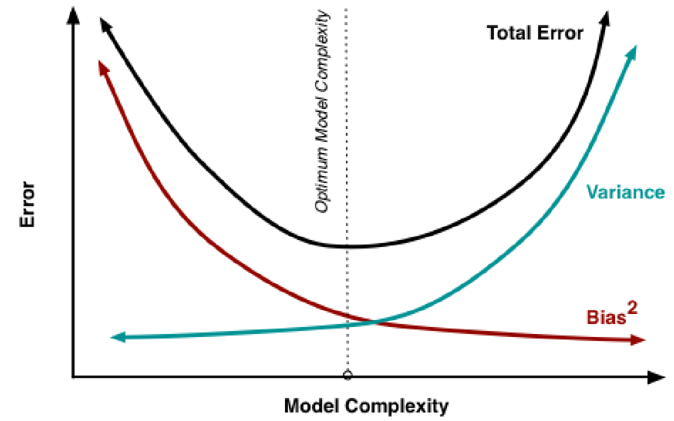
GMM & EM

And some RoC

Last time summary

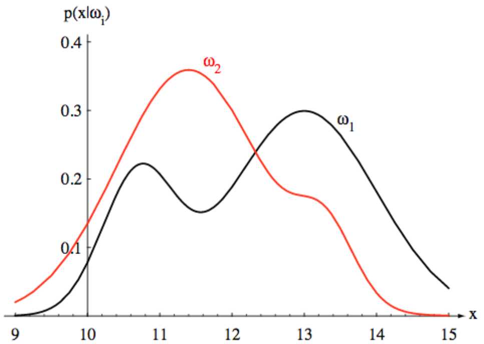
- Bias-Variance trade-off
  - Overfitting and underfitting
- MLE vs MAP estimate
  - How to use the prior
- LRT (Bayes Classifier)
  - Naïve Bayes

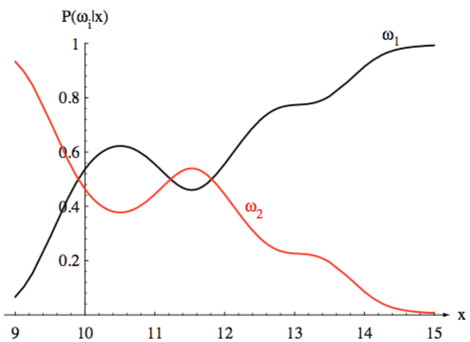


$$\boxed{\frac{P(x|w_1)}{P(x|w_2)}} \quad ? \quad \boxed{\frac{P(w_2)}{P(w_1)}}$$

