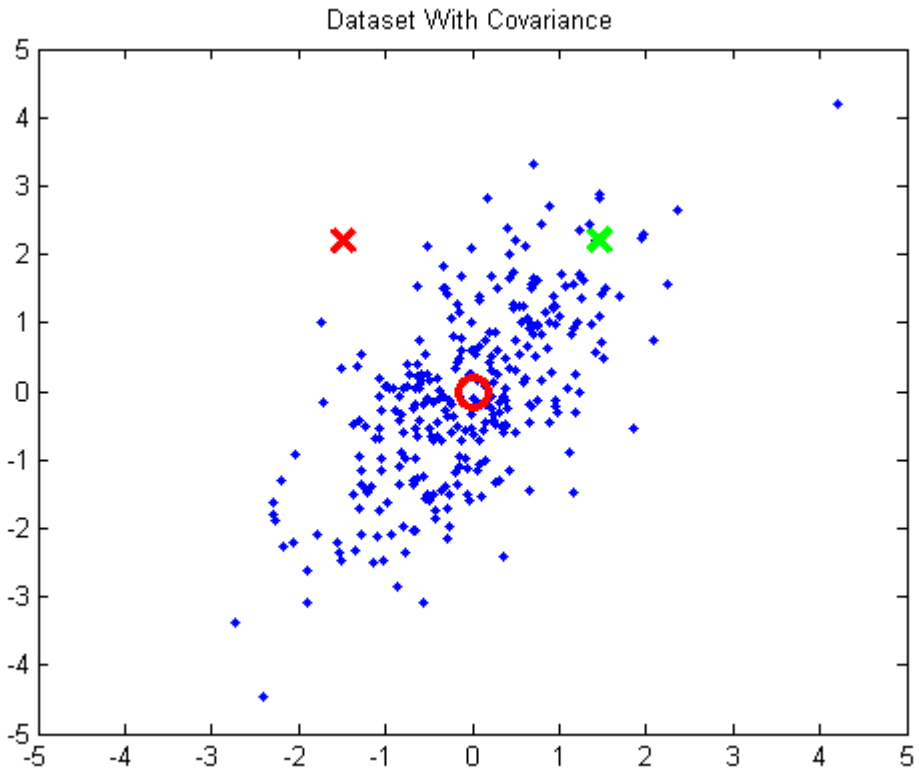
Gaussian Mixture Models Tutorial and MATLAB Code

04 Aug 2014

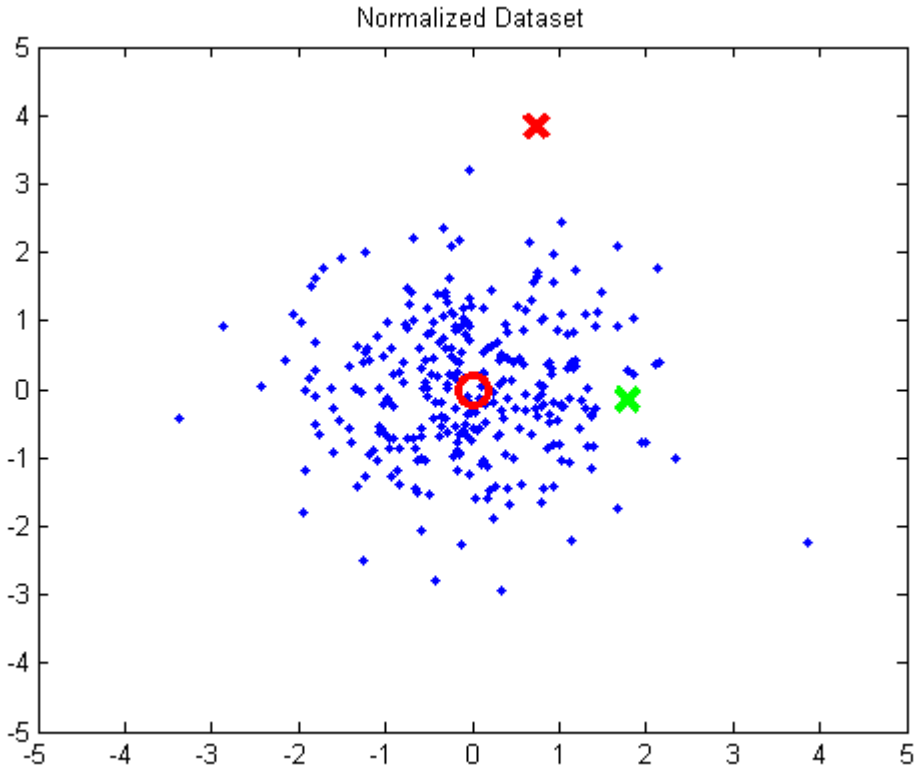
You can think of building a Gaussian Mixture Model as a type of clustering algorithm. Using an iterative technique called Expectation Maximization, the process and result is very similar to k-means clustering. The difference is that the clusters are assumed to each have an independent Gaussian distribution, each with their own mean and covariance matrix.

