that our interest is in grouping the instances to maximize their similarity in \mathbf{y}_i^+ while simultaneously searching for structure that is not already present in \mathbf{y}_i^- . Intuitively, this suggests a model where \mathbf{y}_i^- and the cluster, c_k , are independent given \mathbf{y}_i^+ . This captures that the \mathbf{y}_i^- for an instance should be a consequence of its \mathbf{y}_i^+ features. Thus, \mathbf{y}_i^+ are assumed to be drawn from K clusters, c_1, \dots, c_K , while the \mathbf{y}_i^- are drawn conditioned on \mathbf{y}_i^+ . Now, assuming \mathcal{Y} is independently drawn gives the following log-likelihood, $l(\mathcal{Y})$, for the dataset:

$$(3.1) \quad l(\mathcal{Y}) = \sum_{i=1}^n \log \sum_{k=1}^K p(\mathbf{y}_i^- | \mathbf{y}_i^+) p(\mathbf{y}_i^+ | c_k) p(c_k).$$

3.2 Objective Function for CMBC The unconstrained task would be to find a clustering which max-

imizes (3.1). Here we extend (3.1) to incorporate the prior knowledge as model-level constraints. The term *model-level constraints* refers to restrictions on the space of clustering. We begin by representing the prior knowledge as constraints.

3.2.1 Representing Knowledge as Constraints

Recall from above that both forms of knowledge may be expressed via \mathbf{y}_i^- . A natural way to express the constrained problem is [note an implicit assumption that cluster probabilities are equal]:

$$(3.2) \quad \begin{aligned} \max \quad & l(\mathcal{Y}) \\ \text{s.t.} \quad & p(c_j|\mathbf{y}_i^-) = p(c_k|\mathbf{y}_i^-) \quad \forall i, j, k \\ & \text{where } \mathbf{y}_i^- \text{ is negative information.} \end{aligned}$$

The intuition behind the constraints is to enforce the requirement that the value taken by \mathbf{y}_i^- should not influence the cluster assignments.

The constraints in (3.2) may be too restrictive and allow only degenerate solutions. To soften the constraints, we replace them with the requirement that the conditional entropy, $H(C|Y^-)$, be high, where $H(C|Y^-) = -\sum_{k,i} p(c_k, \mathbf{y}_i^-) \log p(c_k|\mathbf{y}_i^-)$. Note that $H(C|Y^-)$ is maximized when the constraints in (3.2) are met. This results in the following objective function:

$$(3.3) \quad \mathcal{L} = (1 - \gamma)l(\mathcal{Y}) + \gamma H(C|Y^-).$$

where γ acts as a tradeoff between the loglikelihood and the penalty terms. Intuitively, as γ is increased the resulting solution is farther away from the clusterings associated with Y^- .

3.2.2 Approximating the Objective Function

(3.3) Ultimately, we wish to derive methods to maximize the objective function (3.3) and in the next section, we will derive an EM algorithm. Before that, however, it is necessary to approximate (3.3) to facilitate easier computation in the EM algorithm. We perform two approximations: a. bounding the $H(C|Y^-)$ term and b. empirical approximation.

Bounding $H(C|Y^-)$ The objective in (3.3) is problematic in its current form because it contains $H(C|Y^-)$, which is unwieldy to compute from the parameters $p(\mathbf{y}^-|c)$ and results in an expression which is difficult to optimize. We can, however, exploit the fact that $H(C|Y^-) = H(C, Y^-) - H(Y^-)$ to rewrite (3.3) as:

$$(3.4) \quad \mathcal{L} = (1 - \gamma)l(\mathcal{Y}) + \gamma (H(C, Y^-) - H(Y^-)).$$

Then, both terms $H(C, Y^-)$ and $H(Y^-)$ may be

computed easily from $p(\mathbf{y}_i^-|c_k)$ according to: (3.5)

$$H(C, Y^-) = -\sum_{\mathbf{y}_i^- \in Y^-} \sum_{k=1}^K p(\mathbf{y}_i^-|c_k)p(c_k) \log(p(\mathbf{y}_i^-|c_k)p(c_k)), \quad (3.6)$$

$$H(Y^-) = -\sum_{\mathbf{y}_i^- \in Y^-} \sum_{k=1}^K p(\mathbf{y}_i^-|c_k)p(c_k) \log\left(\sum_{k=1}^K p(\mathbf{y}_i^-|c_k)p(c_k)\right).$$

The remaining hurdle is that each term contains a log of sums, which prevents us from deriving closed-form update equations in an EM algorithm: the $H(Y^-)$ sums over k within the log term. Further, as will be seen in **Appendix A**, the $p(\mathbf{y}_i^-|c_k)$, which occurs in the log terms of both $H(Y^-)$ and $H(C, Y^-)$, also requires performing a sum. We address this issue by replacing these entropy terms with quadratic bounds.

The Shannon entropies in (3.5) and (3.6) can be approximated by the commonly-used Havrda-Charvát structural δ -entropy[11] to obtain quadratic expressions, however this approximation can be quite loose³. Instead we make use of quadratic bounds recently presented in [10] and [21]. These lower and upper bounds are built around the *index of coincidence (IC)*, the probability of drawing the same element in two independent trials. From [21], we obtain the lower bound which we denote by $H^l(X)$:

$$(3.7) \quad H(X) \geq H^l(X) \doteq \delta_d - \beta_d \sum_x p(x)^2,$$

where d is the number of elements x , used to define:

$$(3.8) \quad \delta_d = \ln(d+1) + d \ln\left(1 + \frac{1}{d}\right),$$

$$(3.9) \quad \beta_d = (d+1)d \ln\left(1 + \frac{1}{d}\right).$$

From [10], we use the upper bound on $H(X)$, which we denote as $H^u(X)$:

$$(3.10) \quad H(X) \leq H^u(X) \doteq (\ln d) \cdot \left(1 - \frac{1}{1 - \frac{1}{d}} \left(\sum_x p(x)^2 - \frac{1}{d}\right)\right).$$

Applying (3.7) to $H(Y^-, C)$ and (3.10) to $H(Y^-)$ in our objective function (3.4) we obtain:

$$(3.11) \quad \mathcal{L} \geq (1 - \gamma)l(\mathcal{Y}) + \gamma (H^l(C, Y^-) - H^u(Y^-))$$

Empirical Approximation We describe how to compute $p(\mathbf{y}_i^-|c_k)$ and terms in $l(\mathcal{Y})$ in **Appendix A**.

