

Probabilistic Multi-State Split-Merge Algorithm for Coupling Parameter Estimates

Juan K. Lin

Department of Statistics
Rutgers University
Piscataway, NJ 08854
jklin@stat.rutgers.edu

Abstract

A new approach to finding good local maxima of the likelihood function based on synthesizing information from two local maxima is presented. We investigate the coupled EM algorithm (CoEM) for coupling local maxima solutions from two separate EM runs for the multinomial mixture model. The CoEM algorithm probabilistically splits and merges multiple latent states based on conditional independence assumptions and is numerically shown to significantly improve on uncoupled EM or deterministic annealing (DAEM) parameter estimates.

1 Introduction

The EM algorithm (e.g. Dempster, Laird and Rubin 1977) and its variants are fundamental algorithmic building blocks for parameter estimation in latent variable models. The monotonic convergence property of the EM algorithm, as well its ease of derivation and implementation has made it the principal algorithm for model parameter estimation in the machine learning and data mining communities. The monotonic convergence property guarantees convergence to a local maxima. However, when the likelihood function contains many local maxima, how does one go about fitting the model parameters? A simple strategy is to run the EM algorithm with various initial conditions to map out the local maxima in the likelihood function, and simply choose the best local maxima. This is computationally costly especially in high dimensional data and parameter space. Researchers have tackled the local maxima problem in two ways. One strategy is to modify the likelihood cost function landscape. In analogy with a physical annealing process, the deterministic annealing EM algorithm (DAEM) (Rose et.al. 1990, Ueda et.al. 1998) modifies the likelihood function by adding a temperature control parameter, and explores various annealing schedules to try to find good local maxima. Similarly, the information bottleneck EM algorithm (IB-EM) in-

troduces a general class of cost functions which trade-off information compression and preservation (Elidan and Friedman 2003). A second strategy is to use various criteria for selectively splitting and merging clusters to try to escape from poor local maxima (e.g. Brown et.al. 1992, Ueda et.al. 1998, Jain and Neal 2004). Instead of splitting and merging pairs of states, we present an approach which maps all the latent states of two local maxima parameter estimates to each other. The computational goal is to be able to run multiple EM trials in parallel from different initial conditions, and to synthesize the information from multiple local maxima into better parameter estimates. We investigate the coupled EM algorithm (CoEM) for various multinomial mixture models. Recently, these models have received great interest in the data mining and machine learning communities (e.g. Lee and Seung 1999, Hofmann 2001, Lin 2003, Ding and He 2005).

An outline of the paper is as follows. In Sections 2 and 3, the intuition for combining clusters and the CoEM algorithm for the multinomial mixture model is described. Numerical results are presented in Section 4. The CoEM algorithm for a more complex multinomial mixture traffic model is described in Section 5.

2 Intuition

Our main goal is to find a method of combining information contained in multiple EM *soft* clustering solutions. Some hard classification examples will illustrate the intuition. Given objects in the set $\{NYC, Boston, apple, orange, red, blue\}$, let the first partition solution be $\pi_1 = \{NYC, Boston, apple, orange\}, \{red, blue\}$, and the second partition $\pi_2 = \{NYC, Boston\}, \{apple, orange, red, blue\}$. Even though the two partitions are not optimal, they contain information about the correct underlying classes $\pi_1 \wedge \pi_2 = \{NYC, Boston\}, \{apple, orange\}, \{red, blue\}$. Here $\pi_1 \wedge \pi_2$ denotes the combinatorial meet of the two partitions (e.g. Stanley 1986). Consider a second example of combining

classification information. Let the two partitions be $\pi_1 = \{\text{red, green}\}, \{\text{blue, yellow}\}, \{\text{pacific, atlantic}\}$, and $\pi_2 = \{\text{green, yellow}\}, \{\text{red, blue}\}, \{\text{pacific, atlantic}\}$. The combinatorial join of the two partitions is given by $\pi_1 \vee \pi_2 = \{\text{red, green, blue, yellow}\}, \{\text{pacific, atlantic}\}$, which is the correct underlying classification.

In the first example, taking the meet of the partitions correctly splits up the fruit cluster $\{\text{apple, orange}\}$ from their respective incorrect clusters. In the second example, taking the join of the partitions correctly merges all the colors into one class $\{\text{red, green, blue, yellow}\}$. These examples illustrate some intuition behind how two classification solutions can be combined to form more suitable partitions. Combining information from multiple partitions has been investigated from a lattice theoretic perspective in Neumann and Norton (1986) and Barthelemy et.al. (1986), and from a mutual information perspective in Strehl and Ghosh (2002). In this paper we investigate algorithms for combining two *soft* probabilistic clustering solutions.