Comparison To K-Means Clustering

When performing k-means clustering, you assign points to clusters using the straight Euclidean distance. The Euclidean distance is a poor metric, however, when the cluster contains significant covariance. In the below example, we have a group of points exhibiting some correlation. The red and green x's are equidistant from the cluster mean using the Euclidean distance, but we can see intuitively that the red X doesn't match the statistics of this cluster near as well as the green X.



If you were to take these points and normalize them to remove the covariance (using a process called whitening), the green X becomes much closer to the mean than the red X.



The Gaussian Mixture Models approach will take cluster covariance into account when forming the clusters.

Another important difference with k-means is that standard k-means performs a hard assignment of data points to clusters–each point is assigned to the closest cluster. With Gaussian Mixture Models, what we will end up is a collection of independent Gaussian distributions, and so for each data point, we will have a probability that it belongs to each of these distributions / clusters.

Expectation Maximization

For GMMs, we will find the clusters using a technique called "Expectation Maximization". This is an iterative technique that feels a lot like the iterative approach used in k-means clustering.

In the "Expectation" step, we will calculate the probability that each data point belongs to each cluster (using our current estimated mean vectors and covariance matrices). This seems analogous to the cluster assignment step in k-means.

In the "Maximization" step, we'll re-calculate the cluster means and covariances based on the probabilities calculated in the expectation step. This seems analogous to the cluster movement step in k-means.

Initialization

To kickstart the EM algorithm, we'll randomly select data points to use as the initial means, and we'll set the covariance matrix for each cluster to be equal to the covariance of the full training set. Also, we'll give each cluster equal "prior probability". A cluster's "prior probability" is just the fraction of the dataset that belongs to each cluster. We'll start by assuming the dataset is equally divided between the clusters.

Expectation

In the "Expectation" step, we calculate the probability that each data point belongs to each cluster.

We'll need the equation for the probability density function of a multivariate Gaussian. A multivariate Gaussian ("multivariate" just means multiple input variables) is more complex because there is the possibility for the different variables to have different variances, and even for there to be correlation between the variables. These properties are captured by the covariance matrix.

$$g_j(x) = \frac{1}{\sqrt{(2\pi)^n |\Sigma_j|}} e^{-\frac{1}{2}(x-\mu_j)^T \Sigma_j^{-1}(x-\mu_j)}$$

Symbol	Meaning
$g_j(x)$	The PDF of the multivariate Gaussian for cluster j; the probability of this Gaussian producing the input x
j	Cluster number
x	The input vector (a column vector)
n	The input vector length
Σ_j	The n x n covariance matrix for cluster j
$\lvert\Sigma_j\rvert$	The determinant of the covariance matrix
Σ_j^{-1}	The inverse of the covariance matrix

The probability that example point i belongs to cluster j can be calculated using the following:

$$w_j^{(i)} = \frac{g_j(x)\phi_j}{\sum_{l=1}^{k} g_l(x)\phi_l}$$

Symbol	Meaning
$w_j^{(i)}$	The probability that example i belongs to cluster j
$g_j(x)$	The multivariate Gaussian for cluster j
ϕ_j	The “prior probability” of cluster j (the fraction of the dataset belonging to cluster j)
k	The number of clusters

We’ll apply this equation to every example and every cluster, giving us a matrix with one row per example and one column per cluster.