Likelihood ratio

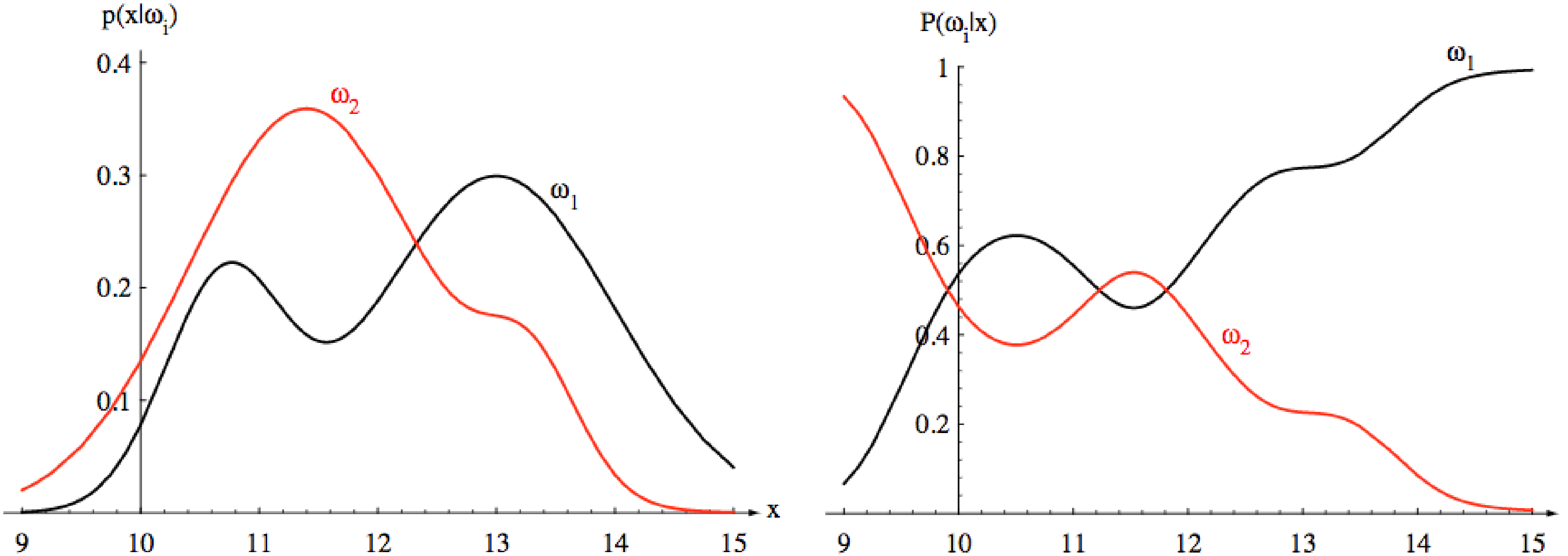
Ratio of priors





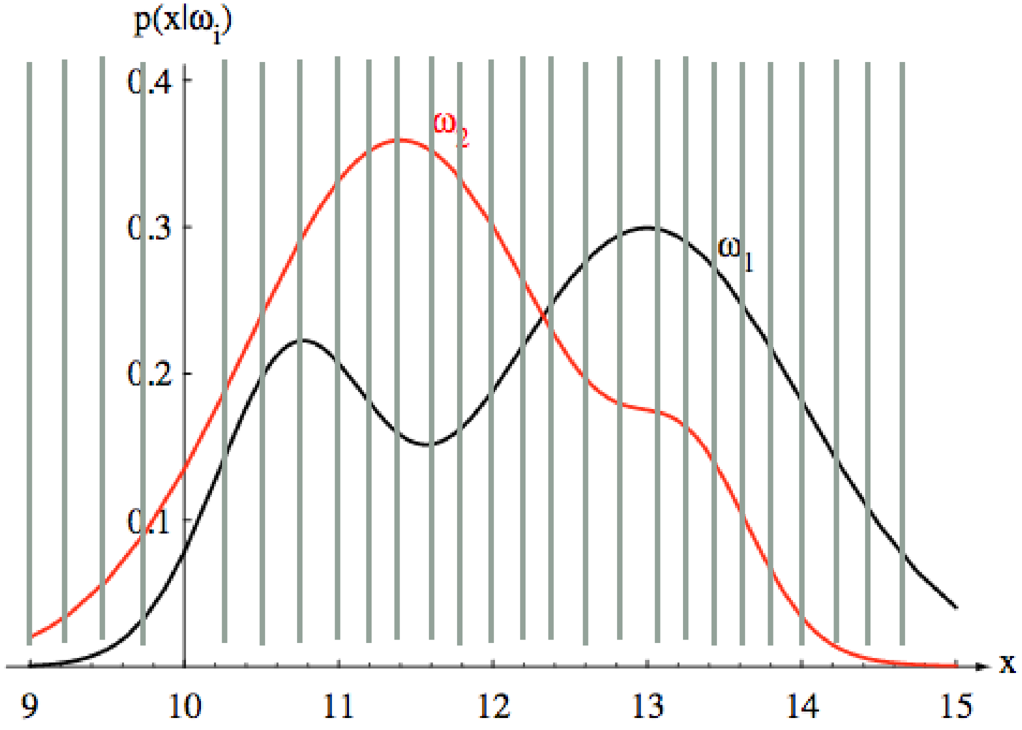
A simple decision rule

- If we can know either p(x|w) or p(w|x) we can make a classification guess



Goal: Find p(x|w) or p(w|x) by finding the parameter of the distribution

A simple way to estimate p(x|w)



Make a histogram!

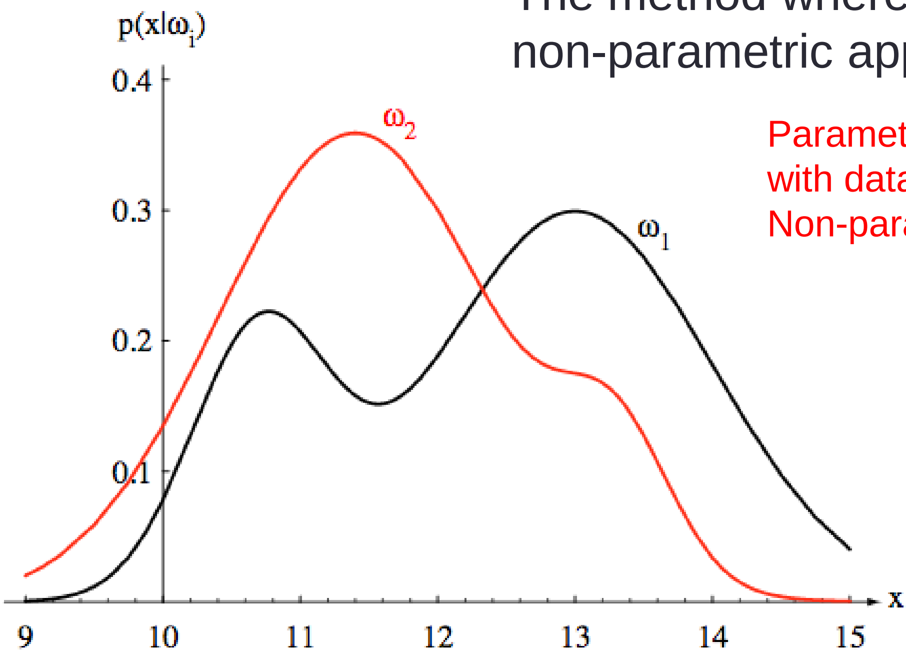
What happens if there is no data in a bin?

The parametric approach

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2}$$

- We assume p(x|w) or p(w|x) follow some distributions with parameter θ

The method where we find the histogram is the non-parametric approach



Parametric: need to assume the distribution (might not fit well with data)
Non-parametric: might encounter sparse bins

Goal: Find θ so that we can estimate p(x|w) or p(w|x)

Maximum Likelihood Estimate (MLE)

$$p(w_i|x) = \frac{p(x|w_i)p(w_i)}{p(x)}$$

$$\text{Posterior} = \frac{\text{likelihood} * \text{prior}}{\text{evidence}}$$

- Maximizing the likelihood (probability of data given model parameters)