3 Coupling EM runs

3.1 Summary of the multinomial mixture model

We wish to formulate an algorithm for combining the information from two EM solutions. In this section, we summarize the multinomial mixture model with one latent variable and a conditional independence assumption. This model has been applied to information retrieval and natural language processing (Brown et.al. 1992, Pereira et.al. 1993, Saul and Pereira 1997, Lee and Seung 1999, Hofmann 2001) and has appeared in the statistics literature as latent class analysis (Everitt 1984). The coupling of EM runs for more structured mixture of multinomials is described in a later section.

Let $\tilde{p}(x, y)$ be the empirically observed joint distribution over discrete random variables X and Y , and let H be a discrete 'class' latent variable. Let the number of states in the random variables be $|X| = |Y| = n$ and $|H| = k$. The model assumes that X and Y are conditionally independent given H , which we write $X \perp Y | H$. Maximizing the likelihood is equivalent to minimizing the following Kullback-Leibler divergence

$$\mathcal{D}(\tilde{p}(x, y) \parallel \sum_h p(x|h)p(y|h)p(h)).$$

with respect to $p(x|h), p(y|h)$ and $p(h)$.

3.2 Initial attempts at coupling EM runs

Suppose two EM algorithms are run in parallel with two different initial conditions. Let H_1 and H_2 be the

latent variables in the two runs. The conditional independence assumptions for the two separate models, $\{X \perp Y | H_1, X \perp Y | H_2\}$, together with the two sets of model parameters specify the marginals $p(x, y, h_1)$ and $p(x, y, h_2)$. We tried embedding the two mixture models of the marginals $p(x, y, h_1)$ and $p(x, y, h_2)$ into a full model of $p(x, y, h_1, h_2)$. First consider the undirected graphical model with edges between X and H_1 , X and H_2 , Y and H_1 , and Y and H_2 . This model graphically links the two underlying graphical models together at the observed variables X and Y . Two undesirable properties of this model are immediately apparent. First this is a loopy graphical model; second, this model's conditional independence assumptions, $X \perp Y | \{H_1, H_2\}$ and $H_1 \perp H_2 | \{X, Y\}$, are not directly consistent with the two multinomial mixture models' conditional independence assumptions. Our second attempt at coupling the mixture models seeks a model of the full joint distribution consistent with the two mixture models. A simple way of modeling $p(x, y, h_1, h_2)$ is to assume $H_1 \perp H_2 | X, Y$, in addition to the two conditional independence assumptions for the two multinomial mixture models $X \perp Y | H_1$, and $X \perp Y | H_2$. This new conditional independence assumption is consistent with the assumptions from the two mixture models since it simply defines the joint as

$$p(x, y, h_1, h_2) = p(h_1|x, y)p(h_2|x, y)\tilde{p}(x, y).$$

Using maximum entropy to pick out a model of the full joint distribution $p(x, y, h_1, h_2)$ given pre-specified marginals $p(x, y, h_1)$ and $p(x, y, h_2)$ selects this exact model. Unfortunately, this is not a graphical model, and parameter estimation is a non-trivial challenge.

3.3 CoEM algorithm for two mixtures of multinomials

Instead of constructing a model for the full joint distribution and worrying about consistency of assumptions, we focus on one-time couplings of parameters from the two mixture models similar to a message passing update. For coupling EM runs, we simply perform a *one-time* "probabilistic split-merge" coupling of two converged EM run parameter estimates, then continue the EM runs separately until convergence. Thus the only difference between regular EM and a coupled EM algorithm is the *one-time* coupling step. This will be referred to as the CoEM algorithm.

We now address the issue of combining parameter estimates. Parameters from the two mixture models $\{p(x|h_1), p(y|h_1), p(h_1)\}$, and $\{p(x|h_2), p(y|h_2), p(h_2)\}$, are assumed to correspond to two local maxima of the likelihood function. To combine information from

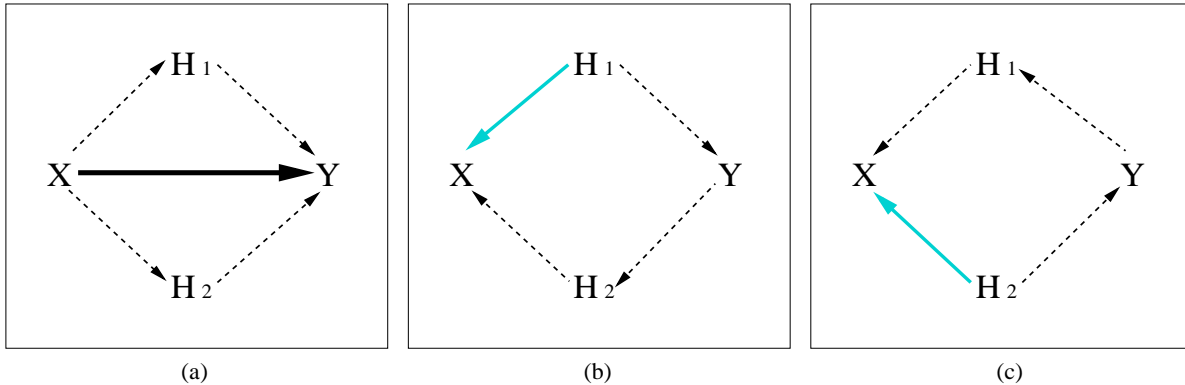


Figure 1: Approximate mapping diagram motivating the coupling updates in the CoEM algorithm. Note, this is not a graphical model diagram.