Maximization

You can gain some useful intuition about the maximization equations if you’re familiar with the equation for taking a weighted average. To find the average value of a set of *m* values, where you have a weight _w _defined for each of the values, you can use the following equation:

$$\bar{y} = \frac{\sum_{i=1}^{m}(w_i y_i)}{\sum_{i=1}^{m} w_i}$$

With this in mind, the update rules for the maximization step are below. I've copied these from the lecture notes on GMMs for Stanford's CS229 course on machine learning (those lecture notes are a great reference, by the way).

$$\begin{aligned}
\phi_j &:= \frac{1}{m}\sum_{i=1}^{m} w_j^{(i)}, \\
\mu_j &:= \frac{\sum_{i=1}^{m} w_j^{(i)} x^{(i)}}{\sum_{i=1}^{m} w_j^{(i)}}, \\
\Sigma_j &:= \frac{\sum_{i=1}^{m} w_j^{(i)} (x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^T}{\sum_{i=1}^{m} w_j^{(i)}}
\end{aligned}$$

The equation for mean (mu) of cluster j is just the average of all data points in the training set, with each example weighted by its probability of belonging to cluster j.

Similary, the equation for the covariance matrix is the same as the equation you would use to estimate the covariance of a dataset, except that the contribution of each example is again weighted by the probability that it belongs to cluster j.

The prior probability of cluster j, denoted as phi, is calculated as the average probability that a data point belongs to cluster j.

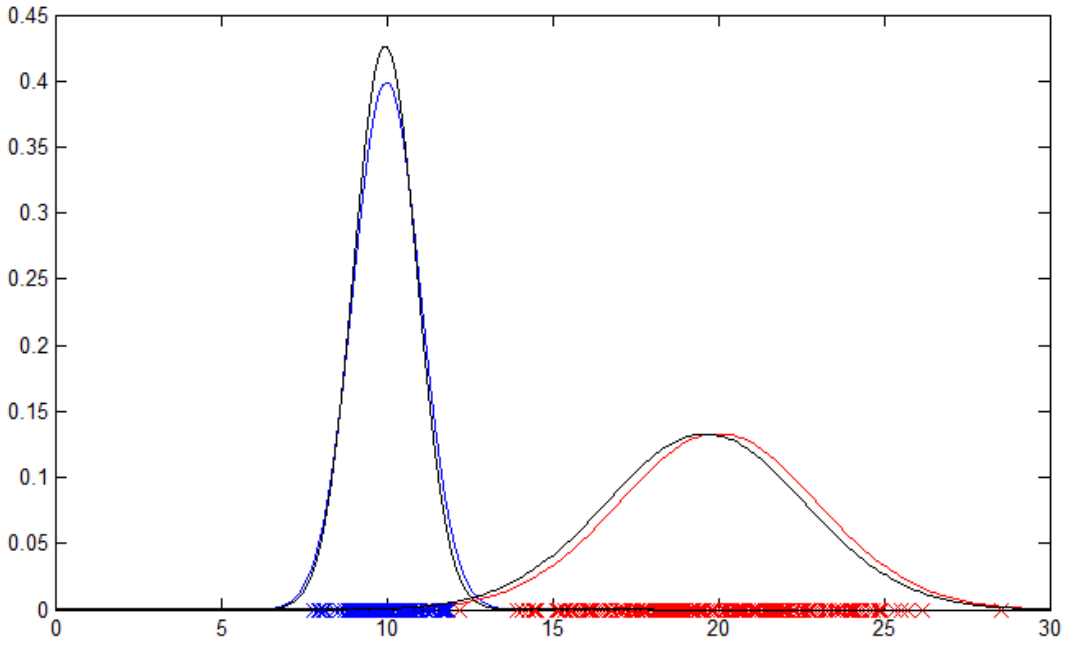
MATLAB Example Code

I've implemented Expectation Maximization for both a 1D and a 2D example. Run 'GMMExample_1D.m' and 'GMMExample_2D.m', respectively. The 1D example is easier to follow, but the 2D example can be extended to n-dimensional data.