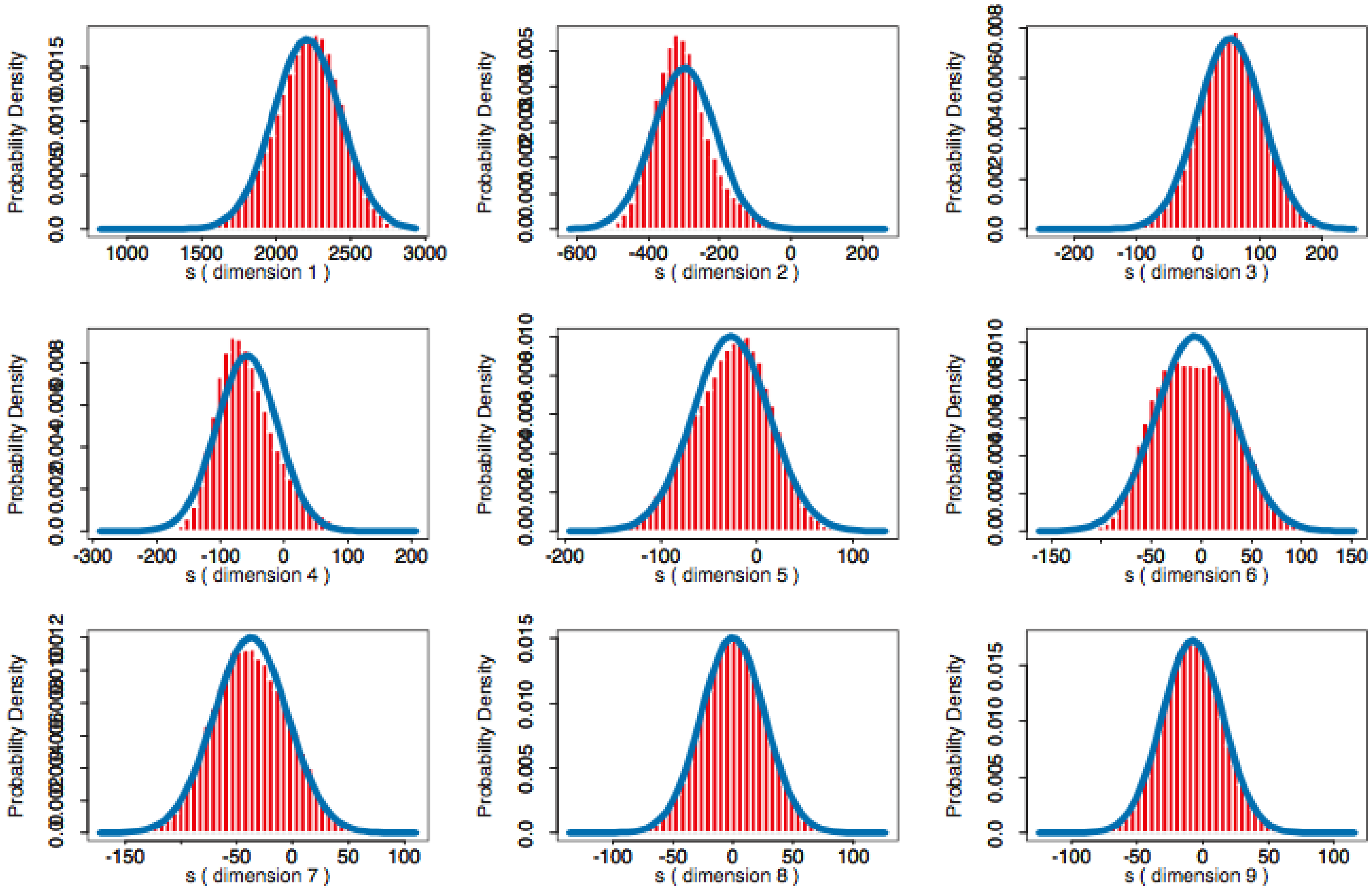
$p(\mathbf{x}|\theta) = L(\theta)$ <- This assumes the data is fixed