the two local maxima, we cannot simply take convex combinations of the corresponding parameters. The ordering of the states for H_1 and H_2 are generally not in direct correspondence with each other. Furthermore, some states in H_1 may be mixtures of the states in H_2 and vice versa. In order to compare one set of parameters to the other set, we need a mapping between the states in the two latent variables.

We investigated mappings between latent states for CoEM based purely on conditional independence assumptions. The CoEM algorithm for mixture of multinomials model is as follows:

-
- 1 **Iterate** two EM runs in parallel until convergence.
 - 2 **Couple:**
 - (a) Assume $H_1 \perp H_2 | Y$,
compute $p_a(h_1, h_2) = \sum_y p(h_1|y)p(h_2|y)p(y)$.
Update $p(x|h_1) = \sum_{h_2} p(x|h_2)p_a(h_2|h_1)$,
and $p(x|h_2) = \sum_{h_1} p(x|h_1)p_a(h_1|h_2)$.
 - (b) Assume $H_1 \perp H_2 | X$,
compute $p_b(h_1, h_2) = \sum_x p(h_1|x)p(h_2|x)p(x)$.
Update $p(y|h_1) = \sum_{h_2} p(y|h_2)p_b(h_2|h_1)$,
and $p(y|h_2) = \sum_{h_1} p(y|h_1)p_b(h_1|h_2)$.
 - 3 **Iterate** the two updated EM runs in parallel until convergence.
-

All conditionals are computed based on the corresponding specified joint distributions. Here the updates for parameters involving X are based on $H_1 \perp H_2 | Y$, while updates for parameters involving Y are based on $H_1 \perp H_2 | X$. This is to retain symmetry in the roles played by X and Y . Figure 1 depicts an approximate

mapping description of the multinomial mixture model, as well as the intuition behind the coupling algorithm. *Note*, these are not graphical model diagrams, as the conditional independence assumptions in CoEM do not exactly correspond to those contained in a DAG interpretation of the diagrams. In Figure 1(a), the dark arrow from X to Y denotes the observed empirical transition mapping $\tilde{p}(y|x)$ obtained from $\tilde{p}(x, y)$. The multinomial mixture model seeks to find compositional mappings from X to H (H_1 and H_2 for the two EM runs) and then Y which approximates $\tilde{p}(y|x)$ optimally in a minimum Kullback-Leibler divergence sense. The updates in CoEM step 2(a) for $p(x|h_1)$ and $p(x|h_2)$ are depicted in Figures 1(b) and (c) respectively.

Other choices of conditional independence assumptions for determining $p(h_1, h_2)$ are explored in sections 4.3 and 4.4. Note that these conditional independence coupling assumptions are used *only* in the coupling step. The **coupling** and **iterate** until convergence steps can be repeated until a satisfactory solution is reached. Except for possible parameters arising from the EM convergence criteria, there are no threshold parameters in the coupling step of the CoEM algorithm to set.

4 Numerical results

We performed numerical experiments comparing regular uncoupled EM runs, coupled EM runs (CoEM), and deterministic annealing EM algorithm runs (DAEM) (Ueda et.al. 1998). In the following subsections, parameter estimation results are reported for synthetically generated data, as well as computer skills, text corpus, and traffic data. Sections 4.2 and 4.3 compare KL-divergence minimization (equivalently likelihood maximization) results. Section 4.4 reports predictive test-set loglikelihood results. We begin with numerical experi-

