Globally Optimal Model-based Clustering via Mixed Integer Nonlinear Programming

Patrick Flaherty
Department of Mathematics & Statistics
UMass Amherst
Amherst, MA 01002
flaherty@math.umass.edu

Pitchaya Wiratchotisatian
Data Science Program
Worcester Polytechnic Institute
Worcester, MA 01609
pwiratchotisatia@wpi.edu

Andrew C. Trapp
Foise Business School
Worcester Polytechnic Institute
Worcester, MA 01609
atrapp@wpi.edu

Abstract

We present an approach for solving for the optimal partition of a data set via global optimization of a maximum a-posteriori (MAP) estimation problem. Our approach formulates the MAP estimation problem for the Gaussian mixture model as a mixed-integer nonlinear optimization problem (MINLP). Our method provides a certificate of global optimality, can accommodate side constraints, and is extendable to other finite mixture models. We propose an approximation to the MINLP that transforms it into a mixed integer quadratic program (MIQP) which preserves global optimality within desired accuracy and improves computational aspects. Numerical experiments compare our method to standard estimation approaches and show that our method finds the globally optimal MAP for some standard data sets, providing a benchmark for comparing estimation methods.

1 Introduction

In the application of clustering models to real data there is often rich prior information about the relationships among the samples or the relationships between the samples and the parameters. For example, in biological or clinical experiments, it may be known that two samples are technical replicates and should belong to the same cluster, or it may be known that the mean value for certain control samples is in a certain range. However, standard model-based clustering methods make it difficult to enforce such hard logical constraints. Just as important, it is often of interest in similar application domains to have a globally optimal solution because the products of inference will be used to make critical medical treatment decisions. Thus, we are motivated to develop a method for achieving a globally optimal solution for the Gaussian mixture model that allows for the incorporation of rich prior constraints.

We begin by recalling the Gaussian mixture model and specifying the maximum a-posteriori (MAP) estimation problem. The probability density function of a finite mixture model is $p(\boldsymbol{y}|\boldsymbol{\theta}, \boldsymbol{\pi}) = \sum_{k=1}^{K} \pi_k p(\boldsymbol{y}|\boldsymbol{\theta}_k)$ where the observed data is $\boldsymbol{y}$ and the parameter set is $\boldsymbol{\phi} = \{\boldsymbol{\theta}, \boldsymbol{\pi}\}$. The data is an n-tuple of d-dimensional random vectors $\boldsymbol{y} = (\boldsymbol{y}_1^T, \ldots, \boldsymbol{y}_n^T)^T$ and the mixing proportion parameter $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)^T$ is constrained to the probability simplex $\mathcal{P}_K = \left\{ \boldsymbol{p} \in \mathbb{R}^K \;\middle|\; \boldsymbol{p} \succeq 0 \text{ and } \mathbf{1}^T\boldsymbol{p} = 1 \right\}$. When the component density, $p(\boldsymbol{y}|\boldsymbol{\theta}_k)$, is a Gaussian density

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

function, $p(\boldsymbol{y}|\boldsymbol{\phi})$ is a Gaussian mixture model with parameters $\boldsymbol{\theta} = (\{\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1\}, \ldots, \{\boldsymbol{\mu}_K, \boldsymbol{\Sigma}_K\})$. Assuming independent, identically distributed (iid) samples, the Gaussian mixture model probability density function is $p(\boldsymbol{y}|\boldsymbol{\theta}, \boldsymbol{\pi}) = \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k \, p\left(\boldsymbol{y}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)$.

A generative model for the Gaussian mixture density function is

$$\begin{aligned} Z_i &\overset{\text{iid}}{\sim} \text{Categorical}(\boldsymbol{\pi}) \quad \text{for } i = 1, \ldots, n, \\ Y_i|z_i, \boldsymbol{\theta} &\sim \text{Gaussian}(\boldsymbol{\mu}_{z_i}, \boldsymbol{\Sigma}_{z_i}), \end{aligned}$$

where $\boldsymbol{\mu} = (\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K)$ and $\boldsymbol{\Sigma} = (\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_K)$. To generate data from the Gaussian mixture model, first draw $z_i \in \{1, \ldots, K\}$ from a categorical distribution with parameter $\boldsymbol{\pi}$. Then, given z_i, draw $\boldsymbol{y}_i$ from the associated Gaussian component distribution $p(\boldsymbol{y}_i|\boldsymbol{\theta}_{z_i})$.