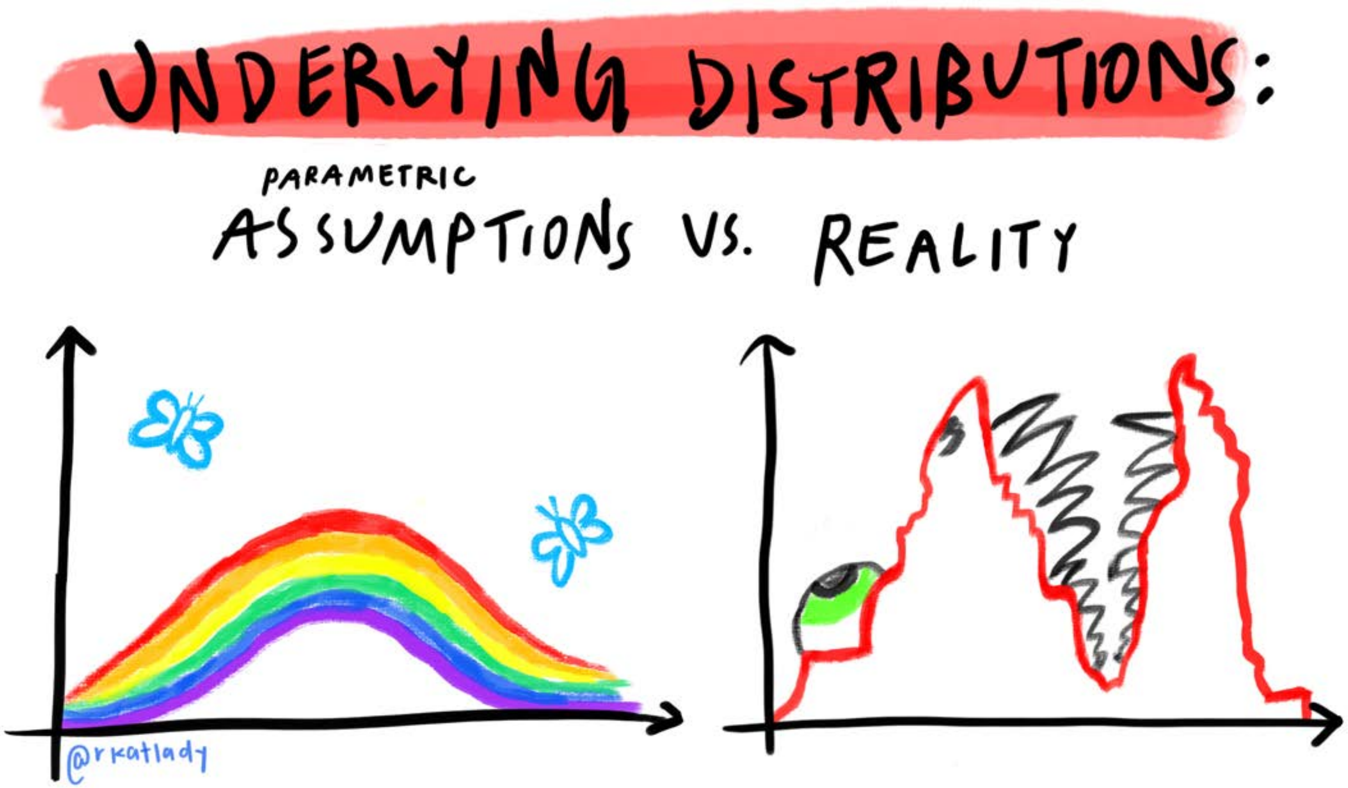
- Usually done on log likelihood

- Take the partial derivative wrt to θ and solve for the θ that maximizes the likelihood

