SCIENCE CHINA
Information Sciences

• LETTER •

March 2023, Vol. 66 139103:1–139103:2
https://doi.org/10.1007/s11432-021-3477-3

A variational hardcut EM algorithm for the mixtures of Gaussian processes

Tao LI & Jinwen MA*

Department of Information and Computational Sciences, School of Mathematical Sciences and LMAM, Peking University, Beijing 100871, *China*

Received 16 June 2021/Revised 9 December 2021/Accepted 26 J

$\boldsymbol{\Lambda} = \{\boldsymbol{\lambda}_i\}_i^N$, $\boldsymbol{\lambda}_i = \{\lambda_{i,k}\}_k^K$ and $q(z_i;\boldsymbol{\lambda}_i)$ is the probability mass function of a categorical distribution, i.e., $q(z_i = k) = \lambda_{i,k}$. According to the mean-field variational inference theory [5], the optimal $q(\boldsymbol{z};\boldsymbol{\Lambda})$ satisfies

$$\lambda_{i,k} \propto \exp(\mathbb{E}_{q\ \boldsymbol{z}_{-i}\ \boldsymbol{\Lambda}}\,[\log p(\boldsymbol{X},\boldsymbol{y},\boldsymbol{z}_{-i}\cup\{z_i=k\};\boldsymbol{\Theta})]), \quad (2)$$

where $\boldsymbol{z}_{-i} = \boldsymbol{z} - \{z_i\}$. Therefore, we can perform the fixed point iteration based on (2) to find the optimal $q(\boldsymbol{z};\boldsymbol{\Lambda})$.

Although the components of $\boldsymbol{z}_{-i}$ are independent, the complete log-likelihood is not separable with respect to $\boldsymbol{z}_{-i}$, thus the expectation inside (2) is intractable. We apply the hardcut approximation to side-step this problem. For $j \neq i$, we approximate $q(z_j;\boldsymbol{\lambda}_j)$ via a deterministic allocation $\tilde{q}(z_j;\boldsymbol{\lambda}_j) = \mathbb{I}(z_j = \arg\max_{k\ \ ,\ldots,K} \lambda_{j,k})$. According to $\tilde{q}(\boldsymbol{z}_{-i};\boldsymbol{\Lambda})$, the latent variables $\boldsymbol{z}_{-i}$ are deterministic, and we write them as $\tilde{\boldsymbol{z}}_{-i}$. We can approximate the intractable expectation in (2) by $\log p(\boldsymbol{X},\boldsymbol{y},\tilde{\boldsymbol{z}}_{-i}\cup\{z_i=k\};\boldsymbol{\Theta})$. Some calculation reveals Eq. (2) can be directly approximated by

$$\begin{aligned}\boldsymbol{\lambda}_{i,k} \propto&\, \pi_k p(\boldsymbol{x}_i;\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)\\ &\frac{p(\boldsymbol{y}_k(\tilde{\boldsymbol{z}}_{-i}\cup\{z_i=k\})|\boldsymbol{X}_k(\tilde{\boldsymbol{z}}_{-i}\cup\{z_i=k\});\boldsymbol{\theta}_k)}{p(\boldsymbol{y}_{-i,k}(\tilde{\boldsymbol{z}}_{-i})|\boldsymbol{X}_{-i,k}(\tilde{\boldsymbol{z}}_{-i});\boldsymbol{\theta}_k)}, \quad (3)\end{aligned}$$

where

$$\begin{aligned}\boldsymbol{X}_{-i,k}(\boldsymbol{z}) &= \{\boldsymbol{x}_j | z_j = k, j = 1,\ldots,N \text{ and } j\neq i\},\\ \boldsymbol{y}_{-i,k}(\boldsymbol{z}) &= \{y_j | z_j = k, k = 1,\ldots,N \text{ and } j\neq i\}.\end{aligned}$$

Finally, we find out that although $\boldsymbol{\lambda}_i$ is important in deriving the algorithm, but it is not essential for implementation purpose. Instead, we can perform iterations on $\tilde{\boldsymbol{z}}$ directly,

$$\begin{aligned}\tilde{z}_i = &\arg\max_{k\ \ ,\ldots,K} \pi_k p(\boldsymbol{x}_i;\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)\\ &\frac{p(\boldsymbol{y}_k(\tilde{\boldsymbol{z}}_{-i}\cup\{\tilde{z}_i=k\})|\boldsymbol{X}_k(\tilde{\boldsymbol{z}}_{-i}\cup\{\tilde{z}_i=k\});\boldsymbol{\theta}_k)}{p(\boldsymbol{y}_{-i,k}(\tilde{\boldsymbol{z}}_{-i})|\boldsymbol{X}_{-i,k}(\tilde{\boldsymbol{z}}_{-i});\boldsymbol{\theta}_k)}. \quad (4)\end{aligned}$$

Once the above variational E-step iteration converges, we obtain an approximate posterior $\tilde{q}(\boldsymbol{z}) = \mathbb{I}(\boldsymbol{z} = \tilde{\boldsymbol{z}})$ and we can calculate the approximate Q-function $\tilde{\mathcal{Q}}(\boldsymbol{\Theta};\boldsymbol{\Theta}^{\mathrm{o\ d}}) = \mathbb{E}_{q\ \boldsymbol{z}}\,[\mathcal{L}(\boldsymbol{\Theta},\boldsymbol{z})] = \mathcal{L}(\boldsymbol{\Theta},\tilde{\boldsymbol{z}})$. Then we maximize $\tilde{\mathcal{Q}}(\boldsymbol{\Theta};\boldsymbol{\Theta}^{\mathrm{o\ d}})$ with respect to $\boldsymbol{\Theta}$ to estimate the parameters. The entire algorithm is summarized in Algorithm A1 (see Appendix A).