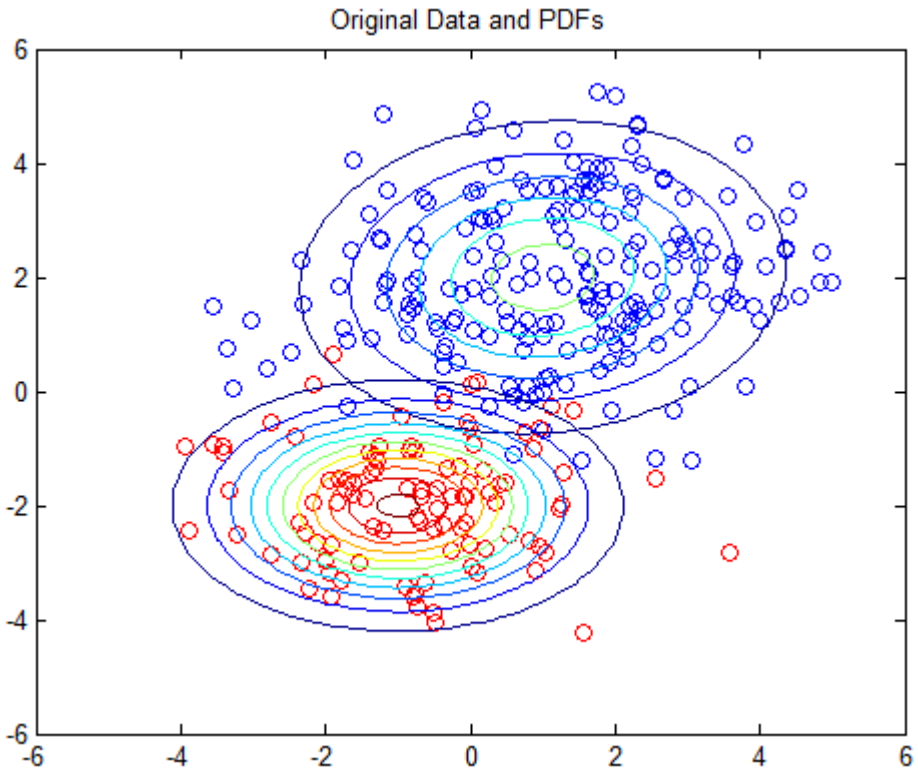
GMM Example Code

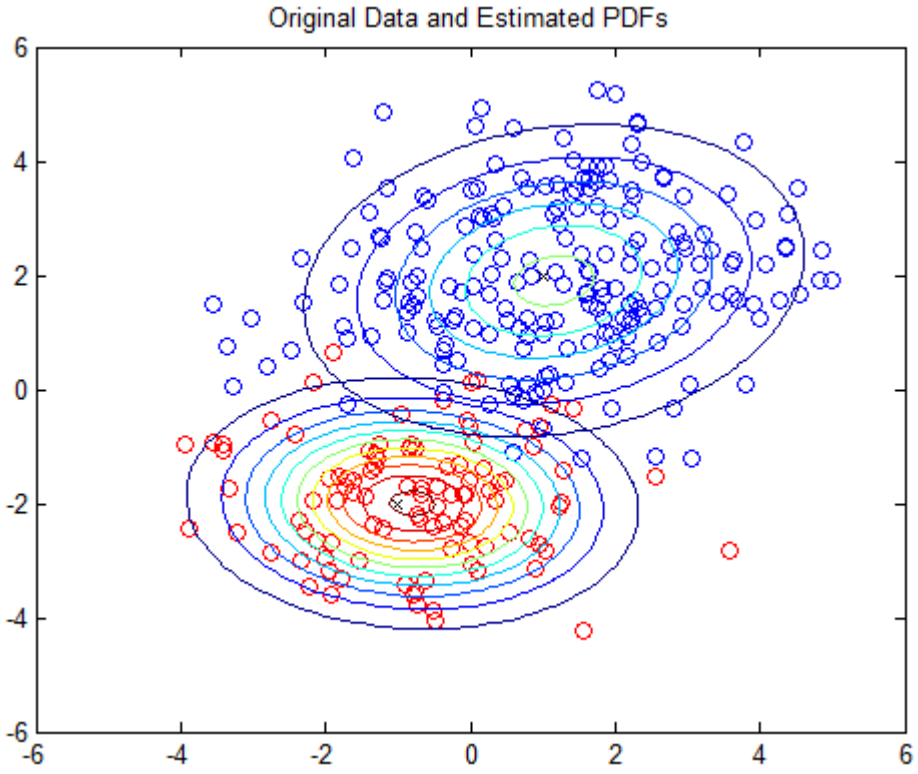
If you are simply interested in using GMMs and don't care how they're implemented, you might consider using the vlfeat implementation, which includes a nice tutorial here. Or if you are using Octave, there may be an open-source version of Matlab's 'fitgmdist' function from their Statistics Toolbox.

The 1D example will output a plot showing the original data points and their PDFs in blue and red. The PDFs estimated by the EM algorithm are plotted in black for comparison.



The 2D example is based on Matlab's own GMM tutorial here, but without any dependency on the Statistics Toolbox. The 2D example plots the PDFs using contour plots; you should see one plot of the original PDFs and another showing the estimated PDFs.


Original Data and PDFs
6
4
2
0
-2
-4
-6
-6
-4
-2
0
2
4
6


Original Data and Estimated PDFs
6
4
2
0
-2
-4
-6
-6
-4
-2
0
2
4
6

ALSO ON **MCCORMICKML.COM**

Key Concepts & Sources · Chris …

2 years ago • 5 comments

BERT Research - Ep. 1 - Key Concepts & Sources 1. Introduction In this …

Speed Up BERT Training · Chris …

2 years ago • 4 comments

Smart Batching Tutorial - Speed Up BERT Training In this blog post / Notebook, …

Negative Sampl Chris McCormic

5 years ago • 129 cor

Word2Vec Tutorial Negative Sampling of the word2vec tut

36 Comments **mccormickml.com** **Privacy Policy** **Login**

