Open, Online, Interactive Visualizations for Learning About Gaussian Mixture Models

By
Stuart Allen

A THESIS

submitted to

Oregon State University

Honors College

in partial fulfillment of
the requirements for the
degree of

Honors Baccalaureate of Science in Computer Science
(Honors Scholar)

Presented May 31, 2023
Commencement June 2023

AN ABSTRACT OF THE THESIS OF

Stuart Allen for the degree of Honors Baccalaureate of Science in Computer Science presented on May 31, 2023. Title:
Open, Online, Interactive Visualizations for Learning About Gaussian Mixture Models

Abstract approved:

Stefan Lee

We construct a website to explain how Gaussian mixture models can be optimized using the expectation maximization algorithm. Previous free, online material on this process has been extremely limited. All sources surveyed failed to entirely describe our identified criteria for an in-depth description and useful visualizations. After surveying a variety of online sources, and researching attributes of useful visualizations for students, we use libraries including Threejs and React-Three-Fiber to construct a website that satisfies our identified criteria. The result is a free, open source website that anyone with a background in calculus and statistics can use to understand and implement Gaussian mixture models optimized with the expectation maximization algorithm. Further research may include more in depth evaluation methods of the effectiveness of this site, and implementations of other explanatory websites on lesser known machine learning topics using our methods. The site is available at the following link: https://gmmthesissite.netlify.app/

Key Words: Gaussian Mixture Models, Computer Science Education, Scientific Visualization

Corresponding e-mail address: allentsu@oregonstate.edu

Open, Online, Interactive Visualizations for Learning About Gaussian Mixture Models

By
Stuart Allen

A THESIS

submitted to

Oregon State University

Honors College

in partial fulfillment of
the requirements for the
degree of

Honors Baccalaureate of Science in Computer Science
(Honors Scholar)

Presented May 31, 2023
Commencement June 2023

Honors Baccalaureate of Science in Computer Science project of Stuart Allen presented on May 31, 2023.

APPROVED:

Stefan Lee, Mentor, representing College of Engineering

Jennifer Parham-Mocello, Committee Member, representing College of Engineering

---

Yue Zhang, Committee Member, representing College of Engineering

---

Toni Doolen, Dean, Oregon State University Honors College

I understand that my project will become part of the permanent collection of Oregon State University Honors College. My signature below authorizes release of my project to any reader upon request.

Stuart Allen, Author

Contents

1 Introduction

This thesis project creates a free, online, and interactive resource for undergraduate students to learn about how Gaussian mixture models (GMMs) are optimized. Creating free statistics, machine learning, and computer science content on the internet is inherently valuable, as it removes barriers to education for often difficult concepts. Web resources in particular allow for interactivity unlike other mediums such as a text book. Other free, online content on optimizing GMMs lacks important details on the subject. For example, most sources surveyed did not explain that there is no closed form solution for an optimal GMM using maximum likelihood estimation. Most sites therefore do not motivate the use of the expectation maximization algorithm, even if they explain its implementation.

In order to implement an educational site for explaining GMMs it is necessary to understand how they are optimized, survey existing content, and research qualities of useful visualizations. We review GMMs and how they are optimized in the background section. A survey of how existing content motivates our project as well as assists in generating criteria for useful visualizations. We then discuss methods of implementing our site based on these explanation criteria. We also generate and offer explanations on how our site implements aspects of useful visualizations from existing research. Finally, we review our product as well as provide suggestions for future research.

2 Background

2.1 Defining Gaussian Mixture Models

GMMs have an associated probability density function using a weighted sum of normal distributions. The probability density of a given point x is given by equation 1, where each μ_i, Σ_i, and π_i correspond to one of k individual d-dimensional Gaussian distributions within the model. The sum of all π_i is 1 in order for the equation to describe a valid probability. Each individual Σ_i is a positive, semi-definite matrix, in order to remain a valid covariance matrix. [1]

$$p(x|\mu, \Sigma, \pi) = \sum_{i=1}^{k} \pi_i \frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}} e^{-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)} \tag{1}$$

GMMs are extremely flexible in comparison to a single, multivariate Gaussian distribution. Our site's main example problem requires finding a probability density estimator for the likelihood of finding a mountain lion at any point on a map. As solitary creatures, an area with multiple mountain lions may be better estimated by a density function with multiple local maxima, and different covariance in different areas. GMMs are therefore apt for our example problem's probability density

estimator.