³We attempted to use this approximation in our experiments but found the approximations to differ substantially from the Shannon entropies and result in very poor performance.

For this paper we assume the $p(c_k)$ are constant. From the computation in (A.5), it is clear that $p(\mathbf{y}_i^-|\mathbf{y}_i^+)$ is constant for a given dataset and so can be ignored in $l(\mathcal{Y})$. Also, from (A.5), we see the $p(\mathbf{y}_i^-|\mathbf{y}_h^+)$ can be estimated only for those \mathbf{y}_i^- in the dataset \mathcal{Y}^- , which requires that the summation in the entropy terms $\tilde{H}^l(C, Y^-)$ and $\tilde{H}^u(Y^-)$ be restricted to those $\mathbf{y}_i^- \in \mathcal{Y}^-$. We denote these empirical approximations as $\tilde{H}^l(C, Y^-)$ and $\tilde{H}^u(Y^-)$ which are defined as:

$$(3.12) \quad \tilde{H}^l(C, Y^-) = \left(\delta_{|Y^-|} - \beta_{|Y^-|} \sum_{\mathbf{y}_i^- \in \mathcal{Y}^-} \sum_{k=1}^K (p(\mathbf{y}_i^-|c_k)p(c_k))^2 \right),$$

$$(3.13) \quad \tilde{H}^u(Y^-) = (\log |Y^-|) \cdot \left(1 - Q \sum_{\mathbf{y}_i^- \in \mathcal{Y}^-} \left(\sum_{k=1}^K p(\mathbf{y}_i^-|c_k)p(c_k) \right)^2 \right),$$

where $Q = \frac{1}{1-1/|Y^-|}$ and where $|Y^-|$ is the number of unique \mathbf{y}^- which occur in the dataset. The $\tilde{H}^l(C, Y^-)$ and $\tilde{H}^u(Y^-)$ terms of (3.11) are replaced with $\tilde{H}^l(C, Y^-)$ and $\tilde{H}^u(Y^-)$ to produced the final approximation of the objective function, $\tilde{\mathcal{L}}$:

$$(3.14) \quad \tilde{\mathcal{L}} = l(\mathcal{Y}^+) + \gamma(\tilde{H}^l(C, Y^-) - \tilde{H}^u(Y^-)).$$

4 EM Algorithm

Select annealing rate $\alpha < 1$. Initialize T to be high. Randomly initialize $p(y_{.j}^+|c_k)$. Loop until hard-assignment obtained:

1. Loop until converged:
 - For $j = \{1 \dots m^+\}$
 - For $k = \{1 \dots K\}$:
 - E-Step Compute assignment expectations:
 - for all instances $i = 1 \dots n$:
$$(4.15) \quad q(c_k|\mathbf{y}_i) \propto p(c_k)p(\mathbf{y}_i^+|c_k)^{1/T}$$
 - M-step Maximize for $p(y_{.j}^+|c_k)$ by solving:

$$(4.16) \quad F_3 p(y_{.j}^+|c_k)^3 + F_2 p(y_{.j}^+|c_k)^2 + F_1 p(y_{.j}^+|c_k) + F_0 = 0$$
2. Decrease temperature $T \leftarrow \alpha T$

Figure 1: CMBC Algorithm: Partial Maximization (CMBCpm)

The E-step [shown in (4.15)] is unaffected by the

Select annealing rate $\alpha < 1$. Initialize T to be high. Randomly initialize $p(y_{.j}^+|c_k)$. Loop until hard-assignment obtained:

1. Loop until converged:
 - E-Step Compute assignment expectations:
 - for all instances $i = 1 \dots n$:
$$(4.17) \quad q(c_k|\mathbf{y}_i) \propto p(c_k)p(\mathbf{y}_i^+|c_k)^{1/T}$$
 - M-step For $k = \{1 \dots K\}, j = \{1 \dots m^+\}$:
 - Maximize for $p(y_{.j}^+|c_k)$ by solving:

$$(4.18) \quad F_3 p(y_{.j}^+|c_k)^3 + F_2 p(y_{.j}^+|c_k)^2 + F_1 p(y_{.j}^+|c_k) + F_0 = 0$$
2. Decrease temperature $T \leftarrow \alpha T$

Figure 2: CMBC Algorithm: Batch Update (CMBCbu)

second and third terms from (3.14)⁴. The M-step finds the maximum likelihood estimates for the model parameters, $p(y_{.j}^+|c_k)$, given the $q(c_k|\mathbf{y}_i)$ from (4.15). The derivation, as described in **Appendix B**, produces the cubic equations in (4.16) and (4.18) and has a closed-form solution due to [4], however in our experiments for ease of implementation we use a numerical method to obtain solutions. We may either perform partial maximization in each iteration using the optimization method of [17] as in Figure 1 or approximate this using a faster batch update as shown in Figure 2. For both approaches, we use a deterministic annealing algorithm[18] which obtains hard-assignments.

5 Information Bottleneck with Side Information [IBwS]

A state-of-the-art existing approach, the IBwS algorithm [5], provides a convenient and elegant way of introducing model-level constraints in optimizing the following objective function:

$$(5.19) \quad \min_{p(c_k|x_i)} I(C; X) - \beta (I(C; Y^+) - \gamma I(C; Y^-)).$$

$I(C; X)$ measures the compactness of the representation of instances X by C . The $I(C; Y^+)$ term measures how well the representation retains information about Y^+ and the $I(C; Y^-)$ term penalizes information which C conveys about Y^- . The value for γ controls the trade-off between retaining information about Y^+ and penalizing information about Y^- . Following an annealing approach as suggested in [20], β may be increased until a hard clustering is found.

⁴The derivation is simple and we have omitted it in the interests of space

6 Experiments

We report experiments on both synthetic and real datasets using CMBCbu [Fig. 1], CMBCpm [Fig. 2] and a deterministic annealing version of IBwS (dIBwS). Results from all experiments are evaluated using the normalized mutual information (NMI) [8], which for clustering C and true labeling L measures how much information C captures about L . It is computed as $NMI(C, L) = \frac{I(C, L)}{H(L)}$ and ranges from 0 [no information] to 1 [$C = L$].