Model of one Gaussian

First 9 MFCC's from [s]: Gaussian PDF



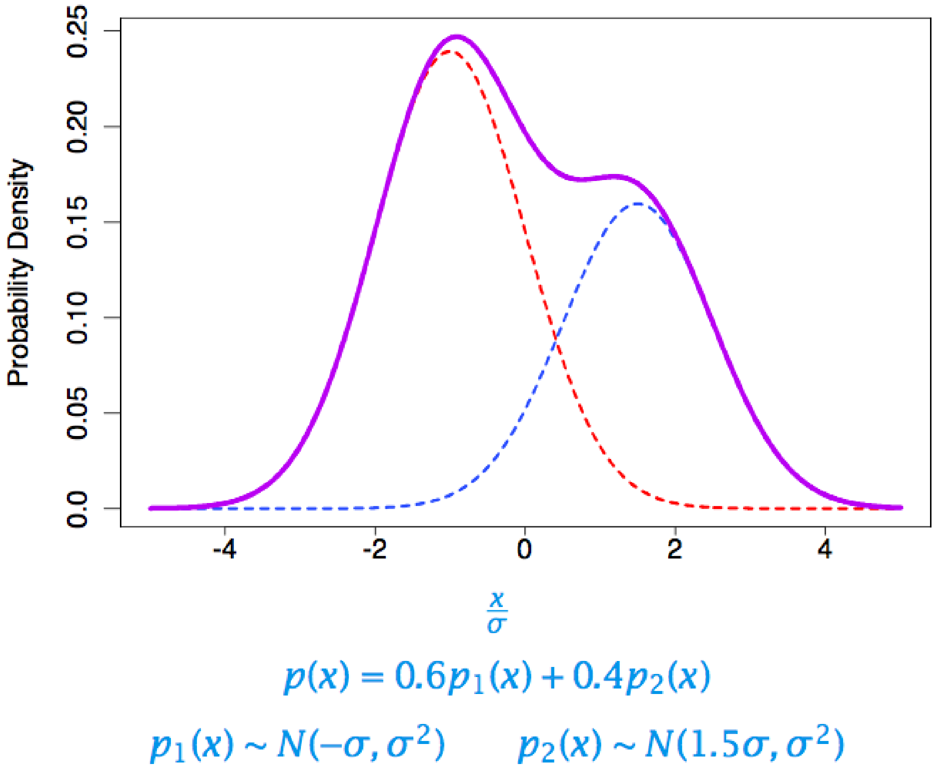


https://www.khstats.com/blog/tmle/tutorial/

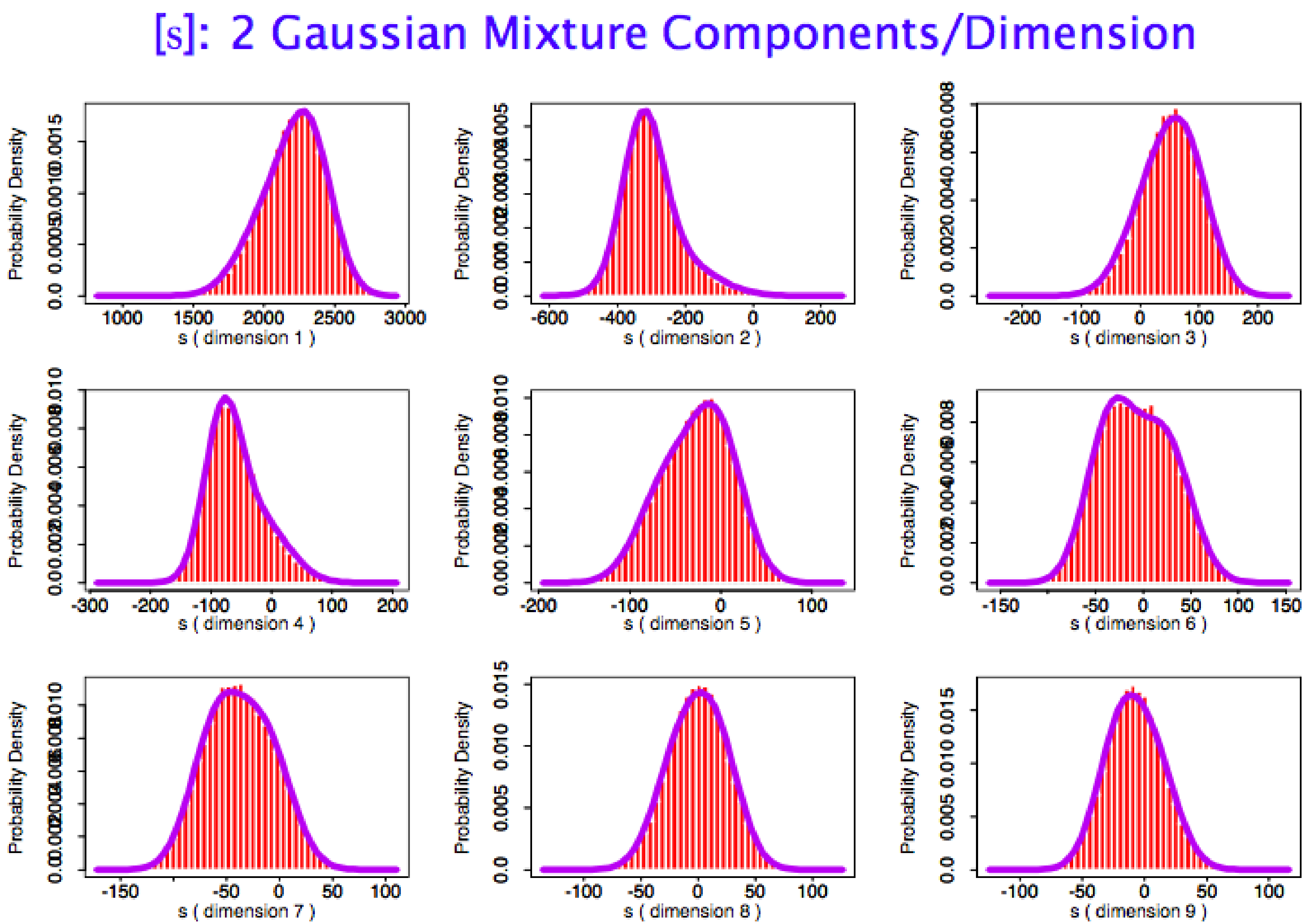
Gaussian Mixture Models (GMMs)

- Gaussians cannot handle multi-modal data well
- Consider a class can be further divided into additional factors
- Mixing weight makes sure the overall probability sums to 1