The posterior distribution function for the generative Gaussian mixture model is

$$p(\boldsymbol{z}, \boldsymbol{\theta}, \boldsymbol{\pi}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{\theta}, \boldsymbol{z}) p(\boldsymbol{z}|\boldsymbol{\pi}) p(\boldsymbol{\theta}, \boldsymbol{\pi})}{p(\boldsymbol{y})}. \tag{1}$$

The posterior distribution requires the specification of a prior distribution $p(\boldsymbol{\theta}, \boldsymbol{\pi})$, and if $p(\boldsymbol{\theta}, \boldsymbol{\pi}) \propto 1$, then MAP estimation is equivalent to maximum likelihood estimation. The MAP estimate for the component membership configuration can be obtained by solving the following optimization problem:

$$\underset{\boldsymbol{z}, \boldsymbol{\theta}, \boldsymbol{\pi}}{\text{maximize}} \; \log p(\boldsymbol{z}, \boldsymbol{\theta}, \boldsymbol{\pi}|\boldsymbol{y}) \quad \text{subject to} \quad z_i \in \{1, \ldots, K\} \; \forall i \text{ and } \boldsymbol{\pi} \in \mathcal{P}_K. \tag{2}$$

Assuming iid sampling, the objective function comprises the following conditional density functions:

$$\begin{aligned} p(\boldsymbol{y}_i|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{z}_i) &= \prod_{k=1}^{K} \left[(2\pi)^{-m/2} |\boldsymbol{\Sigma}_k|^{-1/2} \exp\left(-\frac{1}{2} (\boldsymbol{y}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{y}_i - \boldsymbol{\mu}_k) \right) \right]^{z_{ik}}, \\ p(\boldsymbol{z}_i|\boldsymbol{\pi}) &= \prod_{k=1}^{K} \left[\pi_k \right]^{z_{ik}}, \qquad p(\boldsymbol{\Sigma}, \boldsymbol{\pi}, \boldsymbol{\mu}) \propto 1, \end{aligned}$$

where $z_i \in \{1, \ldots, K\}$ is recast using binary encoding. In the case of one-dimensional data ($d = 1$) and equivariant components ($\boldsymbol{\Sigma}_1 = \cdots = \boldsymbol{\Sigma}_K = \sigma^2$) the MAP optimization problem can be written

$$\begin{aligned} \underset{\boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\pi} \in \Delta_K}{\text{minimize}} \quad & \eta \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} (y_i - \mu_k)^2 - \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \log \pi_k \\ \text{subject to} \quad & \sum_{k=1}^{K} z_{ik} = 1, \qquad i = 1, \ldots, n, \\ & z_{ik} \in \{0, 1\}, \quad i = 1, \ldots, n \text{ and } k = 1, \ldots, K., \\ & -M_k^L \leq \mu_k \leq M_k^U, \qquad k = 1, \ldots, K \end{aligned} \tag{3}$$

where $\eta = \frac{1}{2\sigma^2}$ is the precision, Δ_K is the probability simplex, and M_k^L and M_k^U are real numbers. In a fully Bayesian setting, even if the MAP is of less interest than the full distribution function, the MAP can still be useful as an initial value for a posterior sampling algorithm as suggested by Gelman and Rubin (1996).

Our goal is to solve for the global MAP of this mixed-integer nonlinear program (MINLP) over the true posterior distribution domain while only imposing constraints that are informed by our prior knowledge and making controllable approximations. Importantly, there are no constraints between $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, and $\boldsymbol{z}$ such as $\pi_k = \frac{1}{n} \sum_{i=1}^{n} z_{ik}$ — this is a particular estimator.

Recent work on developing global optimization methods for supervised learning problems has led to impressive improvements in the size of the problems that can be handled[1]. In linear regression, properties such as stability to outliers and robustness to predictor uncertainty have been cast as a mixed-integer quadratic program and solved for samples of size $n \sim 1,000$ (Bertsimas and King, 2016). Best subset selection in the context of regression is typically approximated as a convex problem with the ℓ_1 norm penalty, but can now be solved exactly using the nonconvex ℓ_0 penalty for

[1] see Supplementary Section A for detailed related work

thousands of data points (Bertsimas et al., 2016). Our work is a step in extending this previous work to unsupervised learning problems.

In this work, we achieve approximately two orders of magnitude improvement in computation time over the original MINLP by using a piecewise approximation to the objective entropy term and a constraint-based formulation of the mixed-integer-quadratic objective term, thus converting the problem to a mixed-integer quadratic program (MIQP). Section 2 describes our MIQP formulation and a branch-and-bound algorithm. Section 3 reports the results of comparisons between our methods and standard estimation procedures.

2 Global MAP Estimation with Constraints via Branch-and-Bound

Branch-and-bound algorithms work by iteratively updating a nonincreasing upper bound and a nondecreasing lower bound until the bounds are within some small positive value ϵ. The upper bound is updated by a local feasible solution of Problem 3 and the lower bound is updated by solving a relaxation of Problem 3. The branch-and-bound strategy fixes integer decision variables or bounds continuous variables at each node of the branch-and-bound tree. At a given node of the branch-and-bound search tree, if the lower bound is greater than the best available upper bound (for minimization problems), the subtree is fathomed, that is, excluded from the search space. Branch-and-bound provides for the opportunity to exclude large portions of the search space at each iteration, and if the branching strategy is well-suited to the problem it can drastically reduce the actual computation time for real problems[2]. Our innovations to the standard MINLP branch-and-bound algorithm fall into two categories: changes to the domain and changes to the objective function[3].