2.2 Optimizing Gaussian Mixture Models

There is no closed form solution for optimizing GMMs with maximum likelihood estimation (MLE). For example, equation 2 depicts the log-likelihood expression of a GMM comprised of 2, single-dimensional, Gaussian distributions. If we take the derivative of this expression with respect to any of the individual distributions parameters ($\mu_1, \mu_2, \sigma_1, \sigma_2, \pi_1$, or π_2) the result is an expression that has every other distribution parameter respectively. Even in this simple case, solving with MLE quickly results in a system of non-linear equations. There is no closed form solution, especially not with variable dimensionality and a variable number of individual Gaussian distributions.

$$\begin{aligned} &LL(\mu, \sigma, \pi) = log(\textstyle\prod_{i=1}^{j} p(x_j)) = \\ &log(\textstyle\prod_{i=1}^{j} \pi_1 \frac{1}{\sigma_1 (2\pi)^{1/2}} e^{-\frac{1}{2}(\frac{x_j - \mu_1}{\sigma_1})^2} + \pi_2 \frac{1}{\sigma_2 (2\pi)^{1/2}} e^{-\frac{1}{2}(\frac{x_j - \mu_2}{\sigma_2})^2}) \end{aligned} \tag{2}$$

This sufficiently motivates the use of an algorithm like the expectation maximization (EM) algorithm. In general the EM algorithm works by updating some of a problem's distribution parameters, while holding a latent variable constant. Next the latent variable is updated while the distribution parameters are held constant. This process repeats for some amount of iterations or until a local solution is reached. The EM algorithm is guaranteed to a find a local solution.

In order to use the EM algorithm, we must find a latent variable for our use case. The partial assignment of each datapoint n, given by equation 3, fits our problem criteria. This partial assignment describes the likelihood a data point comes from a particular individual distribution.

$$\gamma_n = [\frac{\pi_1 \frac{1}{(2\pi)^{n/2} |\Sigma_1|^{1/2}} e^{-\frac{1}{2}(x-\mu_1)^T \Sigma_1^{-1} (x-\mu_1)}}{p(x|\mu, \Sigma, \pi)}, ..., \frac{\pi_k \frac{1}{(2\pi)^{n/2} |\Sigma_k|^{1/2}} e^{-\frac{1}{2}(x-\mu_1)^T \Sigma_k^{-1} (x-\mu_k)}}{p(x|\mu, \Sigma, \pi)}] \tag{3}$$

Notice that it is trivial to calculate the partial assignment of each point given our distribution parameters. It is also trivial to update our distribution parameters given the partial assignments of our data set as shown in equations 4, 5, and 6. Keep in mind, N represents the total size of our data set, and any $\gamma_{(xy)}$ represents the partial assignment for data point x that corresponds to individual distribution y. In order to apply the EM algorithm to GMMs, updating the partial assignments given some initial distribution parameters is the Expectation step, and updating the distribution parameters is the Maximization step. These steps are repeated for a set number of iterations or until the log-likelihood converges to a local solution. This is how GMMs are typically optimized. [1]

$$\mu_k^* = \frac{1}{\sum_{n=1}^{N} \gamma_{(nk)}} \sum_{n=1}^{N} \gamma_{(nk)} x_n \tag{4}$$

$$\Sigma_k^* = \frac{1}{\sum_{n=1}^{N} \gamma_{(nk)}} \sum_{n=1}^{N} \gamma_{(nk)} (x_n - \mu_k^*)(x_n - \mu_k^*)^T \tag{5}$$

$$\pi_k^* = \frac{\sum_{n=1}^{N} \gamma_{(nk)}}{N} \tag{6}$$

3 Methods

3.1 Evaluation of Similar Works

This project began with an analysis of current online content that explains how GMMs are optimized. These sources came from the front page of Google as of September of 2022 when searching related queries. While many of these sources do describe necessary concepts for understanding how GMMs are optimized, all sites missed at least some subtleties of the topic. These missed details both major and minor motivate the creation of our site, as well as provide a grounds for eliciting criteria of GMM explanations.