$$P(x) \sim \sum_{k=1}^{K} w_k N(\mu_k, \sigma_k)$$



Mixture of two Gaussians

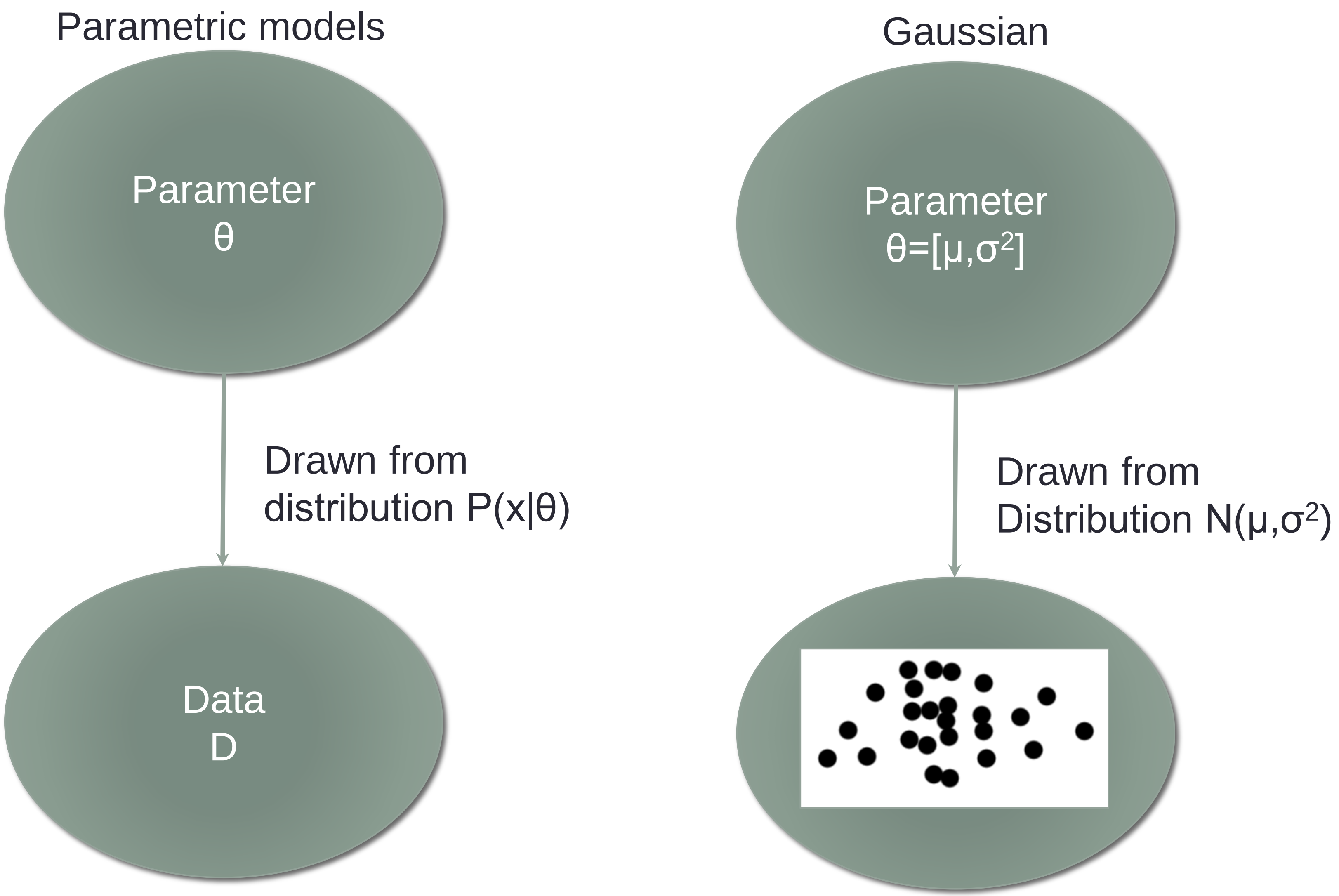


Mixture models

$$p(x) = \sum_k p(k) p_k(x)$$

- A mixture of models from the same distributions (but with different parameters)
- Different mixtures can come from different sub-class
  - Cat class
    - Siamese cats
    - Persian cats
- p(k) is usually categorical (discrete classes)
- Usually the exact class for a sample point is unknown.
  - Latent variable

Parametric models



Maximum A Posteriori (MAP) Estimate

MLE

- Maximizing the likelihood (probability of data given model parameters)

$$\underset{\theta}{\operatorname{argmax}}\ p(\mathbf{x}|\theta)$$

$p(\mathbf{x}|\theta)$
$= L(\theta)$

- Usually done on log likelihood

- Take the partial derivative wrt to θ and solve for the θ that maximizes the likelihood

MAP

- Maximizing the posterior (model parameters given data)

$$\underset{\theta}{\operatorname{argmax}}\ p(\theta|\mathbf{x})$$

- But we don't know $p(\theta|\mathbf{x})$

- Use Bayes rule

$$p(\theta|\mathbf{x}) = \frac{p(\mathbf{x}|\theta)p(\theta)}{p(\mathbf{x})}$$

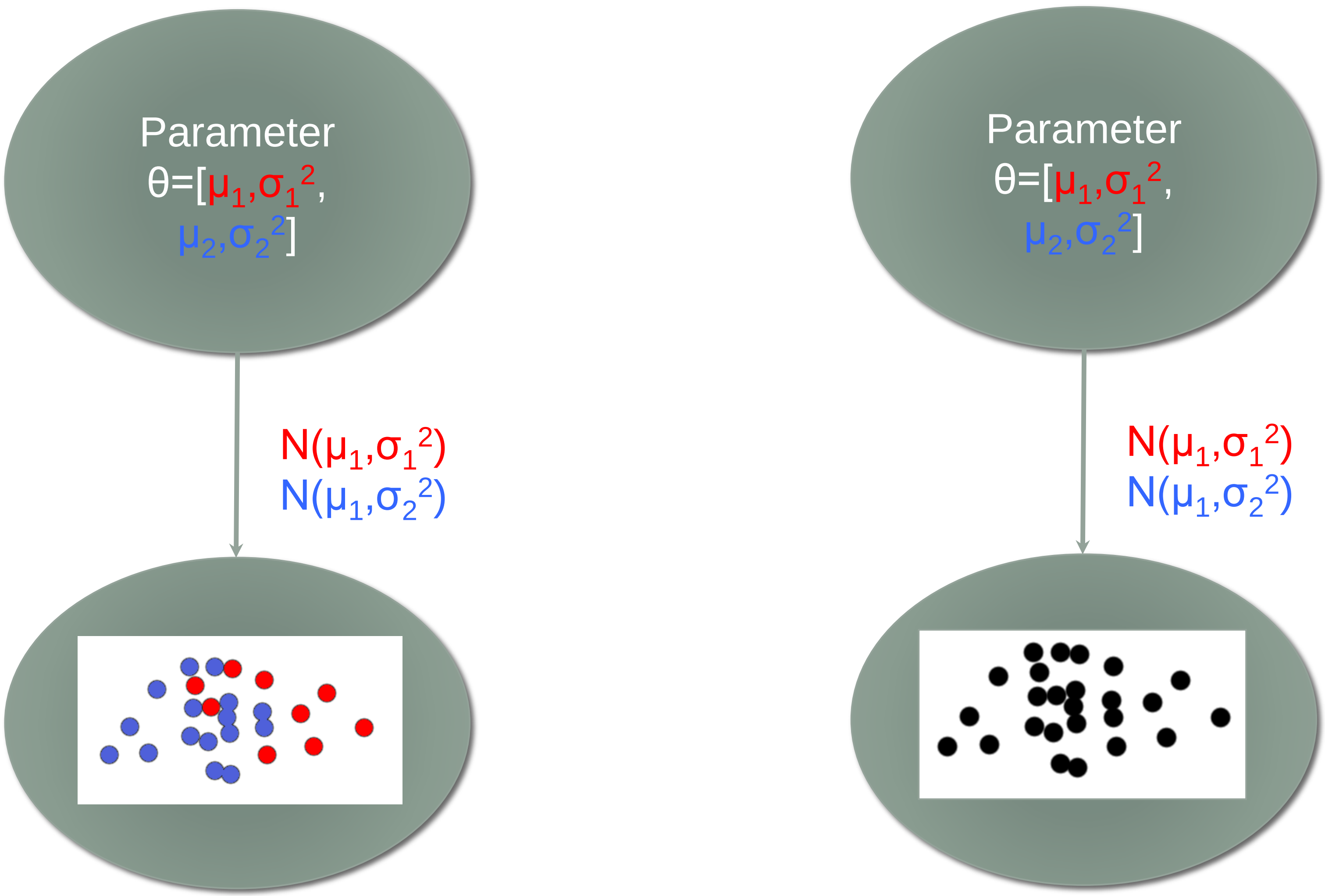
- Taking the argmax for θ we can ignore $p(\mathbf{x})$

- $\underset{\theta}{\operatorname{argmax}}\ p(\mathbf{x}|\theta)\ p(\theta)$

What if some data is missing?