2.1 Domain Constraints

Symmetry-breaking Constraint A common difficulty in obtaining a global optimum for Problem 3 is that the optimal value of the objective is invariant to permutations of the component ordering. In the MINLP framework, we eliminate such permutations from the feasible domain by adding a simple linear constraint $\pi_1 \leq \pi_2 \leq \cdots \leq \pi_K$. This constraint reduces the search space thereby improving computational performance.

Specific Estimators Though Problem 3 does not specify any particular form for the estimators of $\boldsymbol{\pi}$ or $\boldsymbol{\mu}$, it may be of interest to specify the estimators with equality constraints. For example, the standard estimators of the EM algorithm are $\pi_k = \frac{1}{n}\sum_{i=1}^{n} z_{ik}$ and $\boldsymbol{\mu}_k = \frac{\sum_{i=1}^{n} \boldsymbol{y}_i z_{ik}}{\sum_{i=1}^{n} z_{ik}}$, for $k = 1, \ldots, K$. The resulting optimization problem can be written entirely in terms of integer variables and the goal is to find the optimal configuration of $\boldsymbol{z}$.

Logical Constraints An important aspect of the MINLP formulation as a Bayesian procedure is the ability to formulate logical constraints that encode rich prior information. These constraints shape the prior distribution domain in a way that is often difficult to express with standard probability distributions. Problem 3 already has one either/or logical constraint specifying that a data point i can belong to only one component k. One can specify that data point i and j have the same component assignment, $z_{ik} = z_{jk}, \ \forall k$, or that they must be assigned to different components, $z_{ik} + z_{jk} \leq 1, \ \forall k$. A non-symmetric constraint can specify that if data point j is assigned to component k, then i must be assigned to k, $z_{jk} \leq z_{ik}$; on the other hand, if i is not assigned to component k, then j cannot be assigned to component k. Additional logical constraints can be formulated in a natural way in the MINLP such as: packing constraints (from a set of data points $\mathcal{I}$, select at most L to be assigned to component k), $\sum_{i \in \mathcal{I}} z_{ik} \leq L$; partitioning constraints (from a set of data points, select exactly one to be assigned to component k) $\sum_{i \in \mathcal{I}} z_{ik} = L$; and covering constraints (from a set of data points select at least L) $\sum_{i \in \mathcal{I}} z_{ik} \geq L$.

[2] See supplementary Section B for computational complexity discussion and Supplementary Section D for the details of the branch-and-bound algorithm

[3] See Supplementary Section C for discussion of the prior in shaping the objective.

2.2 McCormick-Based Relaxation of the Objective Function

Recall the objective function of Problem 3 is

$$f(\boldsymbol{z}, \boldsymbol{\pi}, \boldsymbol{\mu}; \boldsymbol{y}, \eta) = \eta \sum_{k=1}^{K} \sum_{i=1}^{n} z_{ik} (\boldsymbol{y}_i - \mu_k)^2 - \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \log \pi_k.$$

The objective function is nonlinear due to the product of z_{ik} and μ_k^2 in the first term and the product of the $\log$ function and binary z_{ik} in the second term.

The template-matching term in the objective function has two nonlinearities: $2y_i z_{ik} \mu_k$ and $z_{ik} \mu_k^2$. Such polynomial-integer terms are, in fact, commonly encountered in capital budgeting and scheduling problems (Glover, 1975). We have from Problem 3 that $M_k^L \leq \mu_k \leq M_k^U$ when the bounds are not the same. Given $\sum_k z_{ik} = 1$ and z_{ik} is binary, we can rewrite the term $\sum_k z_{ik} (\boldsymbol{y}_i - \boldsymbol{\mu}_k)^2$ as $(\boldsymbol{y}_i - \sum_k z_{ik} \boldsymbol{\mu}_k)^2$. Then, we introduce a new variable $t_{ik} = z_{ik} \mu_k$, and McCormick's relaxation gives the following four constraints for each (i, k):

$$M_k^L z_{ik} \leq t_{ik} \leq M_k^U z_{ik}, \tag{4}$$

$$\mu_k - M_k^U (1 - z_{ik}) \leq t_{ik} \leq \mu_k - M_k^L (1 - z_{ik}). \tag{5}$$

Now, the objective function term $(y_i - \sum_k t_{ik})^2$ is quadratic in the decision variables and the additional constraints are linear in the decision variables.