3.1.1 *Gaussian Mixture Models Explained*

This site's author, Oscar Carrasco, derives the equations for the EM algorithm for GMMs, as well as an implementation in python in their web article. [2] Carrasco motivates GMMs by detailing how K-means uses hard clustering. They, however, fail to mention the greater flexibility of GMM's covariance properties. Carrasco is very precise about terminology and explanation towards the beginning of the article, but later in the article they introduce symbols without qualifying their meaning in words, such as $\theta*$ which is used in the explanation of the EM algorithm.

Carrasco steps through his implementation of the EM algorithm in Python. The explanation of his implementation is assisted with formulas and explanations. His particular implementation uses K-means to initialize cluster centers, and implies that other situations may need different solutions. He mentions he trained his implementation on the Iris dataset for simplicity, but does not mention the features of that dataset.

3.1.2 *Clustering (4): Gaussian Mixture Models and EM*

This video begins with an explanation as to how covariance can be useful for different clusterings. [3] Other motivations the author, Alexander Ihler, include for GMMs are using clustering as a generative model to collect new samples, compare train and test

sets, as well as estimate missing data points. Ihler explains the EM algorithm as a form of coordinate descent, similar to, but not the same as the K-means algorithm. His explanation is supported by text, equations, and commentary.

Ihler also describes how to evaluate a cluster's fit, and includes an example problem. Ihler describes the negative log-likelihood of the data will increase with more clusters, however, an optimal value of clusters can be found using complexity adjustment. Ihler then uses GMMs to cluster patients into anemic and not anemic groups based on their blood content. The video ends with a recap and some inspiration to use the EM algorithm for other problems in the future.

While Ihler offers many example applications, one particular concept of GMMs they ignore is the pitfall of singularities. Singularities within a GMM occur when the covariance of one individual distribution shrinks around one outlier or other data point. This is relevant as with a different initialization of the blood type data set, this is a possibility as there are several outliers.

3.1.3 *Gaussian Mixture Model*

This Geeks for Geeks article begins by explaining the flexibility of GMMs. [4] However they do not mention nearly all the benefits of GMMs such as the flexibility of covariance. The article mentions why optimization with MLE is difficult. There is some helpful code and commentary for using the Sci-Kit Learn implementation for optimizing GMMs. The site demonstrates the clustering abilities of GMMs, however, makes no reference to their use as probability density estimators.

3.1.4 *In Depth: Gaussian Mixture Models*

This explanation of GMMs begins with an implementation and a visualization of the shortcomings of K-means. [5] The author, Jake VanderPlas, describes some use cases of GMMs, but does not describe why the EM algorithm is necessary for optimization. Most of this book section uses charts, code, and corresponding written explanations to explain the algorithm. VanderPlas, describes how clusterings can be evaluated with cross validation, and how an optimal number of clusters can be chosen with AIC or BIC. The section encourages using GMMs as a density estimator rather than a clustering algorithm. The last section of the article uses a data set of handwritten number images, applies a dimensionality reduction method to the data set, optimizes a GMM for the transformed data, generates sample points from the GMM, then applies the inverse of the dimensionality reduction operation to generate images similar to the original data set.

3.1.5 *Understanding Gaussian Mixture Model*

This article gives a brief explanation of clustering in general and the steps of the EM algorithm. [6] It mentions the downfalls of K-means, as well as an introduction to

multivariate Gaussian distributions in general. The writers do explain the convergence this algorithm can reach, however, it ultimately never explains the necessity of the EM algorithm in the first place.

