




gmr: Gaussian Mixture Regression

Alexander Fabisch¹

¹ Robotics Innovation Center, DFKI GmbH

DOI: [10.21105/joss.03054](https://doi.org/10.21105/joss.03054)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Mikkel Meyer Andersen](#)



Reviewers:

- [@inakleinbottle](#)
- [@soodoku](#)

Submitted: 30 January 2021

Published: 02 June 2021

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

gmr is a Python library for Gaussian mixture regression (GMR). GMR is a regression approach that models probability distributions rather than functions. Hence, it is possible to model multimodal mappings.

In GMR we first learn a joint probability distribution $p(\mathbf{x}, \mathbf{y})$ of input \mathbf{x} and output \mathbf{y} through expectation maximization ([Dempster et al., 1977](#)) and then compute the conditional distribution $p(\mathbf{y}|\mathbf{x})$ to make predictions. Thus, training is the same procedure as in a standard Gaussian mixture model (GMM).

Statement of Need

The library gmr is fully compatible with scikit-learn ([Pedregosa et al., 2011](#)). It has its own implementation of expectation maximization (EM), but it can also be initialized with a GMM from sklearn, which means that we can also initialize it from a Bayesian GMM of scikit-learn. The prediction process for regression is not available in scikit-learn and, thus, will be provided by gmr.

Note that while sklearn has the function `GaussianMixture.predict`, it does not perform regression. This function computes the index of the Gaussian for which the inputs have the highest probability. Furthermore, multi-modal regression often requires a more complicated interface to extract not just the mean but also individual Gaussians or sample from the predicted distribution.

The library gmr provides a simple interface and several useful features to deal with multimodal regression, mixtures of Gaussians, and multivariate Gaussian distributions:

- EM implementation that only requires numpy and scipy
- computation of conditional distributions
- sampling from confidence regions of multivariate Gaussians
- collapsing a GMM to a single Gaussian
- extraction of individual Gaussians from a (conditional) GMM
- plotting of covariance ellipses
- unscented transform ([Uhlmann, 1995](#)) to estimate the effect of a nonlinear function on a Gaussian distribution

Multimodal regression and Gaussian mixture regression has been used mostly by the robotics community. For example, inverse problems such as inverse kinematics cannot be easily modeled with standard regression approaches. Furthermore, Gaussian mixture regression is the basis of many programming by demonstration approaches ([Billard et al., 2008](#)).

Background

Gaussian mixture regression via EM has been proposed first by [Ghahramani & Jordan \(1994\)](#). [Calinon et al. \(2007\)](#) introduced the term Gaussian mixture regression in the context of imitation learning for trajectories of robots and many publications that use GMR in this domain followed. [Stulp & Sigaud \(2015\)](#) present Gaussian mixture regression in a more recent survey.

Training

During the training phase we learn a Gaussian mixture model

$$p(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}, \mathbf{y} | \boldsymbol{\mu}_{\mathbf{x}\mathbf{y}_k}, \boldsymbol{\Sigma}_{\mathbf{x}\mathbf{y}_k})$$

through EM, where $\mathcal{N}(\mathbf{x}, \mathbf{y} | \boldsymbol{\mu}_{\mathbf{x}\mathbf{y}_k}, \boldsymbol{\Sigma}_{\mathbf{x}\mathbf{y}_k})$ are Gaussian distributions with mean $\boldsymbol{\mu}_{\mathbf{x}\mathbf{y}_k}$ and covariance $\boldsymbol{\Sigma}_{\mathbf{x}\mathbf{y}_k}$, K is the number of Gaussians, and $\pi_k \in [0, 1]$ are priors that sum up to one.

Prediction

Gaussian mixture regression can be used to predict distributions of variables \mathbf{y} by computing the conditional distribution $p(\mathbf{y} | \mathbf{x})$. The conditional distribution of each individual Gaussian

$$\mathcal{N}(\mathbf{x}, \mathbf{y} | \boldsymbol{\mu}_{\mathbf{x}\mathbf{y}}, \boldsymbol{\Sigma}_{\mathbf{x}\mathbf{y}})$$

$$\boldsymbol{\mu}_{\mathbf{x}\mathbf{y}} = \begin{pmatrix} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_y \end{pmatrix}, \quad \boldsymbol{\Sigma}_{\mathbf{x}\mathbf{y}} = \begin{pmatrix} \boldsymbol{\Sigma}_{xx} & \boldsymbol{\Sigma}_{xy} \\ \boldsymbol{\Sigma}_{yx} & \boldsymbol{\Sigma}_{yy} \end{pmatrix}$$