Favorite 5 Tweet Share **Sort by Best**

Join the discussion…

LOG IN WITH

OR SIGN UP WITH DISQUS

Name

Nityasri Reddy • 2 years ago

Hi chris,
How can I use Guassian 2D code for an n dimensional data? Do i have to repeat the lines written for 2D for n times for n dimensions or is there another way of doing that?

Reply • Share ›

Robin Thomas • 2 years ago

Hi Chris,

I have a set of sensor values [x y]= [467021 478610], [464025 479352], [465688 478515], [464025 478610] etc..around a ground ground truth [x y]= [466111 478611]. I have to find 'how is the distribution of points by the sensor' around the ground truth. Which distribution is helpful?

I have plans to work on Expectation Maximization [EM] and clustering using Gaussian mixture model (GMM) Algorithms.

Do i need to transform my input sensor values into Gaussian by applying normalization or any power transform techniques, before applying EM/ GMM algorithm.
OR

Can i use the same [x y] values directly to EM/GMM algorithm without normalization/ standardization techniques as both the values are almost on the scale.

I am beginner in this field. Help me please.....

Reply • Share ›

drisya p • 2 years ago

can you give me the code for background subtraction from an image using gaussian mixture model

Reply • Share ›

Shafi • 4 years ago

Can you help me by providing background subtraction method using your GMM code?