The cross-entropy term, $z_{ik} \log \pi_k$ is a source of nonlinearity in Problem 3. Approximating this nonlinearity with a piecewise linear function has two benefits. The accuracy of the approximation can be controlled by the number of breakpoints in the approximation. The breakpoint locations can be adaptively set as the optimization iterations progress to gain higher accuracy in the region of the MAP and the approximation can be left coarser elsewhere. Indeed, optimally fitting a piecewise-linear function is itself a hard problem, but good convex optimization methods have been developed to solve it (Magnani and Boyd, 2008), so we employ regular breakpoints in the MIQP implemented in for our experiments in Section 3.

3 Experiments

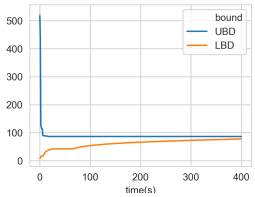
We obtained the Iris (`iris`, $n = 150$), Wine Quality (`wine`, $n = 178$), and Wisconsin Breast Cancer (`brca/wdbc`, $n = 569$) data sets from the UCI Machine Learning Repository (Dua and Graff, 2017). A 1-d projection of `iris` was obtained by projecting on the first principal component (designated `iris1d`), and only the following features were employed for the `brca` data set: worst area, worst smoothness, and mean texture. The `wine` data set is 13-dimensional and the `brca` data set is 3-dimensional. Since our goal is to obtain the global MAP given the data set rather than a prediction, all of the data was used for estimation. The component standard deviation was fixed to 0.4^2 for the `iris1d` data (close to the average standard deviation) and the precision matrices were fixed to the mean across all of the true component precision matrices for `wine` and `brca`[4].

Table 2 in Supplementary Section F shows the comparison of our proposed branch-and-bound methods (MINLP, MIQP) with standard local search methods (EM, SLSQP, SA). Our primary interest lies in achieving a measure of convergence to the global optimum and the relative gap indicates the proximity of the upper and lower bounds. On all of the data sets, all methods find roughly the same optimal value. The MINLP method consistently has a fairly large gap and the MIQP method has a much smaller gap indicating that it provides a faster guarantee of global optimality.

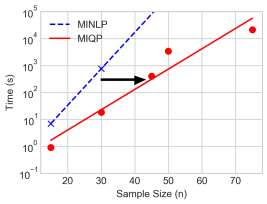
We assessed the convergence to the global optimum for the `iris1d` data set restricted to $n = 45$ data points (15 in each of the three true components). Figure 1a shows that the upper bound converges very quickly to the global optimum, and it takes the majority of the computaton time to (computationally) prove optimality within a predetermined ϵ threshold. Figure 1b shows the computation time vs. sample size for the `iris1d` data set for the MINLP and the MIQP methods. Our MIQP approach reduces computation time by a multiplier that increases with sample size and is around two orders of magnitude for $n = 45$. The MINLP is computationally infeasible for larger sample sizes, while the MIQP is able to provide a certificate of global optimality[5].

[4] See Supplementary Section E for details of the experimental protocol.

[5] See Supplementary Section G for a more complete discussion of these results.



(a) Convergence of upper and lower bounds for MIQP method.



(b) MIQP improves upon the MINLP solution computation time as shown by the shift of curve to the right.

Figure 1: Global convergence and comparison between MINLP and MIQP

References

Tobias Achterberg, Thorsten Koch, and Alexander Martin. Branching rules revisited. *Operations Research Letters*, 33(1):42–54, 2005. doi: 10.1016/j.orl.2004.04.002. URL `https://doi.org/10.1016/j.orl.2004.04.002`.

Christophe Andrieu, Nando de Freitas, Arnaud Doucet, and Michael I. Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 50(1):5–43, Jan 2003. ISSN 1573-0565. doi: 10.1023/A:1020281327116. URL `https://doi.org/10.1023/A:1020281327116`.

Sivaraman Balakrishnan, Martin J. Wainwright, and Bin Yu. Statistical guarantees for the EM algorithm: From population to sample-based analysis. *Ann. Statist.*, 45(1):77–120, 02 2017. doi: 10.1214/16-AOS1435. URL `https://doi.org/10.1214/16-AOS1435`.

Matthew J. Beal and Zoubin Ghahramani. The variational bayesian em algorithm for incomplete data: with application to scoring graphical model structures. In JM Bernardo, MJ Bayarri, JO Berger, AP Dawid, D Heckerman, AFM Smith, and M West, editors, *Bayesian Statistics 7: Proceedings of the Seventh Valencia International Meeting*, volume 7, pages 453–464. Oxford University Press, 2003.

Dimitris Bertsimas and Angela King. OR forum–an algorithmic approach to linear regression. *Operations Research*, 64(1):2–16, 2016. doi: 10.1287/opre.2015.1436. URL `https://doi.org/10.1287/opre.2015.1436`.

Dimitris Bertsimas and Angela King. Logistic regression: From art to science. *Statistical Science*, 32(3):367–384, 2017. doi: 10.1214/16-sts602. URL `https://doi.org/10.1214/16-sts602`.