3.2 Explanation Criteria

This survey of existing content informs our decisions on explanation criteria in our site. All sites surveyed contained important aspects of any explanation, however, a site with the union of these aspects would create a more informative resource. We elicit admirable explanation qualities from these surveyed sites as well as how they are implemented in our resource in this section.

3.2.1 Motivating Density Estimation

GMMs make powerful probability density estimation models. Sites such as VanderPlas' included motivations such as using an optimized GMM for generating new points similar to an original data set. [5] We motivate density estimation in our site using our example problem.

Our example problem details how a hiker must choose a path that has the minimum chance of encountering a mountain lion, given the location of mountain lion sightings on a map. A GMM can be optimized for this data set to construct a probability density estimator of mountain lion sightings over the hiker's map. A Riemann sum line integral of each trail over this estimator reveals the safest path. A Riemann sum line integral visualization from our site can be seen in figure 1.

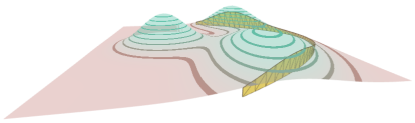

Figure 1: A visualization of a Riemann sum line integral along a GMM.

3.2.2 Motivating Clustering

Using GMMs for clustering gives a second use to the soft assignments generated by the EM algorithm. Sites like Ihler's demonstrate the effectiveness of optimized GMMs for clustering in their blood dataset. [3] The covariance properties of this distribution were necessary in order to create a viable clustering of their data.

We motivate clustering by having readers identify a 2D graph of points clustered by K-means or an optimized GMM. The example we provide highlights the importance of covariance in clustering methods, as the clustering generated by GMMs provides a more obvious fit. This example question is included in figure 2.

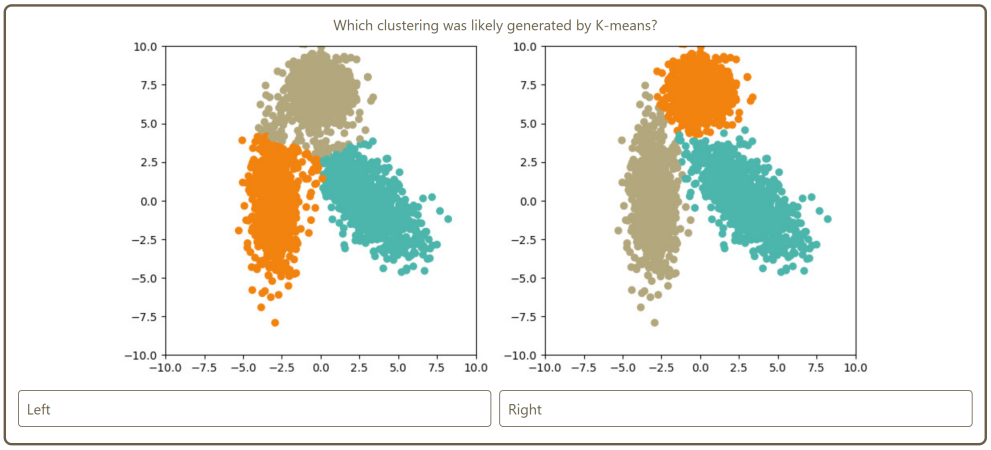

Figure 2: Students must identify which clustering was created using K-means.

3.2.3 Description of the Problems with Direct Optimization

Sites like the Geeks for Geeks article included an explanation of the difficulties of optimizing a GMM with MLE. [4] This criterion is necessary, as it is unwise to use the EM algorithm instead of MLE when possible. When MLE is viable, it yields a global solution with a closed form. This leads to less computational resources required in almost any case. The EM algorithm is useful for optimizing GMMs, but should not be used blindly.

We introduce students to problems with GMMs and MLE with a small scale problem. This smaller scale problem involves a GMM with two individual Gaussian distributions in a one dimensional space. Students must identify the derivative operator applied to the log-likelihood expression will result in an expression with all other distribution parameters involved, as seen in figure 3. We then explain this creates a system of non-linear equations that has no closed form solution.