Reply • Share ›

Kemas10 Alfaris • 4 years ago

Hi, can you give me a numerical calculation about this method? Thanks

Reply • Share ›

aisyah • 4 years ago

Hi Chris,

If i want to use the initial value for parameter covariance matrix, i would use kmeans to estimate the initial values in matlab.

This is my code:
fData = faithfuldat;
[idx,C] = kmeans(Xobserve,2);
theta.mu = C';
theta.tau = hist(idx,2)/length(idx);
for j = 1:K
theta.Sigma(:,:,j) = cov(fData (idx==j,:));
end

However, unluckily the covariance matrix generates these values:

(:,:,1) =

0.1722 0.7471

see more

Reply • Share ›

Anwar Annas • 5 years ago

Hi chris

I would like ask to you. Could you explain to me Number of data point in code"GMMExample_2D"? I still confuse about that?

Thank you

Reply • Share ›

Chris McCormick Mod ➚ Anwar Annas • 5 years ago

I just picked an arbitrary number of sample points to generate for the purpose of the example. There's no real significance to the number.

Reply • Share ›

Anwar Annas ➚ Chris McCormick • 5 years ago

Thank for you reply.. If I have 2 matrix (m x n), can the number of sample points be generated from m x n?

Thank you

Reply • Share ›

Chris McCormick Mod ➚ Anwar Annas • 5 years ago

If you already have a dataset (your m x n matrix) that you're trying to cluster, then you don't need to generate any points. Generating points isn't part of the GMM process, it's just part of my simple example.

Reply • Share ›

Pratik Raj • 5 years ago • edited

Hi Chris,
I have to use GMM for supervised learning for my project(Language identification), do you have code for that in matlab. Can you please help me out ??
data description - audio files of 10 different languages (almost 100 mb data for each language)

Reply • Share ›

Chris McCormick Mod ➚ Pratik Raj • 5 years ago

Hi Pratik,
Before you perform supervised learning on some data, you first need to perform some feature extraction on the data to get good feature vectors.

I'm not familiar with speech processing, but I believe an old standard for feature extraction on speech is MFCC (Mel-frequency cepstral oefficients); there's a tutorial here.

Good luck!

Reply • Share ›

chebbi safa • 5 years ago

Hi Chris,
Thank you very much for the very simple and interesting tutoriel.
Can you please mention a simple numeric example with a vector of acoustic features describing a voice signal.
Thanks in advance

Reply • Share ›

Chris McCormick Mod ➚ chebbi safa • 5 years ago

Hi chebbi,

I'd like to provide some more GMM examples someday, but I don't have any currently, sorry!

Reply • Share ›

rohith mars • 5 years ago

Hi Chris,

Thanks a lot for the amazing information. The concept is explained in a lucid way and the MATLAB codes are very clear.

Can you please throw more light on the convergence of the EM estimation? I see in your code that you compare the estimated means of current and previous iteration and check the equality? Is there any better way to do this? Sometimes we stop the iterations based on certain thresholds of difference between estimated parameters. Do you think it is valid in this case as well?

Also can you explain in a similar way the log likelihood estimations used in standard EM algorithms? Why have you not chosen the log likelihood estimations here? Hope you understand my query.

Thanks in advance. Please keep the posts coming. They are very helpful.

Reply • Share ›

Chris McCormick Mod ➚ rohith mars • 5 years ago • edited

Hi rohith,

Thanks for the kind words!

Yes, there are many valid ways of choosing when to stop clustering algorithms such as GMM. Using a threshold is a pretty practical method since beyond a certain point the extra iterations may not be adding much value to the results.

I'm not familiar with the log likelihood approaches, sorry! Maybe I'll take a look at them if I get a chance to revisit this topic.

take a look at them if I get a chance to revisit this topic.