6.1 Synthetic Data We generate synthetic datasets which contain multiple clusterings to test the ability of the algorithms to find those clusterings. In particular, we generate data with 4 underlying clusterings, $\{Q^{(1)}, Q^{(2)}, Q^{(3)}, Q^{(4)}\}$. The strength of the clustering is controlled by the number of features associated with it; 6 features are associated with $Q^{(1)}$, 5 with $Q^{(2)}$, 4 with $Q^{(3)}$ and 3 with $Q^{(4)}$, resulting in an 18-dimensional set. The resultant dataset contains 4 underlying clusterings ordered according to decreasing strength: $Q^{(1)} \succ Q^{(2)} \succ Q^{(3)} \succ Q^{(4)}$. Each $Q^{(l)}$ groups the data into 2 clusters where each cluster has a representative binary vector. Drawing an instance now consists of, for each of $\{Q^{(1)}, Q^{(2)}, Q^{(3)}, Q^{(4)}\}$, randomly selecting one of the two clusters and assigning the binary vector for that cluster. Noise is introduced by randomly flipping each feature with some probability p_{noise} . We have divided the experiments according to the two forms of prior knowledge.

6.1.1 Known Clusterings The first experiment evaluates the ability of the algorithms to find successive clusterings and is divided into three sessions. For Session 1, we assume that one clustering, $Q^{(1)}$, is known, for Session 2 we assume that $Q^{(1)}$ and $Q^{(2)}$ are known, and for Session 3, we assume that $Q^{(1)}, Q^{(2)}$, and $Q^{(3)}$ are known. In each session we consider datasets with p_{noise} ranging from 0.1 to 0.3. The value of γ for each of the CMBC and IBwS algorithms is independently optimized over 20 possible settings for baseline setting $p_{noise} = 0.1$ and this value is retained for all other p_{noise} settings. Setting γ in this manner allows us to investigate the robustness with respect to γ of the algorithms if applied to different datasets. We will later investigate the ranges of effective γ for each of the p_{noise} . We compare performance against a deterministic annealing version of Expectation Maximization (EM) [7], which does not incorporate any prior knowledge. Results are shown in Table 2.

Uncovering Underlying Structure We first evaluate the algorithms over all three sessions for the baseline setting where $p_{noise} = 0.1$. Here we expect the

best performance as this is the setting for which the algorithms have been tuned. The EM algorithm does not incorporate any prior knowledge and so obtains the same solution across sessions. It typically discovers the most prominent clustering, $Q^{(1)}$. Of the 100 datasets, EM obtains a solution with $NMI(C, Q^{(1)}) > 0.75$ for 87 sets and $NMI(C, Q^{(2)}) > 0.75$ for 13. In none of the trials does EM obtain solutions with $NMI(C, Q^{(3)}) > 0.75$ or $NMI(C, Q^{(4)}) > 0.75$ which demonstrates the failure of random initialization to discover less-prominent clusterings. Performance among the CMBC and IBwS algorithms is approximately the same. While the performance of CMBCbu at finding the next most prominent clustering lags somewhat in Session 1, where there is a single known clustering, it improves relative to the other two algorithms in Sessions 2 and 3, where there are multiple known clusterings. The lower score of CMBCbu at discovering $Q^{(2)}$ in Session 1 occurs because in several of the trials, it instead discovers $Q^{(3)}$. This is not a failure of the algorithm; it is successfully accomplishing the task of discovering a novel clustering, however it is not discovering the next most prominent clustering. We will discuss this phenomenon further when looking at other noise settings.

Examining results for higher noise settings, we find the performance of dIBwS drops dramatically. For example, consider Session 2 with $p_{noise} = 0.20$. CMBCbu finds solutions that average $NMI(C, Q^{(3)}) = 0.4782$, and CMBCpm that average $NMI(C, Q^{(3)}) = 0.5007$ whereas for dIBwS, $NMI(C, Q^{(3)}) = 0.0008$. The dIBwS algorithm is finding solutions similar to known clustering $Q^{(2)}$. This behaviour is consistent for the higher noise ($p_{noise} \geq 0.20$) settings throughout Sessions 2 and 3 where dIBwS fails to discover unknown clusterings. Interestingly, in these cases, dIBwS seems to consistently discover the weakest of the known clusterings, as can be seen by looking at Sessions 2 and 3. In Session 2, where $Q^{(2)}$ is the weakest known clustering, $NMI(C, Q^{(2)})$ is 0.8404 and 0.8601 for p_{noise} set to 0.20 and 0.30. In Session 3, where $Q^{(3)}$ is the weakest known clustering, $NMI(C, Q^{(3)})$ is 0.9800 and 0.7270 for p_{noise} set to 0.20 and 0.30. For all of these settings, the solutions obtained by CMBCbu and CMBCpm are most like the target clustering, whereas dIBwS largely fails to discover the target clustering.

In comparing CMBCbu and CMBCpm, there is not a clear winner. As we saw in the baseline setting of $p_{noise} = 0.10$, CMBCbu's performance relative to CMBCpm and dIBwS generally improves across sessions as there are more known clusterings. There does not, however, appear to be a clear trend within a given session as the noise is increased.

Finally, in Sessions 1 and 2, where there are multi-

Table 2: Mean NMI for 100 synthetic sets generated with 1000 instances according to the procedure in Section 6.1. Parameter settings are: $\alpha = 0.5$, $\gamma^{CMBCbu} = .97$, $\gamma^{CMBCpm} = .95$, $\gamma^{IBwS} = 2.5714$.

Session 1: $Q^{(1)}$ is known. Goal is discovery of $Q^{(2)}$, $Q^{(3)}$ or $Q^{(4)}$.

		Session 1: $\mathbf{y}^- = Q^{(1)}$			
p_{noise}	Algorithm	$NMI(C, Q^{(1)})$	$NMI(C, Q^{(2)})$	$NMI(C, Q^{(3)})$	$NMI(C, Q^{(4)})$
0.10	CMBCbu	0.0007	0.8653	0.0591	0.0006
	CMBCpm	0.0008	0.9297	0.0008	0.0006
	dIBwS	0.0008	0.9072	0.0173	0.0007
	EM	0.8089	0.1217	0.0007	0.0008
0.20	CMBCbu	0.0007	0.6136	0.0531	0.0008
	CMBCpm	0.0007	0.6599	0.0163	0.0007
	dIBwS	0.0107	0.6595	0.0114	0.0006
	EM	0.6443	0.0429	0.0012	0.0015
0.30	CMBCbu	0.0006	0.2546	0.0553	0.0048
	CMBCpm	0.0006	0.2796	0.0279	0.0025
	dIBwS	1.0000	0.0008	0.0006	0.0008
	EM	0.3129	0.0430	0.0022	0.0015

Session 2: $Q^{(1)}$ and $Q^{(2)}$ are known. Goal is discovery of $Q^{(3)}$ or $Q^{(4)}$.