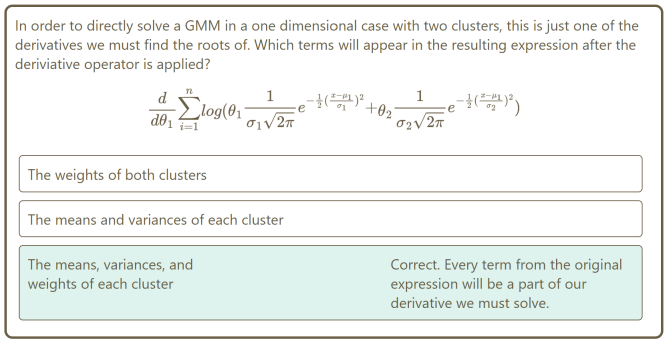

Figure 3: Students begin to understand why GMMs cannot be optimized with MLE by answering this question.

3.2.4 Description of Convergence

The EM algorithm is guaranteed to reach a local optimum solution. This means although the algorithm can achieve a global optimum it is unlikely for large datasets, and many individual distributions. This algorithm is also not robust to different initializations of distribution parameters. Some sources surveyed included this information, including *Understanding Gaussian Mixture Model*. [6] This information is integral to using GMMs in real world problems as different initialization strategies can lead to better representations of an original dataset. Our EM algorithm description and interactive visualization demonstrates the importance of knowing what kind of convergence the EM algorithm yields. Our site encourages restarting the visualization to see how different initializations can lead to varied results after optimization, thus students are aware that the EM algorithm only converges to a local solution. Figure 4 reflects a version of the EM visualization that resulted in a GMM that is representative of the data set.

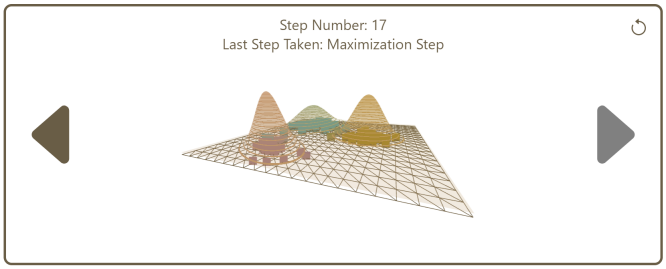

Figure 4: Our site's EM visualization that resulted in a GMM that is representative of the data set, for one particular initialization.

3.2.5 Description of Evaluating Model Fit

Model fit can be computed many ways, one of the simplest being the log likelihood of each data point. Regardless of loss metric, loss generally will decrease as the number of individual distributions increases until there are the same number of data points as individual distributions. However, a model will reach a point where adding individual distributions causes diminishing returns to the loss metric. This is generally when an ideal number of individual distributions has been reached for the given model. Not all sites mentioned model fit, however, VanderPlas included several metrics of evaluation. [5] Our site allows readers to visually evaluate a model's fit in the EM visualization, as seen in figure 4, and also with our explanation of log-likelihood.

3.2.6 Description of Singularities

No site surveyed included a description of singularities. This is an important criterion for data sets with many outliers. While, simple to mitigate by ensuring a minimum

value for diagonal entries of each covariance matrix in a GMM, readers should be prepared with this knowledge for their own implementations. We include this knowledge in the site by asking readers to select which probability density estimator is more representative of a dataset. One with a singularity and another obviously better fit alternative, as seen in figure 5.