is defined by

$$\boldsymbol{\mu}_{\mathbf{y} | \mathbf{x}} = \boldsymbol{\mu}_y + \boldsymbol{\Sigma}_{yx} \boldsymbol{\Sigma}_{xx}^{-1} (\mathbf{x} - \boldsymbol{\mu}_x)$$

$$\boldsymbol{\Sigma}_{\mathbf{y} | \mathbf{x}} = \boldsymbol{\Sigma}_{yy} - \boldsymbol{\Sigma}_{yx} \boldsymbol{\Sigma}_{xx}^{-1} \boldsymbol{\Sigma}_{xy}.$$

In a Gaussian mixture model we compute the conditional distribution of each individual Gaussian and their priors

$$\pi_{\mathbf{y} | \mathbf{x}_k} = \frac{\mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_{\mathbf{x}_k}, \boldsymbol{\Sigma}_{\mathbf{x}_k})}{\sum_{l=1}^K \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_{\mathbf{x}_l}, \boldsymbol{\Sigma}_{\mathbf{x}_l})}$$

to obtain the conditional distribution

$$p(\mathbf{y} | \mathbf{x}) = \sum_{k=1}^K \pi_{\mathbf{y} | \mathbf{x}_k} \mathcal{N}(\mathbf{y} | \boldsymbol{\mu}_{\mathbf{y} | \mathbf{x}_k}, \boldsymbol{\Sigma}_{\mathbf{y} | \mathbf{x}_k}).$$

Examples

Here is an example of a dataset where multiple outputs \mathbf{y} are valid predictions for one input \mathbf{x} . It was introduced by [Bishop \(1994\)](#).

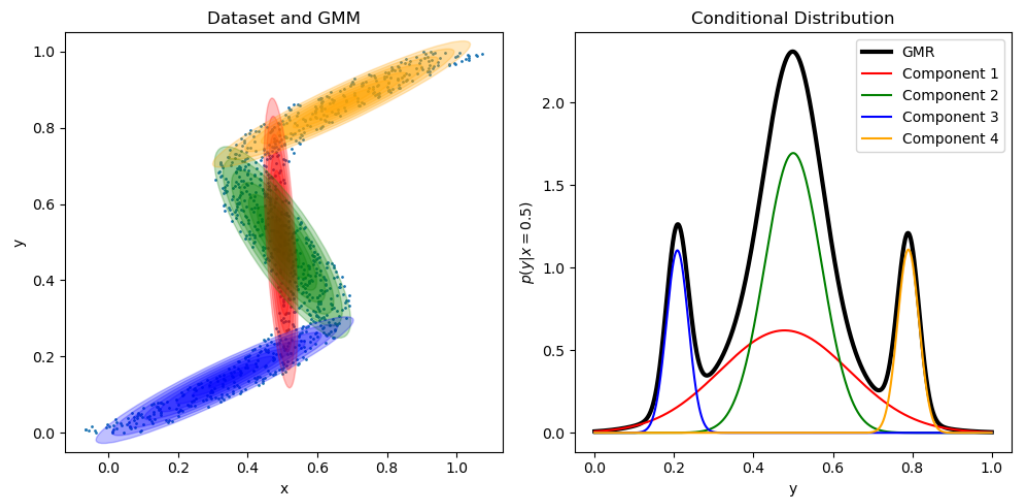


Figure 1: Multimodal regression.

On the left side of [Figure 1](#) we see the training data and the fitted GMM indicated by ellipses corresponding to its components. On the right side we see the predicted probability density $p(y|x = 0.5)$. There are three peaks that correspond to three different valid predictions. Each peak is represented by at least one of the Gaussians of the GMM.

We can use GMR to represent demonstrated motions. Here is an example in 2D, in which we have a dataset that is a sequence of positions x and corresponding velocities \dot{x} . We train a GMM to represent $p(x, \dot{x})$. Then we can generate a new trajectory by iteratively sampling $\dot{x}_t \sim p(\dot{x}|x = x_t)$ and computing the next position as $x_{t+1} = x_t + \Delta t \dot{x}_t$.