		Session 2: $\mathbf{y}^- = Q^{(1)}, Q^{(2)}$			
p_{noise}	Algorithm	$NMI(C, Q^{(1)})$	$NMI(C, Q^{(2)})$	$NMI(C, Q^{(3)})$	$NMI(C, Q^{(4)})$
0.10	CMBCbu	0.0006	0.0008	0.8336	0.0009
	CMBCpm	0.0005	0.0006	0.8300	0.0008
	dIBwS	0.0006	0.0008	0.8235	0.0085
	EM	0.8089	0.1217	0.0007	0.0008
0.20	CMBCbu	0.0006	0.0008	0.4782	0.0498
	CMBCpm	0.0005	0.0007	0.5007	0.0317
	dIBwS	0.0108	0.8404	0.0833	0.0006
	EM	0.6443	0.0429	0.0012	0.0015
0.30	CMBCbu	0.0110	0.0009	0.1958	0.0205
	CMBCpm	0.0006	0.0006	0.1520	0.0283
	dIBwS	0.1407	0.8601	0.0008	0.0007
	EM	0.3129	0.0430	0.0022	0.0015

Session 3: $Q^{(1)}$, $Q^{(2)}$ and $Q^{(3)}$ are known. Goal is discovery of $Q^{(4)}$.

		Session 3: $\mathbf{y}^- = Q^{(1)}, Q^{(2)}, Q^{(3)}$			
p_{noise}	Algorithm	$NMI(C, Q^{(1)})$	$NMI(C, Q^{(2)})$	$NMI(C, Q^{(3)})$	$NMI(C, Q^{(4)})$
0.10	CMBCbu	0.0009	0.0006	0.0006	0.8176
	CMBCpm	0.0008	0.0006	0.0005	0.7997
	dIBwS	0.0007	0.0006	0.0005	0.8068
	EM	0.8089	0.1217	0.0007	0.0008
0.20	CMBCbu	0.0009	0.0005	0.0006	0.5220
	CMBCpm	0.0009	0.0005	0.0005	0.4351
	dIBwS	0.0006	0.0208	0.9800	0.0006
	EM	0.6443	0.0429	0.0012	0.0015
0.30	CMBCbu	0.0086	0.0013	0.0068	0.2107
	CMBCpm	0.0047	0.0008	0.0101	0.1160
	dIBwS	0.0393	0.1985	0.7270	0.0006
	EM	0.3129	0.0430	0.0022	0.0015

ple unknown clusterings, dIBwS almost always finds the next most prominent clustering whereas CMBCbu and CMBCpm occasionally discover less prominent clusterings (e.g. $Q^{(3)}$ and $Q^{(4)}$ in Sessions 1 and 2 in Table 2.) In general, the CMBC algorithms were more sensitive to initialization than dIBwS which often obtains the same solution regardless of initialization. This is despite the fact that all three algorithms are using a deterministic annealing framework.

Robustness of γ Parameter. While the performance of dIBwS deteriorates for $p_{noise} > 0.10$, performance can be improved by re-tuning γ . This was tested on Session 3 with results shown in Figures 4 and 3. It can easily be seen in Figure 3 that, with CMBCbu, an optimal γ value for one p_{noise} setting is successful on other p_{noise} settings. For example, $\gamma = .95$ is optimal for $p_{noise} = 0.10$, but also scores well for $p_{noise} = 0.20$ and $p_{noise} = 0.30$. On the other hand, the fact that the curves are not nested in Figure 4 indicates that dIBwS is more sensitive to the value of γ . For example, the optimal $\gamma = 2.6667$ on $p_{noise} = 0.10$ fails completely on $p_{noise} = 0.20$ and $p_{noise} = 0.30$.

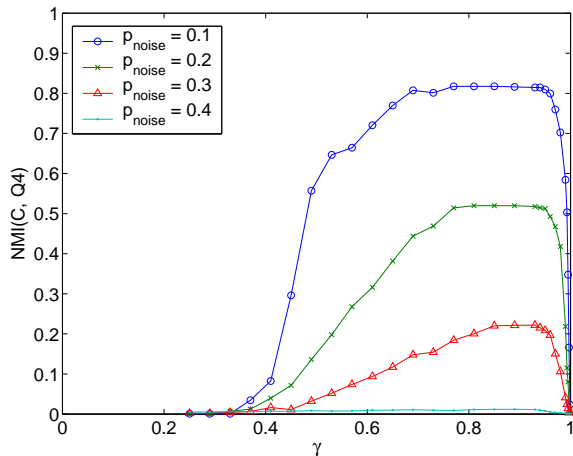


Figure 3: Similarity to target clustering $Q^{(4)}$ in Session 3 for varying γ , CMBCbu: optimal γ for one p_{noise} works for other p_{noise}

6.1.2 Negative Features Table 3 shows the results of a similar set of experiments using negative features instead of known clusterings. Half of the features associated with each clustering were set to be in \mathbf{y}^- . The use of multiple negative features means information which was previously part of the \mathbf{y}^+ is now instead part of the \mathbf{y}^- . For example, consider Session 1 where in the previous experiments \mathbf{y}^+ contained 6 features associated with the known clustering. In these experiments, 3 of those features are instead part of \mathbf{y}^- meaning \mathbf{y}^+

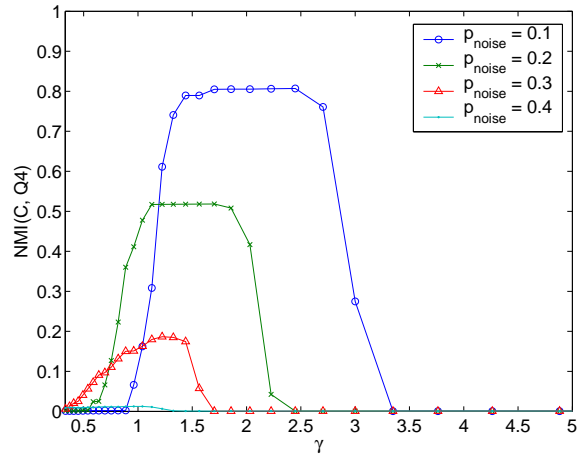


Figure 4: Similarity to target clustering $Q^{(4)}$ in Session 3 for varying γ , p_{noise} . dIBwS: optimal γ for one p_{noise} fails for other p_{noise}

now contains only 3 remaining features associated with the known clustering. Examining the baseline settings ($p_{noise} = 0.10$), the algorithms appear competitive except for Session 3, where CMBCpm substantially underperforms CMBCbu and dIBwS. As in the previous set of experiments, we note for Session 1 that the CMBC techniques do occasionally discover the less prominent clusterings, as evidenced by the fact that $NMI(C, Q^{(3)})$ is 0.0333 and 0.0253 for CMBCbu and CMBCpm whereas it is 0.0008 for dIBwS. This phenomenon is consistent across noise settings for Session 1, however in Session 2, the results are mixed. In Session 2, dIBwS also discovers less prominent clusterings.