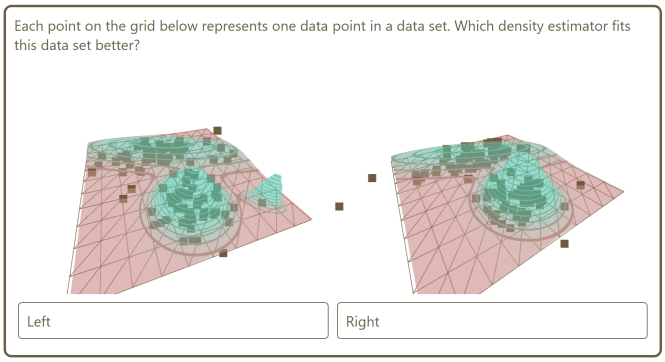

Figure 5: Students must identify a GMM with a singularity is not a better representation of a data set.

3.3 Visualization Criteria

We also elicit criteria for effective visualizations for computer science education with existing material on the subject. Most sites surveyed included some kind of visualization, though almost none had any interactive component. Instead, our site implements the findings of existing research for effective visualizations. We discuss these findings and their implementation in our site in this section.

3.3.1 Active Learning Principles

The effectiveness of Algorithm Visualizations (AVs) may be enhanced by understanding different pedagogical theories. In their metastudy, Christopher Hundhausen and other researchers found that the most studied pedagogical theories in terms of AVs include epistemic fidelity, the idea that student's learn by creating symbolic models of the real world in their mind, and cognitive constructivism, the idea students construct their understanding of the world through personal experience. [7] In either case, Hundhausen found that students benefit most from how an AV is used by students rather than the actual visuals students see. Their results and both of these mental models suggest that active learning activities support students best. Students who have to engage with an AV, rather than passively viewing an animation, will certainly include the activity in their subjective understanding of the world, or better manipulate a symbolic analog of an algorithm in their mind.

We engage students in active learning with our visualizations. All visualizations have camera controls over two dimensional functions at the very least. Visualizations that have only this much interactivity are meant to demonstrate an example of the particular type of 2D function. All other visualizations are accompanied by questions, as seen in figure 5, to help the student build either symbolic models or personal experience with different aspects of GMMs.

3.3.2 Meaningful Engagement in Visualization

There are many ways to use interactivity in AVs to increase positive outcomes for a student. Generally, AVs that use no form of student interaction do not increase positive student outcomes any more than a normal lecture. [8] Fouh and others suggest instead that there are multiple forms of student interactivity, including but not limited to, simply responding to questions or forming predictions about an algorithm, and presenting the information to another student. Furthermore they venture that creating presentations is the best form of interactivity, even better than a student creating their own visualization. This matches others findings that active learning activities that use AVs tend to benefit students. [9] Any sort of independence students are granted with AVs is a great supplement to lecture or article.

This also motivates our inclusion of multiple choice questions, as seen in figure 5. This is a form of interactive feedback on the web that is natural to implement. Readers also engage in predicting when stepping through the EM visualization, as seen in figure 4, as they observe how effective different initializations can lead to different results.

3.3.3 Intuitive Controls and UI

It is important to avoid common pitfalls when creating engaging activities for students. If using an AV includes monotonous work, students may grow resentful of the activity. It is also best to keep AVs as simple as possible so students spend more time understanding the subject of the AV rather than its visual representation. One method to mitigate monotony is using canned examples, predetermined inputs meant to demonstrate special cases of an algorithm. Canned examples with an AV can expand a student's knowledge as they may not have tried all important cases of an algorithm on their own. [10] Saraiya and others identified that useful AVs included intuitive UIs, control of speed and playback of animations, and pseudocode in their paper *Effective Features of Algoritmic Visualizations*.

This motivates our inclusion of back navigation throughout the site, as well as comparing canned examples. Readers are asked to differentiate between better and worse probability density estimators in order to understand the flexibility of GMMs. Students are allowed to reanswer based on feedback for all questions. The page behavior when selecting a wrong answer can be seen in figure 6. The EM visualization also allows students to navigate both forward and backwards at their desired pace

with click-through navigation, and restart multiple times, as seen by the arrows and restart symbol in figure 4.