Comparisons with the other algorithms on the learning of MGPs. The iteration formula (4) is very similar to the Gibbs sampling step in the MCMC-EM algorithm. The only difference is that the MCMC-EM algorithm samples z_i according to the probability, while the VHEM algorithm assigns z_i to be the class label with the highest probability.

In the hardcut EM algorithm, dependences among samples in the same Gaussian process component are ignored, and we only use $p(y_i|\boldsymbol{x}_i;\boldsymbol{\theta}_k)$ to measure the probability that the i-th sample belonging to the k-th Gaussian process component. In this way, we do not need to perform iterations since $\{z_i\}_i^N$ are independent. From (4), we can see that in the VHEM algorithm, when we calculate the probability that the i-th sample coming from the k-th component, other samples temporarily assigned with label k are also taken into consideration. Let

$$\begin{aligned}\boldsymbol{c} &= c(\boldsymbol{X}_{-i,k}(\tilde{\boldsymbol{z}}_{-i}),\boldsymbol{x}_i;\boldsymbol{\theta}_k),\\ \boldsymbol{C}_{-} &= c(\boldsymbol{X}_{-i,k}(\tilde{\boldsymbol{z}}_{-i}),\boldsymbol{X}_{-i,k}(\tilde{\boldsymbol{z}}_{-i});\boldsymbol{\theta}_k),\end{aligned}$$

then the last term of (4) is given by

$$\begin{aligned}\log &\frac{p(\boldsymbol{y}_k(\tilde{\boldsymbol{z}}_{-i}\cup\{\tilde{z}_i=k\})|\boldsymbol{X}_k(\tilde{\boldsymbol{z}}_{-i}\cup\{\tilde{z}_i=k\});\boldsymbol{\theta}_k)}{p(\boldsymbol{y}_{-i,k}(\tilde{\boldsymbol{z}}_{-i})|\boldsymbol{X}_{-i,k}(\tilde{\boldsymbol{z}}_{-i});\boldsymbol{\theta}_k)}\\ &= -\frac{1}{2}\log(2\pi) - \frac{1}{2}\log(\theta_{k,}\ \ + \sigma_k - \boldsymbol{c}^{\mathrm{T}}\boldsymbol{C}_{-}^{-}\ \boldsymbol{c})\\ &\quad - \frac{(\boldsymbol{y}_{-i,k}(\tilde{\boldsymbol{z}}_{-i})^{\mathrm{T}}\boldsymbol{C}_{-}^{-}\ \boldsymbol{c} - y_i)}{2(\theta_{k,}\ \ + \sigma_k - \boldsymbol{c}^{\mathrm{T}}\boldsymbol{C}_{-}^{-}\ \boldsymbol{c})}. \quad (5)\end{aligned}$$

The hardcut EM algorithm can be regarded as a further approximation of the VHEM algorithm when $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{C}_{-}^{-}$ is replaced by $\boldsymbol{0}$. This is not realizable because $\boldsymbol{C}_{-}^{-}$ is always invertible, and $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{C}_{-}^{-} = \boldsymbol{0}$ implies $\boldsymbol{c} = \boldsymbol{0}$, which cannot hold for general covariance functions. However, when $\boldsymbol{x}_i$ is far from the points in $\boldsymbol{X}_{-}$, we may expect $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{C}_{-}^{-} \approx \boldsymbol{0}$. For the case $\boldsymbol{x}_i$ lies in the high-probability region of the l-th component, the difference between the VHEM algorithm and the hardcut EM algorithm for computing $\boldsymbol{\lambda}_{i,k}, k\neq l$ is negligible. If $\boldsymbol{x}_i$ lies in the overlapping region of the k-th component and the l-th component, the difference would be significant.

Furthermore, Eq. (5) also presents an intuitive interpretation on the VHEM algorithm from the perspective of leave-one-out cross validation [6]. See Appendix B for more detailed discussion.

Experiments. See Appendix C.

Conclusion. We have proposed a new kind of variational inference based learning algorithm (VHEM) for the generative MGP model. The main advantages of the VHEM algorithm are three folds. First, compared with the hardcut EM algorithm, its derivation relies on variational inference; thus its theory is more solid and sound. Second, it connects existing learning algorithms for MGP, including the hardcut EM algorithm, the MCMC-EM algorithm, and the LOOCV algorithm. Last, it is remarkably faster than the MCMC-EM algorithm and significantly more accurate than the hardcut EM algorithm. The VHEM algorithm is able to achieve comparable performances with the MCMC-EM algorithm, with the cost of a little longer running times compared with the hardcut EM algorithm. To balance performance and computational cost, the VHEM algorithm is a good choice for real applications.

Acknowledgements This work was supported by National Key Research and Development Program of China (Grant No. 2018AAA0100205).

Supporting information Appendixes A–C. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Tresp V. Mixtures of Gaussian processes. In: Proceedings of Advances in Neural Information Processing Systems, 2001. 654–660
2 Chen Z Y, Ma J W, Zhou Y T. A precise hard-cut EM algorithm for mixtures of Gaussian processes. In: Proceedings of International Conference on Intelligent Computing. Berlin: Springer, 2014. 68–75
3 Wu D, Ma J W. An effective EM algorithm for mixtures of Gaussian processes via the MCMC sampling and approximation. Neurocomputing, 2019, 331: 366–374
4 Williams C K, Rasmussen C E. Gaussian Processes for Machine Learning. Cambridge: MIT Press, 2006
5 Bishop C M. Pattern Recognition and Machine Learning. Berlin: Springer, 2006
6 Yang Y, Ma J W. An efficient EM approach to parameter learning of the mixture of Gaussian processes. In: Proceedings of International Symposium on Neural Networks. Berlin: Springer, 2011. 165–174