The most striking difference between these results and the results in the previous section is that the performance of dIBwS does not deteriorate strongly as the noise increases. In fact, in Sessions 1 and 3, the dIBwS continues to outperform CMBCpm and CMBCbu as the noise increases, e.g. in Session 1, $p_{noise} = 0.30$, the $NMI(C, Q^{(2)})$ is 0.3245 for dIBwS while for CMBCpm and CMBCbu it is 0.3064 and 0.2148. In Session 3, $p_{noise} = 0.30$, $NMI(C, Q^{(3)})$ is 0.1488 for dIBwS while for CMBCbu and CMBCpm it is 0.1221 and 0.0578. In these experiments where noise features are given, dIBwS does not share the same sensitivity to parameter settings as in the known clustering experiments.

6.2 Real Data In this section we report our experiments on a sub-set of the RCV-1 corpus [19]. We define three collections, each with two different clusterings based on *Topic* and *Region*. E.g. in reut2x2a [described below], documents can be clustered into *Topic* cate-

Table 3: Mean NMI for 100 synthetic sets generated with 1000 instances according to the procedure in Section 6.1. Parameter settings are: $\alpha = 0.5$, $\gamma^{CMBCbu} = .9$, $\gamma^{CMBCpm} = .9$, $\gamma^{IBwS} = 1.2222$.

Session 1: Some features associated with $Q^{(1)}$ are known. Goal is discovery of $Q^{(2)}$, $Q^{(3)}$ or $Q^{(4)}$.

Session 1: $\mathbf{y}^- = 50\%$ of the features associated with $Q^{(1)}$					
p_{noise}	Algorithm	$NMI(C, Q^{(1)})$	$NMI(C, Q^{(2)})$	$NMI(C, Q^{(3)})$	$NMI(C, Q^{(4)})$
0.10	CMBCbu	0.0008	0.8923	0.0333	0.0006
	CMBCpm	0.0008	0.9012	0.0253	0.0007
	dIBwS	0.0008	0.9289	0.0008	0.0007
0.20	CMBCbu	0.0007	0.6554	0.0204	0.0007
	CMBCpm	0.0032	0.6473	0.0155	0.0030
	dIBwS	0.0008	0.6799	0.0006	0.0006
0.30	CMBCbu	0.0025	0.3064	0.0229	0.0012
	CMBCpm	0.0062	0.2148	0.0391	0.0089
	dIBwS	0.0009	0.3245	0.0044	0.0006

Session 2: Some features associated with $Q^{(1)}$ and $Q^{(2)}$ are known. Goal is discovery of $Q^{(3)}$ or $Q^{(4)}$.

Session 2: $\mathbf{y}^- = 50\%$ of the features associated with $Q^{(1)}, Q^{(2)}$					
p_{noise}	Algorithm	$NMI(C, Q^{(1)})$	$NMI(C, Q^{(2)})$	$NMI(C, Q^{(3)})$	$NMI(C, Q^{(4)})$
0.10	CMBCbu	0.0007	0.0009	0.8302	0.0006
	CMBCpm	0.0007	0.0009	0.8115	0.0167
	dIBwS	0.0007	0.0008	0.8224	0.0085
0.20	CMBCbu	0.0010	0.0011	0.5104	0.0269
	CMBCpm	0.0323	0.0042	0.4453	0.0299
	dIBwS	0.0007	0.0008	0.5269	0.0112
0.30	CMBCbu	0.0063	0.0012	0.2115	0.0103
	CMBCpm	0.0212	0.0017	0.1367	0.0215
	dIBwS	0.0007	0.0006	0.1153	0.0191

Session 3: Some features associated with $Q^{(1)}$, $Q^{(2)}$ and $Q^{(3)}$ are known. Goal is discovery of $Q^{(4)}$.

Session 3: $\mathbf{y}^- = 50\%$ of the features associated with $Q^{(1)}, Q^{(2)}, Q^{(3)}$					
p_{noise}	Algorithm	$NMI(C, Q^{(1)})$	$NMI(C, Q^{(2)})$	$NMI(C, Q^{(3)})$	$NMI(C, Q^{(4)})$
0.10	CMBCbu	0.0008	0.0006	0.0006	0.8175
	CMBCpm	0.0459	0.0220	0.0128	0.6931
	dIBwS	0.0008	0.0006	0.0006	0.8047
0.20	CMBCbu	0.0173	0.0005	0.0006	0.5026
	CMBCpm	0.0979	0.0137	0.0119	0.3017
	dIBwS	0.0035	0.0006	0.0006	0.5100
0.30	CMBCbu	0.0711	0.0030	0.0017	0.1221
	CMBCpm	0.0637	0.0125	0.0145	0.0578
	dIBwS	0.0184	0.0006	0.0037	0.1488

gories [MCAT, ECAT] or into *Region* categories [UK, India].

reut2x2a: *Topic* = [MCAT (Markets), GCAT (Government/Social)], *Region* = [UK, India], 3400 documents, 2513 terms.

reut2x2b: *Topic* = [I35 (Motor Vehicles and Parts), I81 (Banking and Financial Services)], *Region* = [Germany, USA], 1547 documents, 2235 terms.

reut2x2c: *Topic* = [GPOL (Domestic Politics), GDIP (International Relations)], *Region* = [France, UK], 800 documents, 2194 terms.

For each collection, stopwords were removed and a frequency cutoff was used to filter out the low frequency terms. We then evaluated the algorithms according to two scenarios. Each scenario consists of assuming one of either *Topic* or *Region* clusterings is known and then evaluating the algorithms on their ability to find the other clustering. Tables 4 and 5 show the best of 10 random initializations for each of the algorithms.