Dimitris Bertsimas, Angela King, and Rahul Mazumder. Best subset selection via a modern optimization lens. *The Annals of Statistics*, 44(2):813–852, 2016. doi: 10.1214/15-aos1388. URL `https://doi.org/10.1214/15-aos1388`.

Pierre Le Bodic and George L. Nemhauser. How important are branching decisions: Fooling MIP solvers. *Operations Research Letters*, 43(3):273–278, 2015. doi: 10.1016/j.orl.2015.03.003. URL `https://doi.org/10.1016/j.orl.2015.03.003`.

Michael Braun and Jon McAuliffe. Variational inference for large-scale models of discrete choice. *Journal of the American Statistical Association*, 105(489):324–335, 2010. doi: 10.1198/jasa.2009.tm08030. URL `https://doi.org/10.1198/jasa.2009.tm08030`.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Andrew Gelman and Donald B Rubin. Markov chain monte carlo methods in biostatistics. *Statistical Methods in Medical Research*, 5(4):339–355, 1996. doi: 10.1177/096228029600500402. URL `https://doi.org/10.1177/096228029600500402`.

Fred Glover. Improved linear integer programming formulations of nonlinear integer problems. *Management Science*, 22(4):455–460, 1975. doi: 10.1287/mnsc.22.4.455. URL `https://doi.org/10.1287/mnsc.22.4.455`.

LLC Gurobi Optimization. Gurobi optimizer reference manual, 2018. URL `http://www.gurobi.com`.

Chi Jin, Yuchen Zhang, Sivaraman Balakrishnan, Martin J. Wainwright, and Michael I. Jordan. Local maxima in the likelihood of gaussian mixture models: Structural results and algorithmic consequences. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 4123–4131, USA, 2016. Curran Associates Inc. ISBN 978-1-5108-3881-9. URL http://dl.acm.org/citation.cfm?id=3157382.3157558.

Jan Kronqvist, David E. Bernal, Andreas Lundell, and Ignacio E. Grossmann. A review and comparison of solvers for convex MINLP. *Optimization and Engineering*, 20(2):397–455, Jun 2019. ISSN 1573-2924. doi: 10.1007/s11081-018-9411-8. URL https://doi.org/10.1007/s11081-018-9411-8.

A. H. Land and A. G. Doig. An automatic method of solving discrete programming problems. *Econometrica*, 28(3):497, 1960. doi: 10.2307/1910129. URL https://doi.org/10.2307/1910129.

Kenneth Lange, David R. Hunter, and Ilsoon Yang. Optimization transfer using surrogate objective functions. *Journal of Computational and Graphical Statistics*, 9(1):1, 2000. doi: 10.2307/1390605. URL https://doi.org/10.2307/1390605.

Alessandro Magnani and Stephen P. Boyd. Convex piecewise-linear fitting. *Optimization and Engineering*, 10(1):1–17, 2008. doi: 10.1007/s11081-008-9045-3. URL https://doi.org/10.1007/s11081-008-9045-3.

Julien Mairal. Optimization with first-order surrogate functions. In *Proceedings of the 30th International Conference on Machine Learning*, 2013.

Katta G. Murty and Santosh N. Kabadi. Some NP-complete problems in quadratic and nonlinear programming. *Mathematical Programming*, 39(2):117–129, 1987. doi: 10.1007/bf02592948. URL https://doi.org/10.1007/bf02592948.

George Nemhauser and Laurence Wolsey. *Integer and Combinatorial Optimization*. []. John Wiley & Sons, Inc., 1988. doi: 10.1002/9781118627372. URL https://doi.org/10.1002/9781118627372.

Jorge Nocedal and Stephen J. Wright. *Numerical optimization*. Springer, New York, 2006. ISBN 9780387303031.

N. V. Sahinidis. *BARON 17.8.9: Global Optimization of Mixed-Integer Nonlinear Programs,* User's Manual, 2017.

Douglas R. Smith. *On the Computational Complexity of Branch and Bound Search Strategies*. PhD thesis, Naval Postgraduate School, 1979.

Mohit Tawarmalani and Nikolaos V. Sahinidis. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*, 103(2):225–249, 2005. doi: 10.1007/s10107-005-0581-8. URL https://doi.org/10.1007/s10107-005-0581-8.

Martin J. Wainwright and Michael I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305, 2007. doi: 10.1561/2200000001. URL https://doi.org/10.1561/2200000001.

David J. Wales and Jonathan P. K. Doye. Global optimization by basin-hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 atoms. *The Journal of Physical Chemistry A*, 101(28):5111–5116, 1997. doi: 10.1021/jp970984n. URL https://doi.org/10.1021/jp970984n.

Chong Wang and David M. Blei. Variational inference in nonconjugate models. *J. Mach. Learn. Res.*, 14(1):1005–1031, April 2013. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=2502581.2502613.