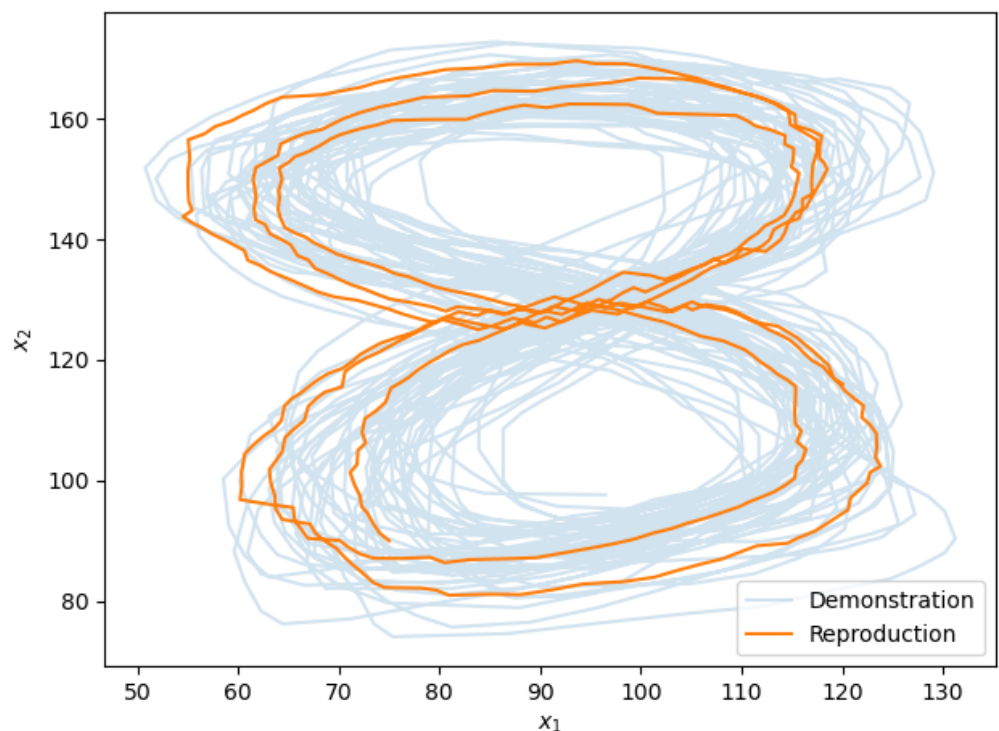


Figure 2: Imitation learning.

In [Figure 2](#) we can see that in the middle of the eight we have multiple modes: one velocity vector would lead to the left and one to the right. Sampling from the conditional GMM is only one possible solution here. Another one would be to select the component that contributes the most to the probability density and take its mean. When we sample, we often want to ensure that we do not end up in a region of low probability. Hence, we can resample as long as we are not in an α -confidence region, where $\alpha \in [0, 1]$. This strategy is used here and is directly provided by the library.

Acknowledgments

This work was supported through a grant of the German Federal Ministry of Economic Affairs and Energy (BMW, FKZ 50 RA 1701) and one grant from the European Commission (870142).

References

- Billard, A., Calinon, S., Dillmann, R., & Schaal, S. (2008). Robot programming by demonstration. In B. Siciliano & O. Khatib (Eds.), *Springer handbook of robotics* (pp. 1371–1394). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-30301-5_60
- Bishop, C. M. (1994). *Mixture density networks*. Aston University.
- Calinon, S., Guenter, F., & Billard, A. (2007). On learning, representing, and generalizing a task in a humanoid robot. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(2), 286–298. <https://doi.org/10.1109/TSMCB.2006.886952>
- Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1), 1–38. <http://www.jstor.org/stable/2984875>
- Ghahramani, Z., & Jordan, M. (1994). Supervised learning from incomplete data via an EM approach. In J. Cowan, G. Tesauro, & J. Alspector (Eds.), *Advances in neural information processing systems* (Vol. 6, pp. 120–127). Morgan-Kaufmann. <https://proceedings.neurips.cc/paper/1993/file/f2201f5191c4e92cc5af043eebfd0946-Paper.pdf>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Stulp, F., & Sigaud, O. (2015). Many regression algorithms, one unified model: A review. *Neural Networks*, 69, 60–79. <https://doi.org/10.1016/j.neunet.2015.05.005>
- Uhlmann, J. (1995). *Dynamic map building and localization: New theoretical foundations* [PhD thesis]. University of Oxford.