The experiments on real data confirmed the patterns observed with synthetic data. Specifically, CMBCbu and CMBCpm outperform dIBwS on all experiments except for the reut2x2b set in Scenario 2 where CMBCpm and dIBwS tie and dIBwS scores better than CMBCbu by 1.28x. For all other experiments the median performance gain with respect to dIBwS is 1.54x for CMBCpm and 1.66x for CMBCbu. Performance of CMBCbu and CMBCpm is considerably stronger than dIBwS in Scenario 2 - reut2x2a where CMBCbu has a NMI of 0.7576 and CMBCpm has a NMI of 0.7621 but dIBwS has a NMI of only 0.1600. In Scenario 1 - reut2x2c, CMBCbu and CMBCpm outperform dIBwS and find reasonably high quality solutions with NMIs of 0.3162 and 0.3257 for CMBCbu and CMBCpm respectively while dIBwS scores 0.1903, further showing that the outperformance of CMBCbu and CMBCpm with respect to dIBwS can be substantial.

We now consider the only experiment where CMBCbu and CMBCpm did not outperform dIBwS. On Scenario 2 - reut2x2b, dIBwS, with a NMI of 0.0934 outscores CMBCbu (NMI = 0.0729) and ties with CMBCpm, also with a NMI of 0.0934. In fact, dIBwS obtains the same clustering in both scenarios, highlighting a difficulty with using extrinsic measures like NMI to evaluate success. Namely, there may be other prominent structure in the dataset which is not associated with our known categorizations. Then, the algorithm may discover this unassociated structure in both scenarios.

The presence of other prominent structures is a possible explanation for for the low NMI scores in Scenario

1 - reut2x2a where CMBCbu (NMI = 0.0189) and CMBCpm (NMI = 0.0014) beats dIBwS (NMI = 0.0002). Clearly the algorithms are not discovering the clustering we expect. indicating that the “Topic” clustering we have in mind may not prominent in the data. This hypothesis is a reasonable explanation for this dataset which includes substantially more documents (n=3400) than the other two (n=1547 and n=800.) We intend to study these effects further to better determine the causes of the poor performance.

Finally, these experiments confirm there is no penalty associated with using the batch update approach of CMBCbu instead of the partial maximization approach of CMBCpm. This is an advantageous finding as the runtime of the CMBCbu algorithm is comparable with dIBwS whereas CMBCpm incurs a greater computational cost.

7 Conclusions

In this paper we have addressed the problem of systematically uncovering multiple clusterings underlying a dataset. We have proposed a constrained likelihood model and an associated EM algorithm with two variants [CMBCbu and CMBCpm]. These algorithms, along with an existing state-of-the-art information bottleneck variant [dIBwS], have been evaluated on both synthetic and real data in their ability to systematically uncover new clusterings. On synthetic data, all algorithms showed the ability to uncover multiple underlying clusterings even under noise. The performance of CMBCbu and CMBCpm was more robust to noise with respect to parameter settings than dIBwS. With increasing amounts of information the performance of the dIBwS algorithm was comparable to that of CMBCbu and CMBCpm.

The experiments on real data confirmed the patterns observed with synthetic data. In the high-dimensional, high-noise and minimal-information setting of real data, the CMBC methods outperformed dIBwS, where only in one case did dIBwS not come in last. However, the results of CMBCpm come at a higher computational cost while the CMBCbu algorithm, on the other hand, is comparable in runtime to the dIBwS algorithm. Notably, there appeared to be no penalty for using the batch update approach of CMBCbu instead of partial maximization as in CMBCpm.

The CMBC methods rely on several assumptions, namely: clusters have roughly equal probability, the prior knowledge is generated according to a mixture of relevant features, and features are binary. It remains to be seen whether these methods may be modified to relax these assumptions. In general, however, the results are very encouraging and this direction of research certainly

Table 4: Scenario 1: Topic clustering is known. NMI of best of 10 random initializations on the Reuters RCV1 sets. The goal is to find the Region clustering. We use $\delta = 0.5$ and γ is tuned for each set. Highest scores are in bold.

dataset	Algorithm	$y^- = Topic$	
		$NMI(C; Topic)$	$NMI(C; Region)$
reut2x2a	CMBCbu($\gamma = .997$)	0.0020	0.0189
	CMBCpm($\gamma = .99$)	0.0151	0.0014
	dIBwS ($\gamma = 4.8824$)	0.0150	0.0002
reut2x2b	CMBCbu($\gamma = .99$)	0.0086	0.1245
	CMBCpm ($\gamma = .95$)	0.0047	0.1155
	dIBwS ($\gamma = 3.1667$)	0.0934	0.0748
reut2x2c	CMBCbu($\gamma = .99$)	0.0019	0.3162
	CMBCpm ($\gamma = .95$)	0.0031	0.3257
	dIBwS ($\gamma = 24$)	0.0879	0.1903

Table 5: Scenario 2: Region clustering is known. NMI of best of 10 random initializations on the Reuters RCV1 sets. The goal is to find the Topic clustering. We use $\delta = 0.5$ and γ is tuned for each set. Highest scores are in bold.

dataset	Algorithm	$y^- = Region$	
		$NMI(C; Topic)$	$NMI(C; Region)$
reut2x2a	CMBCbu($\gamma = .997$)	0.7576	0.0001
	CMBCpm($\gamma = .99$)	0.7621	0.0001
	dIBwS ($\gamma = 4.8824$)	0.1600	0.0000
reut2x2b	CMBCbu($\gamma = .99$)	0.0729	0.0082
	CMBCpm($\gamma = .95$)	0.0934	0.0748
	dIBwS ($\gamma = 4.8824$)	0.0934	0.0748
reut2x2c	CMBCbu($\gamma = .997$)	0.4382	0.0001
	CMBCpm($\gamma = .99$)	0.4570	0.0001
	dIBwS ($\gamma = 4.8824$)	0.4294	0.0017

warrants further study.

References

- [1] S. Basu, A. Banerjee, and R. J. Mooney. Semi-supervised clustering by seeding. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 27–34. Morgan Kaufmann Publishers Inc., 2002.
- [2] S. Basu, M. Bilenko, and R. J. Mooney. A probabilistic framework for semi-supervised clustering. In *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 59–68. ACM Press, 2004.
- [3] M. Bilenko, S. Basu, and R. J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *Twenty-first international conference on Machine learning*. ACM Press, 2004.
- [4] G. Cardano. *Ars Magna*. Nurnberg, 1545.
- [5] G. Chechik and N. Tishby. Extracting relevant structures with side information. In *Advances in Neural Information Processing Systems 15*, 2002.
- [6] D. Cohn, R. Caruana, and A. McCallum. Semi-supervised clustering with user feedback, 2003.
- [7] A. Dempster, N. Laird, and D. B. Rubin. Maximum likelihood from incomplete data. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.
- [8] X. Z. Fern and C. E. Brodley. Random projection for high dimensional data clustering: A cluster ensemble approach. In *The 20th International Conference on Machine Learning*, 2003.
- [9] D. Gondek and T. Hofmann. Non-redundant data clustering. In *4th IEEE International Conference on Data Mining*, 2004.
- [10] P. Harremoës and F. Topsøe. Details for inequalities between entropy and index of coincidence derived from information diagrams. *IEEE Transactions on Information Theory*, 47:2944–2960, 2001.
- [11] J. H. Havrda and F. Charvát. Quantification methods of classification processes: Concepts of structural entropy. In *Kybernetika*, 3, 1967.
- [12] T. Hertz, A. Bar-Hillel, and D. Weinshall. Boosting margin based distance functions for clustering. In *Twenty-first international conference on Machine learning*. ACM Press, 2004.