Supplementary Information

A Related Work

Many MAP methods can be interpreted as a specific combination of a relaxation of Problem 3 and a search algorithm for finding a local or global minimum. Table 1 summarizes these relationships.

Method	Domain Relaxation	Objective Approximation	Search Algorithm
EM Algorithm	✓	–	coordinate/stochastic descent
Variational EM	✓	✓	coordinate/stochastic descent
SLSQP	✓	✓	coordinate descent
Simulated Annealing	✓	–	stochastic descent

Table 1: Summary of approximation methods for MAP estimation in an optimization framework.

EM Algorithm The EM algorithm relaxes the domain such that $z_{ik} \in [0, 1]$ instead of $z_{ik} \in \{0, 1\}$. The decision variables of the resulting biconvex optimization problem are partitioned into two groups: $\{\boldsymbol{z}\}$ and $\{\boldsymbol{\mu}, \boldsymbol{\pi}\}$. The search algorithm performs coordinate ascent on these two groups. There are no guarantees for the global optimality of the estimate produced by the EM algorithm. While Balakrishnan et al. (2017) showed that the global optima of a mixture of well-separated Gaussians has a relatively large region of attraction, Jin et al. (2016) showed that inferior local optima can be arbitrarily worse than the global optimum.[6]

Variational EM The variational EM algorithm introduces a surrogate function $q(\boldsymbol{z}, \phi|\boldsymbol{\xi})$ for the posterior distribution $p(\boldsymbol{z}, \boldsymbol{\phi}|\boldsymbol{y})$ (Beal and Ghahramani, 2003). First, the surrogate is fit to the posterior by solving $\hat{\boldsymbol{\xi}} \in \arg\min_{\boldsymbol{\xi}} \text{KL}(q(\phi, \boldsymbol{z}|\xi) \,||\, p(\phi, \boldsymbol{z}|\boldsymbol{y}))$. Then the surrogate is used in place of the posterior distribution in the original optimization problem $\hat{\phi}, \hat{\boldsymbol{z}} \in \arg\min_{\phi, \boldsymbol{z}} \log q(\theta, \boldsymbol{z}|\boldsymbol{\xi})$. The search algorithm performs coordinate ascent on $\{\phi, \boldsymbol{z}\}$ and $\boldsymbol{\xi}$. The computational complexity is improved over the original MAP problem by selecting a surrogate that has favorable structure (linear or convex) and by relaxing the domain of the optimization problem. This surrogate function approach has been discovered independently in many fields; it is alternatively known as majorization-minimization (Lange et al., 2000) and has deep connections with Franke-Wolfe gradient methods and block coordinate descent methods (Mairal, 2013). The domain of the problem can be viewed as a marginal polytope and outer approximations of the marginal polytope lead to efficient sequential approximation methods that have satisfying theoretical properties (Wainwright and Jordan, 2007).

SLSQP Sequential Least-Squares Quadratic Programming (SLSQP) is a popular general-purpose constrained nonlinear problem method that uses a quadratic surrogate function to approximate the Lagrangian (Nocedal and Wright, 2006). In SLSQP, the surrogate function is construed as a quadratic approximation of the Lagrangian of the original problem. The domain of the original problem is also relaxed so that the constraint cuts are approximated by linear functions. Like variational EM, SLSQP iterates between fitting the surrogate function and optimizing over the decision variables. Quadratic surrogate functions have also been investigated in the context of variational EM for nonconjugate models (Wang and Blei, 2013; Braun and McAuliffe, 2010).

Simulated Annealing Simulated annealing methods are theoretically guaranteed to converge to a global optimum of a nonlinear objective. However, choosing the annealing schedule for a particular problem is challenging and the guarantee of global optimality only exists in the limit of the number of steps; there is no general way to choose the annealing schedule or monitor convergence (Andrieu et al., 2003). Furthermore, designing a sampler for the binary $\boldsymbol{z}$ can be challenging unless the domain is relaxed. Even so, modern simulated annealing-type methods such a basin hopping have shown promise in practical applications (Wales and Doye, 1997).

Branch-and-Bound In many practical MINLPs, it is critical to obtain the global optimum with a certificate of optimality, that is, with a (computational) proof that no better solution exists. For these situations, branch-and-bound methods, first proposed by Land and Doig (1960), have seen the most

[6] Figure 1 of Jin et al. (2016) illustrates the complexity of the likelihood surface for the Gaussian mixture model (GMM).

success. While MINLP problems remain NP-hard in general, the scale of problems that can be solved to global optimality has increased dramatically in the past 20 years (Bertsimas and King, 2017). The current state-of-the-art solver for general MINLPs is the Branch-and-Reduce Optimization Navigator (BARON). BARON exploits general problem structure to branch-and-bound on both discrete and continuous decision variables (Sahinidis, 2017).