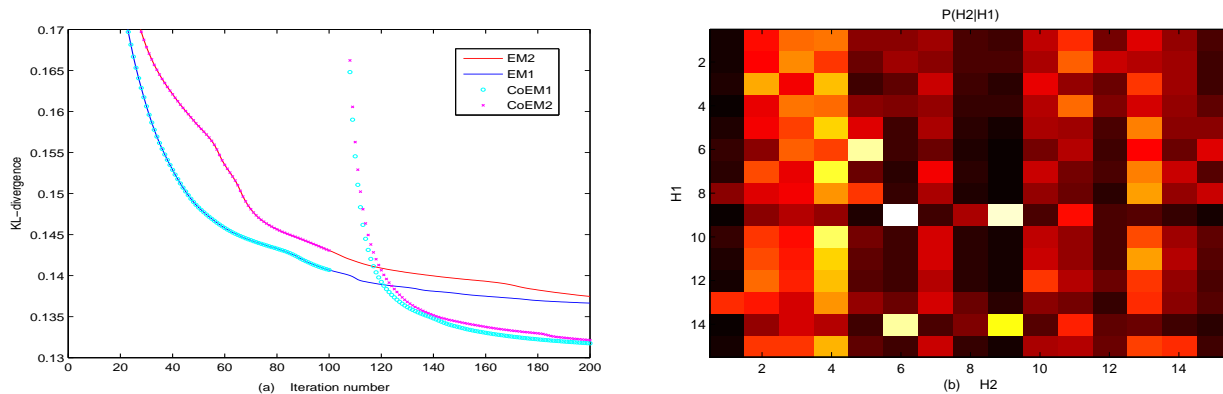


Figure 2: (a) KL-divergence as a function of iteration number for EM and CoEM runs. (b) Transition matrix $p(h_2|h_1)$ used after iteration 100 for CoEM algorithm.

ments on a small computer skills data set to demonstrate the coupling.

4.1 Computer skills data

We applied the CoEM multinomial mixture model algorithm to a computer skills co-occurrence dataset, courtesy of Prof. Richard Martin and the IT consulting firm Comrise. This data set is small and intuitive enough to permit a more detailed analysis of the CoEM algorithm. The raw data consists of a collection of job descriptions, each of which contains a set of computer skills the hiring manager considers important for the job. A co-occurrence matrix is constructed which tabulates the number of times each pair of 159 computer skills appear in a job description. The entries along the diagonal of the co-occurrence matrix contain the number of times each skill occurred over all the job descriptions. The multinomial mixture model was used to find computer skills topic clusters.

Two EM trials are run in parallel with 15 latent states for H for 100 iterations. Subsequently, the two trials are run for an additional 100 iterations both with and without a CoEM coupling step. The CoEM coupling step consists of the updates $p(x|h_2) = \sum_{h_1,y} p(x|h_1)p(h_1|y)p(y|h_2)$, and $p(x|h_1) = \sum_{h_2,y} p(x|h_2)p(h_2|y)p(y|h_1)$. The progression of the KL-divergence is shown in Figure 2(a). The KL-divergence for the EM and CoEM trials are identical for the first 100 iterations. For CoEM, the coupling step after iteration 100 increases the KL-divergence, though after a few additional iterations, the CoEM trials arrive at significantly lower KL-divergences compared with their uncoupled EM counterparts.

The coupling transition matrix $p(h_2|h_1) = \sum_x p(h_2|y)p(y|h_1)$ used after iteration 100 is depicted

in Figure 2(b). We focus on state $H_1 = 1$. From CoEM step 2(a), the updated $p(x|h_1 = 1)$ is a convex combination of the distributions $p(x|h_2)$ with weights given by the first row of the transition matrix in Figure 2(b). The coefficients in $p(h_2|h_1 = 1)$ has large contributions for $h_2 = \{3, 4, 11\}$. Though this is a soft probabilistic clustering model, we look at the MAP assigned states to help us understand how the CoEM algorithm couples the EM runs. MAP assignment for latent state $H_1 = 1$ (state 1 in EM run 1) gives the computer skills $\{pc(ibm), windows95, msoffice, dos\}$ after both 100 and 200 iterations for EM run 1. After 100 iterations, latent state $H_2 = 3$ contains skills $\{windowsnt, windows95, visualc++, visualbasic\}$, $H_2 = 4$ contains $\{sunos, solaris, unix\}$, and $H_2 = 11$ contains $\{pc(ibm), msoffice, msoffice97, msproject\}$. After 200 iterations, the CoEM run for H_1 gives MAP assigned skills for $H_1 = 1$ of $\{pc(ibm), windows95, windowsnt, dos, visualc++, visualbasic, vbscript\}$, while the uncoupled EM run 1 retains the same MAP assigned skills $\{pc(ibm), windows95, msoffice, dos\}$. The coupling step merged the skills $\{windowsnt, visualc++, visualbasic, vbscript\}$ into cluster $H_1 = 1$, and split off the skill $\{msoffice\}$ into a new cluster containing $\{msoffice, msoffice97, msproject, msexchange\}$. The coupling of other latent states showed similar behavior. Intuitively, large coefficients across a row in Figure 2(b) give rise to a merging of states, while large coefficients across a column lead to splitting of states. Thus, in contrast to pairwise merge, or single split algorithms, CoEM can merge more than two states, and split a single state into multiple states.

4.2 Synthetic data

We synthetically generated both exactly decomposable and noisy empirical joint distributions $\tilde{p}(x, y)$.