- [13] A. K. Jain and R. C. Dubes. *Algorithms for clustering data*. Prentice-Hall, 1988.
- [14] D. Klein, S. D. Kamvar, and C. D. Manning. From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In *Proceedings of the 19th International Conference on Machine Learning*, 2002.
- [15] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297, Berkeley, 1967.
- [16] R. Neal and G. Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In M. I. Jordan, editor, *Learning in Graphical Models*. Kluwer, 1998.
- [17] M. J. D. Powell. *A Fortran Subroutine for Solving Systems of Nonlinear Algebraic Equations*, chapter 7. 1970.
- [18] K. Rose. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems, 1998.
- [19] T. Rose, M. Stevenson, and M. Whitehead. The Reuters corpus volume 1 – from yesterday’s news to tomorrow’s language resources. In *Proceedings of the Third International Conference on Language Resources and Evaluation*, 2002.
- [20] N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method. In *Proc. of the 37th Annual Allerton Conference on Communication, Control and Computing*, pages 368–377, 1999.
- [21] F. Topsøe. Entropy and index of coincidence, lower bounds. preprint, 2003.
- [22] K. Wagstaff and C. Cardie. Clustering with instance-level constraints. In *Proc. of the 17th International Conference on Machine Learning*, pages 1103–1110, 2000.
- [23] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems*, 15, 2003.

A Computation of $p(\mathbf{y}_i^+ | c_k)$, $p(\mathbf{y}_i^- | \mathbf{y}_h^+)$ and $p(\mathbf{y}_i^- | c_k)$

A.1 Computation of $p(\mathbf{y}_i^+ | c_k)$ Assuming class-conditional independence of the binary features:

$$(A.1) \quad p(\mathbf{y}_i^+ | c_k) = \prod_{j=1}^{m^+} p(y_{ij}^+ | c_k)$$

$$(A.2) \quad = \prod_{j=1}^{m^+} p(y_{ij}^+ | c_k)^{y_{ij}^+} (1 - p(y_{ij}^+ | c_k))^{1 - y_{ij}^+}.$$

A.2 Computation of $p(\mathbf{y}_i^- | \mathbf{y}_h^+)$ Computing $p(\mathbf{y}_i^- | c_k)$ [described below] requires computing $p(\mathbf{y}_i^- | \mathbf{y}_h^+)$. To address the sparsity of \mathbf{y}_h^+ in the

observed data and to ensure tractability, we assume \mathbf{y}_i^- is generated according to a mixture of the features y_{hj}^+ where we assume that $p(y_{hj}^+)$ is uniform:

(A.3)

$$(A.4) \quad p(\mathbf{y}_i^- | \mathbf{y}_h^+) = \sum_{j=1}^{m^+} p(\mathbf{y}_i^- | y_{hj}^+) p(y_{hj}^+) = \frac{1}{m^+} \sum_{j=1}^{m^+} p(\mathbf{y}_i^- | y_{hj}^+) \\ = \frac{1}{m^+} \sum_j p(\mathbf{y}_i^- | y_{.j}^+ = 1)^{y_{hj}^+} p(\mathbf{y}_i^- | y_{.j}^+ = 0)^{1 - y_{hj}^+},$$

where (A.4) follows from the fact that the data is binary. The $p(\mathbf{y}_i^- | y_{.j}^+ = 1)$ and $p(\mathbf{y}_i^- | y_{.j}^+ = 0)$ are estimated from data using

(A.5)

$$p(\mathbf{y}_i^- | y_{.j}^+ = b) = \frac{|\{\mathbf{y}_t \in \mathcal{Y} : \mathbf{y}_t^- = \mathbf{y}_i^- \text{ and } y_{tj}^+ = b\}|}{|\{\mathbf{y}_t \in \mathcal{Y} : y_{tj}^+ = b\}|},$$

for $b \in \{0, 1\}$.

E.g. for the data in Table 1,

$$p([1 \ 0] | y_{.1}^+ = 1) = 2/4, \\ p([1 \ 0] | y_{.2}^+ = 1) = 4/5, \\ p([1 \ 0] | y_{.3}^+ = 1) = 4/4.$$

A.3 Computation of $p(\mathbf{y}_i^- | c_k)$ The $p(\mathbf{y}_i^- | \mathbf{y}_h^+)$ are fixed, so the $p(\mathbf{y}_i^- | c_k)$ are consequences of the choice of $p(y_{.j}^+ | c_k)$ as can be seen in Lemma A.1.

LEMMA A.1. *The $p(\mathbf{y}_i^- | c_k)$ may be expanded as:*

$$(A.6) \quad p(\mathbf{y}_i^- | c_k) = \frac{1}{m^+} \sum_j \sum_{y_{.j}^+ \in \{0,1\}} p(\mathbf{y}_i^- | y_{.j}^+) p(y_{.j}^+ | c_k).$$

Sketch of proof By marginalizing over all possible \mathbf{y}_h^+ we obtain: $p(\mathbf{y}_i^- | c_k) = \sum_{\mathbf{y}_h^+ \in Y^+} p(\mathbf{y}_i^- | \mathbf{y}_h^+) p(\mathbf{y}_h^+ | c_k)$. Substituting the definitions from (A.2) and (A.4), restating the summation as used in (A.6), grouping terms, and simplifying the expression produces the final result.