B Computational Complexity

Problems in the MINLP class are NP-hard in general and present two primary computational challenges (Murty and Kabadi, 1987). First, as the number of data points increases, the size of the configuration space of $\boldsymbol{z}$ increases exponentially (Nemhauser and Wolsey, 1988). Second, the nonlinear objective function can have many local minima. Despite these worst-case complexity results, MINLP problems are increasingly often solved in practice. Some work on the assymetric traveling salesman problem — another NP-hard MINLP — provided evidence that it can be solved in time $O(n^3 \log^2(n))$ on average (Smith, 1979). Good empirical performance is often due to exploiting problem-specific structures and powerful algorithmic improvements such as branch-and-bound, branch-and-cut, and branch-and reduce algorithms (Bodic and Nemhauser, 2015; Tawarmalani and Sahinidis, 2005).

C Objective Function Shaping Strategies

Prior Distribution The objective function can be shaped by the prior distribution, $p(\boldsymbol{\mu}, \boldsymbol{\pi})$. In the formulation of Problem 3 a uniform prior was selected such that $p(\boldsymbol{\mu}, \boldsymbol{\pi}) \propto 1$ and the prior does not affect the optimizer. But, an informative prior such as a multivariate Gaussian $\boldsymbol{\mu} \sim \text{MVN}(\mathbf{0}, \boldsymbol{S})$ could be used to regularize $\boldsymbol{\mu}$, or a non-informative prior such as Jeffrey's prior could be used in an objective Bayesian framework.

D Algorithm Strategies

Branch-and-Bound for MAP The general branch-and-bound algorithm (Algorithm 1) is implemented as a tree with nodes indexed in a priority queue. A node in the candidate priority queue contains a relaxed subproblem and a lower bound for the subproblem.

Algorithm 1: Branch-and-Bound

Initialize the candidate priority queue to consist of the relaxed MINLP and set UBD $= \infty$ and GLBD $= -\infty$
while *candidate queue is not empty* **do**
  Pop the first node off the priority queue, solve the subproblem, and store the optimal value f^*.
  if *integer solution for* $\boldsymbol{z}$ **then**
    if $f^* \leq$ *UBD* **then**
      Update UBD $\leftarrow f^*$
      Remove node j from candidate queue where LBD$_j >$ UBD $\forall j$
    end
  else
    Select a branching variable z_{ik} based on branching strategy
    Push node for candidate relaxed subproblem j on queue adding constraint $z_{ik} = 0$ and set LBD$_j = f^*$
    Push node for candidate relaxed subproblem j' on queue adding constraint $z_{ik} = 1$ and set LBD$_{j'} = f^*$
  end
  Set GLBD $= \min_j$ LBD$_j$ for all j in candidate queue
end

Branching Strategies A critical step in the branch-and-bound algorithm is the selection of the variable to branch on in the tree. If the branching strategy leads to a child node subproblem with a lower bound that is greater than the best current upper bound, then the subtree is fathomed and the search space is reduced by a factor of up to one half, drastically improving the computational efficiency. Though the choice of branching strategy should be tailored to the problem, there are three popular general strategies (Achterberg et al., 2005): *Most Infeasible Branching* – choose the

integer variable with the fraction part closest to 0.5 in the relaxed subproblem solution, *Pseudo Cost Branching* — choose the variable that provided the greatest reduction in the objective when it was previously branched on in the tree, and *Strong Branching* — test a subset of the variables at both of their bounds and branch on the one that provides the greatest reduction in the objective function.

Most Infeasible Branching Choose the integer variable with the fraction part closest to 0.5 in the relaxed subproblem solution.

Pseudo Cost Branching Choose the variable that provided the greatest reduction in the objective when it was previously branched on in the tree.

Strong Branching Test a subset of the variables at both of their bounds and branch on the one that provides the greatest reduction in the objective function.

The strong branching strategy can be computationally expensive because two optimization problems must be solved for each candidate, so loose approximations such as linear programs are often used. The pseudo cost strategy does not provide much benefit early in the iterations because there is not much branching history to draw upon.

Most Integral Branching We implemented a *most integral* branching strategy for the GMM MAP problem. The idea is to first find a solution to a relaxed problem where $z_{ik} \in [0, 1]$. Then, identify those z_{ik} variables that are closest to 1 — that is, variables that are most definitively assigned to one component.[7] Those z_{ik} variables are then chosen for branching with the expectation that the relaxed subproblem branch with the constraint that $z_{ik} = 0$ will have a lower bound that is greater than the best upper bound and the subtree can be fathomed. We evaluate the performance of this branching strategy in comparison to the strategies implemented in state-of-the-art commercial solvers in Section 3.