Reply • Share ›

Zhaleh Manbari • 5 years ago

Hi Chris.
I'm Master Student in Artificial intelligence.
for my thesis, i need to study mixture of expert.(or mixture of Gaussian).
I work at high dimension Data, and i want to fit GMM or other mixture of models on these data.
can you guide that how to do it?
I download your code GMM. but this code is for 2D.
thanks a lot.

Reply • Share ›

Chris McCormick Mod ➚ Zhaleh Manbari • 5 years ago

Hi Zhaleh,
It's possible to modify the code to support an arbitrary number of dimensions. I'd like to create an example of that, but I don't know that I will get to it soon, sorry!

1 Reply • Share ›

Owais Khan • 5 years ago

Hi, Can you please share with some helpful data or notes regarding clustering of wheat seeds using Expectation Maximization. Unfortnately, i didnt find any book or other source, from where i can get sufficient details about this algorithm, its architecture and its MATLAB Code.

Reply • Share ›

Darsana Malyakkal Premanand • 5 years ago

Hi, excellent article!!!
I have a doubt, can i use the same method given in the matlab code to detect road from an image captured by a UAV Vehicle?

Reply • Share ›

Chris McCormick Mod ➚ Darsana Malyakkal Premanand
• 5 years ago • edited

Thanks, glad it was helpful!
On road detection--I'm not sure! I think that's pretty far outside the scope of this article.

Reply • Share ›

disqus_LSIIOBmjco • 5 years ago

Hi Chris, super helpful article, thank you, any idea of the approach you would take to modify the GMM for use in voice conversion?

Reply • Share ›

Chris McCormick Mod ↗ disqus_LSIIOBmjco • 5 years ago

I'm not really familiar with that, sorry! Thanks for the kind words, though!

^ | v • Reply • Share ›

AJ • 5 years ago

Very easy to understand. Thank you. How do I prepare my data if I want to model face images to apply for face verification? Did I uses the feature of each enrolled individual and also the unknown individual?

^ | v • Reply • Share ›

Chris McCormick Mod ↗ AJ • 5 years ago

Not sure--do you already have a feature extraction algorithm in place for representing the faces as vectors? And you just want to use a GMM as a classifier to determine who a face belongs to?

^ | v • Reply • Share ›

N_Ti ↗ Chris McCormick • 5 years ago

So the vectors will come from the feature extraction? Yes, I want to use GMM as classifier. I'd read about GMM-UBM based speaker verification. This is what I try to implement. Is it I'm going to right path?

^ | v • Reply • Share ›

Zaynab • 5 years ago

Thank you
Can you help me for apply this algorithm to detect to detect two thresholds on a gray level image (in MATLAB)

^ | v • Reply • Share ›

Chris McCormick Mod ↗ Zaynab • 5 years ago

Not sure what you mean exactly by detecting thresholds--could you clarify?

^ | v • Reply • Share ›

Zaynab ↗ Chris McCormick • 5 years ago

I want to detect the edge of my gray scale image..For this reason, i have an histogram that it present a guassien mixture model.intersection of models indicates the threshold value.
PS: Image and his histogram
View — uploads.disquscdn.com View — uploads.disquscdn.com
Thank you for your answer

^ | v • Reply • Share ›

Chris McCormick Mod ↗ Zaynab • 5 years ago

I see, so you're trying to identify that value around 0.12 where the two Gaussian distributions intersect? Once you've learned the parameters for the two distributions, could you set the two Gaussian equations equal to one another and solve for x?

Reply • Share ›

Zaynab ➚ Chris McCormick • 5 years ago

Yes i want to do this automatically ;how??

Reply • Share ›

Chris McCormick Mod ➚ Zaynab • 5 years ago

Does this help?

Reply • Share ›

Aziz shammas • 6 years ago • edited

very helpful and simplified explanation, thank you!

Reply • Share ›

Related posts

Combining Categorical and Numerical Features with Text in BERT 29 Jun 2021

How To Build Your Own Question Answering System 27 May 2021

2020 NLP and NeurIPS Highlights 23 Mar 2021