B Derivation of Class-Conditional M-Step Equations

LEMMA B.1. *The $p(y_{.j}^+ | c_k)$ are maximized for solutions to the cubic equation:*

$$F_3 p(y_{.h}^+ | c_k)^3 + F_2 p(y_{.h}^+ | c_k)^2 + F_1 p(y_{.h}^+ | c_k) + F_0 = 0.$$

where

$$F_3 = \frac{-2\gamma p(c_k)^2}{m^{+2}} (Q \ln |Y^-| - \alpha_{|Y^-|}) A_2(h, h),$$

$$F_2 = \frac{2\gamma p(c_k)^2}{m^{+2}} (Q \ln |Y^-| - \alpha_{|Y^-|}) A_2(h, h) + \frac{2 \ln |Y^-| \cdot Q p(c_k)}{m^{+2}} \left(\sum_{r,l:(r,l) \neq (k,h)} p(c_r) A_2(h, l) \theta_{lk} + A_1(h) \right) - \frac{2 \ln |Y^-| \cdot \beta_{|Y^-|} p(c_k)^2}{m^{+2}} \left(\sum_{l:l \neq h} A_2(h, l) \theta_{lk} + A_1(h) \right),$$

$$F_1 = - (1 - \gamma) \sum_{\mathbf{y}_i \in \mathcal{Y}} q(c_k | \mathbf{y}_i) + \frac{2 \ln |Y^-| \cdot Q p(c_k)}{m^{+2}} \left(\sum_{r,l:(r,l) \neq (k,h)} p(c_r) A_2(h, l) \theta_{lk} + A_1(h) \right) - \frac{2 \ln |Y^-| \cdot \beta_{|Y^-|} p(c_k)^2}{m^{+2}} \left(\sum_{l:l \neq h} A_2(h, l) \theta_{lk} + A_1(h) \right),$$

$$F_0 = (1 - \gamma) \sum_{\mathbf{y}_i \in \mathcal{Y}} q(c_k | \mathbf{y}_i) y_{ih}^+.$$

The A terms are defined as:

$$A_2(j, l) = \sum_{\mathbf{y}_i^- \in \mathcal{Y}^-} (p(\mathbf{y}_i^- | y_j^+ = 1) - p(\mathbf{y}_i^- | y_j^+ = 0)) \cdot (p(\mathbf{y}_i^- | y_l^+ = 1) - p(\mathbf{y}_i^- | y_l^+ = 0)),$$

$$A_1(j) = \sum_{\mathbf{y}_i^- \in \mathcal{Y}^-} p(\mathbf{y}_i^- | y_l^+ = 0) \cdot (p(\mathbf{y}_i^- | y_j^+ = 1) - p(\mathbf{y}_i^- | y_j^+ = 0)),$$

$$A_0 = \sum_{\mathbf{y}_i^- \in \mathcal{Y}^-} \sum_{j,l=1}^{m^+} p(\mathbf{y}_i^- | y_j^+ = 0) p(\mathbf{y}_i^- | y_l^+ = 0).$$

B.1 Expanding the $p(\mathbf{y}_i^- | c_k)$ terms in the $\tilde{H}^1(C, Y^-)$ term from (3.14) Recall that $\tilde{H}^1(C, Y^-)$ is defined in (3.13) in terms of $p(\mathbf{y}_i^- | c_k)$. The parameters for the model, however, are $p(\mathbf{y}_i^+ | c_k)$. We now expand the $p(\mathbf{y}_i^- | c_k)$ according to (A.6) in order to obtain expressions explicitly in terms of $p(\mathbf{y}_i^- | c_k)$. We will need

the following quantity which appears in $\tilde{H}^1(C, Y^-)$:

$$(B.1) \quad \sum_{\mathbf{y}_i^- \in \mathcal{Y}^-} \sum_{k=1}^K p(\mathbf{y}_i^-, c_k) \cdot p(\mathbf{y}_i^-, c_k) = \sum_{\mathbf{y}_i^- \in \mathcal{Y}^-} \sum_{k=1}^K p(c_k)^2 \frac{1}{m^{+2}} \sum_{j=1}^{m^+} \sum_{l=1}^{m^+} [(p_{ij}^1 - p_{ij}^0)(p_{il}^1 - p_{il}^0) \cdot \theta_{jk} \theta_{lk} + (p_{il}^0(p_{ij}^1 - p_{ij}^0)) \theta_{jk} + (p_{ij}^0(p_{il}^1 - p_{il}^0)) \theta_{lk} + (p_{ij}^0 p_{il}^0)]$$

where we have substituted $p_{ij}^0 = p(\mathbf{y}_i^- | y_j^+ = 0)$, $p_{ij}^1 = p(\mathbf{y}_i^- | y_j^+ = 1)$ and $\theta_{jk} = p(y_j^+ | c_k)$ for simplification. To further simplify the expression, we introduce constants A_0, A_1, A_2 according to the definitions in (B.1) and obtain the expanded expression for $\tilde{H}^1(C, Y^-)$:

$$(B.2) \quad \tilde{H}^1(C, Y^-) = \alpha_{|Y^-|} - \beta_{|Y^-|} \sum_{k=1}^K p(c_k)^2 \frac{1}{m^{+2}} \cdot \left\{ \sum_{j,l} A_2(j, l) \theta_{jk} \theta_{lk} + 2 \sum_j A_1(j) \theta_{jk} + A_0 \right\}.$$

B.2 Expanding the $p(\mathbf{y}_i^- | c_k)$ terms in the $H(Y^-)$ term of (3.14) In a similar fashion, one obtains the expanded version of $\tilde{H}^u(Y^-)$ from (3.13):

$$(B.3) \quad \tilde{H}^1(Y^-) = (\ln |Y^-|) \cdot \left(1 - Q \sum_{k=1}^K \sum_{r=1}^K p(c_k) p(c_r) \frac{1}{m^{+2}} \cdot \left\{ \sum_{j,l} A_2(j, l) \theta_{jk} \theta_{lr} + 2 \sum_j A_1(j) \theta_{jk} + A_0 \right\} - \frac{Q}{|Y^-|} \right).$$

B.3 Finding $p(\mathbf{y}_i^+ | c_k)$ which maximize (3.14) The critical points of (3.14) may be obtained as follows: First, the expanded versions given in (B.2) and (B.3) are substituted into (3.14). Next, the standard variational approximation for EM is applied [for details, see [16]]. Finally, the result is differentiated with respect to θ_{hk} and equated to zero. The result is a cubic equation. This is because the $p(\mathbf{y}_i^+ | c_k)$ are linear in the derivatives of (B.2) and (B.3). The y_{ij}^+ appear as the exponent of $p(\mathbf{y}_i^+ | c_k)$ in (A.2) which itself appears in the first term of (3.14). Since the y_{ij}^+ are binary, we are guaranteed that after differentiating we obtain a cubic equation. Using the fact that $A_2(j, l) = A_2(l, j)$, and grouping terms results in the cubic equation in Lemma B.1.