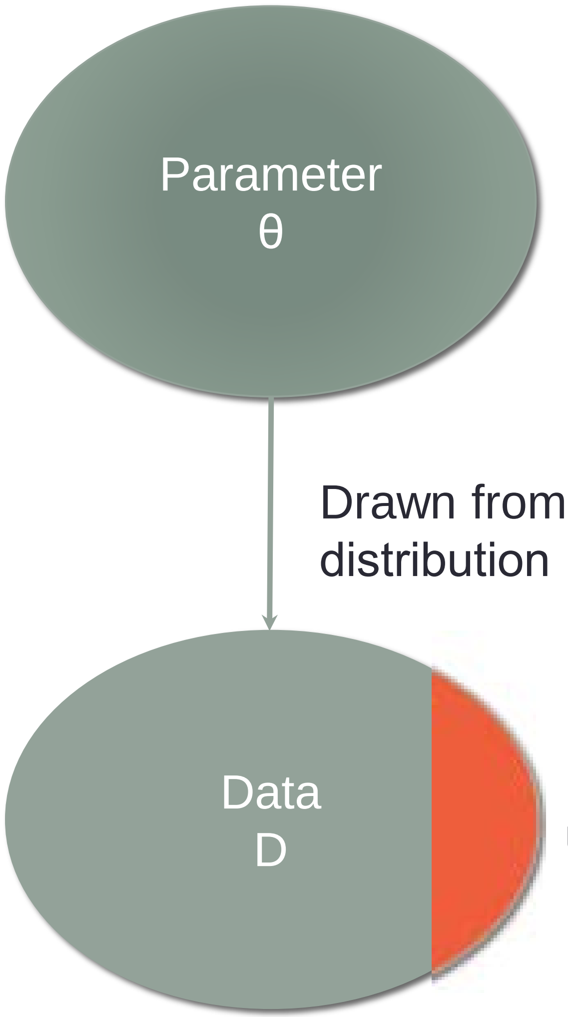
Mixture of Gaussian

Unknown mixture labels



Estimating missing data

Parametric models



Drawn from
distribution $P(x,k|\theta)$

Latent variables,k
unknown

Need to estimate both the latent
Variables and the model parameters.

Slight difference in notation

$p(\mathbf{x}|\theta)$ vs $p(\mathbf{x};\theta)$

θ as a RV at a fixed value vs θ as a fixed parameter

Most of the time can be used interchangeably

Estimating latent variables and model parameters

- GMM $$p(x) = \sum_k p(k) N(\mu_k, \sigma_k)$$
- Observed $(x_1, x_2, \ldots, x_N)$
- Latent $(k_1, k_2, \ldots, k_N)$ from K possible mixtures
- Parameter for p(k) is ϕ , $p(k = 1) = \phi_1$, $p(k = 2) = \phi_2 \ldots$

$$l(\phi, \mu, \Sigma) = \Sigma_{n=1}^{N} log p(x^{(i)}; \phi, \mu, \sigma)$$

$$= \Sigma_{n=1}^{N} log \boxed{\Sigma_{l=1}^{K}} p(x_n | k_{n,l}; \mu, \sigma) p(k_{n,l}; \phi)$$

Make things hard to solve

Cannot be solved by differentiating

Assuming k

- What if we somehow know k_n?
- Maximizing wrt to ϕ, μ, σ gives

$$\phi_j = \frac{1}{N} \Sigma_{n=1}^{N} 1(k_n = j)$$

$$\mu_j = \frac{\Sigma_{n=1}^{N} 1(k_n = j) x_n}{\Sigma_{n=1}^{N} 1(k_n = j)}$$

$$\sigma_j^2 = \frac{\Sigma_{n=1}^{N} 1(k_n = j)(x_n - \mu_j)^2}{\Sigma_{n=1}^{N} 1(k_n = j)}$$

$$1(condition)$$

Indicator function. Equals one if condition is met. Zero otherwise

Iterative algorithm

- Initialize ϕ, μ, σ
- Repeat till convergence
  - Expectation step (E-step) : Estimate the latent labels **k**
  - Maximization step (M-step) : Estimate the parameters ϕ, μ, σ given the latent labels
- Called Expectation Maximization (EM) Algorithm
- How to estimate the latent labels?

Iterative algorithm

- Initialize ϕ, μ, σ
- Repeat till convergence
  - Expectation step (E-step) : Estimate the latent labels **k** by finding the pdf of k given everything else $p(k|\ x\ ; \phi, \mu, \sigma)$
  - Maximization step (M-step) : Estimate the parameters ϕ, μ, σ given the latent labels by maximizing the expectation of the log likelihood
- Extension of MLE for latent variables
  - MLE : argmax $\log p(x;\theta)$
  - EM : argmax $\log \Sigma_k\ p(x, k;\theta)$

How to evaluate $\log \Sigma_k\ p(x, k;\theta)$ when we don't know k?