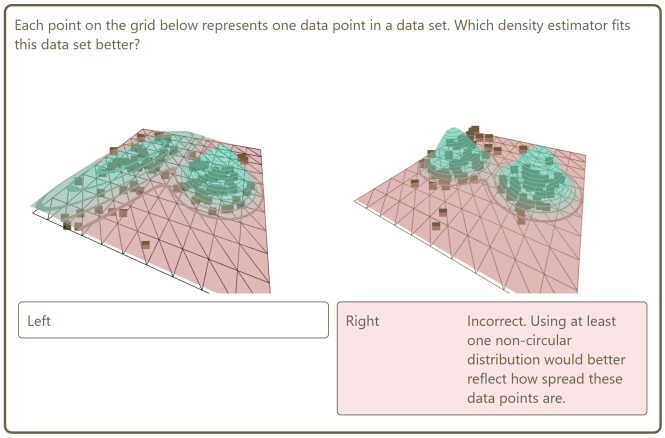

Figure 6: Students can answer questions multiple times.

4 Results

We yield a website that fits our criteria for in depth GMM explanations as well as effective visualizations. These explanation criteria are a motivation of density estimation and clustering, a description of the EM algorithm's convergence, evaluating model fit, singularities, and problems with direct optimization. These visualization criteria include using active learning principles, meaningful engagement, and intuitive controls in a visualization. This website is a more complete and comprehensive version of existing free, online content that also uses effective methods of creating useful visualizations for students.

Our website is constructed with React, React-Three-Fiber, Threejs, WebGL and hosted with Netlify. React is a javascript framework that promotes compartmentalization and reusable components in a web page. We chose React as much of our site requires many visualizations that are similar, but include key and subtle changes. React-Three-Fiber allows Threejs components and scenes to be developed and rendered efficiently, which suits a React site with multiple visualizations. Threejs is a library that provides a more readable framework for using WebGL, however we still use WebGL for vertex and fragment shaders throughout the site. Lastly, Netlify is able to host our site without charge as this site has no need for a back end or database component.

The site is available here at the following link: https://gmmthesissite.netlify.app/

5 Further Research

Further analysis of the effectiveness of this site at teaching about optimizing GMMs is possible. One possible method may include creating a questionnaire of content on GMMs from our explanation criteria. This questionnaire could act as a pre and post assessments for a random sample of computer science or statistics students who would be shown our site. Some students could be shown different sites or content from our survey analysis to see the overall efficacy of our site compared to others. Lastly, seeing how students with a low grade point average perform when shown our site in comparison to others may confirm Saraiya and others' research that visualizations with intuitive UI disproportionately help students with lower grade point averages. [10]

References

[1] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2016.

[2] O. C. Carrasco, "Gaussian mixture models explained," 06 2019.

[3] A. Ihler, "Clustering (4): Gaussian mixture models and em," https://www.youtube.com/watch?v=qMTuMa86NzU, 03 2015.

[4] "Gaussian mixture model," https://www.geeksforgeeks.org/gaussian-mixture-model/.

[5] J. VanderPlas, *Python Data Science Handbook*. O'Reilly Media, Inc, 2023.

[6] G. L. Team, "Understanding gaussian mixture model," 11 2022.

[7] C. D. HUNDHAUSEN, S. A. DOUGLAS, and J. T. STASKO, "A meta-study of algorithm visualization effectiveness," *Journal of Visual Languages Computing*, vol. 13, no. 3, pp. 259–290, 2002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1045926X02902375

[8] E. Fouh, M. Akbar, and C. Shaffer, "The role of visualization in computer science education," *Computers in the Schools*, vol. 29, pp. 95–117, 01 2012.

[9] D. Schweitzer and W. Brown, "Interactive visualization for the active learning classroom," *SIGCSE Bull.*, vol. 39, no. 1, p. 208–212, mar 2007. [Online]. Available: https://doi.org/10.1145/1227504.1227384

[10] P. Saraiya, C. A. Shaffer, D. S. McCrickard, and C. North, "Effective features of algorithm visualizations," *Proceedings of the 35th SIGCSE technical symposium on Computer science education*, Mar 2004.