E Experimental Protocol

The EM, SLSQP, and simulated annealing algorithms were initialized using the k-means algorithm; no initialized was provided to the MINLP and MIQP methods. The EM, SLSQP, and SA experiments were run using algorithms defined in `python` and all variants of our approach were implemented in GAMS, a standard optimization problem modeling language. The estimates provided by EM, SLSQP, and SA are not guaranteed to have $z_{ik} \in \{0, 1\}$ so we rounded to the nearest integral value, but we note that solutions from these methods are not guaranteed to be feasible for the MAP problem. MINLP problems were solved using BARON (Tawarmalani and Sahinidis, 2005; Sahinidis, 2017) and mixed-integer quadratic constrained program (MIQCP) problems were solved using Gurobi (Gurobi Optimization, 2018); both are state-of-the-art general-purpose solvers for their respective problems (Kronqvist et al., 2019). We report the upper bound (best found) and lower bound (best possible) of the negative MAP value if the method provides it. For the timing results in Figure 1b we ran both methods on a single core, and for the results in Table 2 we ran all methods on a single core except the MIQP method which allowed multithreading, so we used 16 cores for that method. The computation time for all methods was limited to 12 hours. The metrics for estimating $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, and $\boldsymbol{z}$ are shown in results Table 2.

F Comparison to Local Search Methods

In this section we compare our MINLP and MIQP formulations to local search methods.

G Discussion

We formulated the MAP estimation problem as a MINLP and cast several local MAP estimation methods in the framework of the MINLP. We identified three aspects of the MINLP that can be adjusted to incorporate prior information and improve computational efficiency. We suggested an approximation that converts the MINLP to a MIQP that offers control over the approximation error at the expense of computational time. We showed that the MIQP solution pushes the frontier of sample size that can be handled which still providing a way to encode prior information in the form of hard constraints.

[7] Recall that $z_{ik} = 0$ only constrains data point i to be not assigned to component k, but which of the other $K - 1$ components it is assigned to is not fixed.

		Local			Global (BnB)	
Data Set	Metric	EM	SLSQP	SA	MINLP	MIQP
iris-1d	$-\log$ MAP	280.60	287.44	283.28	280.02	282.71
	LBD	—	—	—	9.27	161.60
	$\sup \lvert\hat{\boldsymbol{\pi}} - \boldsymbol{\pi}\rvert$	0.075	0.013	0.000	0.093	0.165
	$\lVert\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\rVert_2$	0.278	0.065	0.277	0.356	0.356
	$1/n \sum_i \sup \lvert\hat{\boldsymbol{z}}_i - \boldsymbol{z}_i\rvert$	0.067	0.067	0.087	0.093	0.093
wine	$-\log$ MAP	1367.00	1368.71	1368.71	1366.85	1390.13
	LBD	—	—	—	-2.2×10^5	183.42
	$\sup \lvert\hat{\boldsymbol{\pi}} - \boldsymbol{\pi}\rvert$	0.005	0.066	0.066	0.006	0.167
	$\lVert\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\rVert_2$	2.348	1.602	1.652	1.618	14.071
	$1/n \sum_i \sup \lvert\hat{\boldsymbol{z}}_i - \boldsymbol{z}_i\rvert$	0.006	0.006	0.006	0.006	0.022
brca	$-\log$ MAP	1566.49	1662.97	1662.97	1566.40	1578.49
	LBD	—	—	—	-2.7×10^4	332.30
	$\sup \lvert\hat{\boldsymbol{\pi}} - \boldsymbol{\pi}\rvert$	0.167	0.127	0.127	0.169	0.122
	$\lVert\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\rVert_2$	394.07	321.11	320.60	401.47	418.05
	$1/n \sum_i \sup \lvert\hat{\boldsymbol{z}}_i - \boldsymbol{z}_i\rvert$	0.169	0.139	0.139	0.169	0.174

Table 2: Comparison of expectation-maximization (EM), sequential least squares programming (SLSQP), basin-hopping simulated annealing (SA), branch-and-bound on the MINLP using BARON (MINLP), branch-and-bound on the MIQP using Gurobi (MIQP). The solution reported for MINLP and MIQP are the best feasible solution found. The total variation distance metric is used for $\boldsymbol{\pi}$, the L_2 distance is used for $\boldsymbol{\mu}$, and the average (across samples) total variation distance is used for $\boldsymbol{z}$.

Our numerical experiments show that our approach can reasonably handle data sets in the range of $n \sim 50$. For larger data sets, our method still provides upper/lower bounds on the globally optimal solution, but these bounds are looser than for the smaller data sets for a fixed computation budget. While $n \sim 50$ may seems like an unrealistically small data set size, in fact, almost all phase I clinical trials have $n \sim 20$ and many DNA sequencing experiments in biology use data sets with $n \sim 50$ samples. In these applications, the ability to encode prior information about the relationships between samples is particularly important.

Our results suggest several areas of further improvements in computational efficiency. We explored the most-integral branching strategy, but found the computational time performance was not as good as the commercially available branching heuristics in the state-of-the-art solvers. Reformulations of the optimization problem are often at the heart of computational improvements in practice and we expect that such reformulations will improve computational efficiency paralleling the improvements in supervised learning (Bertsimas et al., 2016).