Convex functions and Jensen's inequality

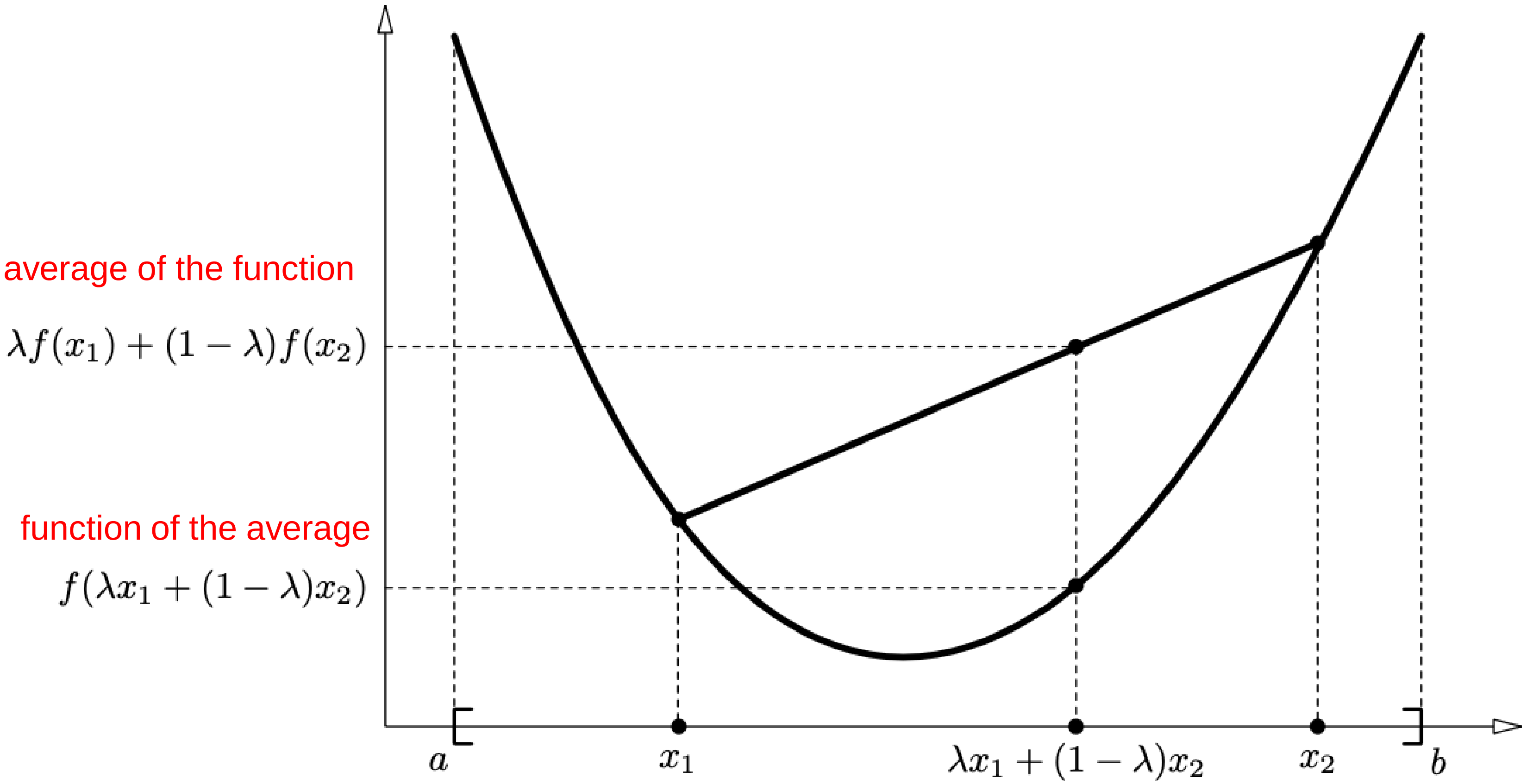


Figure 1: f is *convex* on $[a, b]$ if $f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2)$ $\forall x_1, x_2 \in [a, b], \quad \lambda \in [0, 1]$.

Jensen's inequality

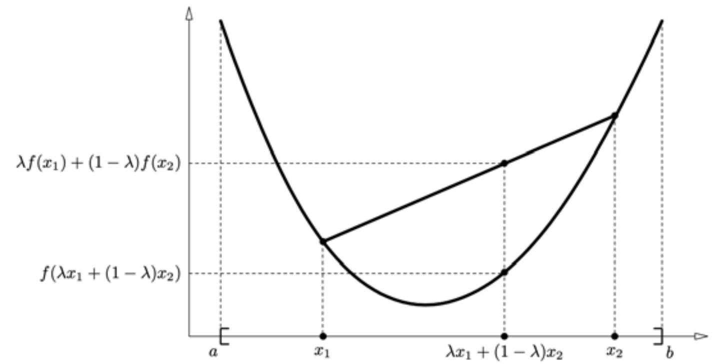


Figure 1: f is *convex* on $[a,b]$ if $f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2)$ $\forall x_1, x_2 \in [a,b], \quad \lambda \in [0,1]$.

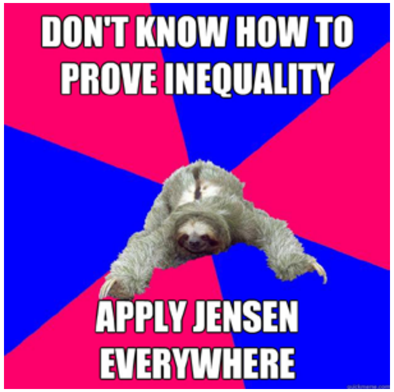
Let f be a convex function on interval I

If $x_1, x_2, \ldots, x_n$ is in I,

$w_1, \ldots, w_n > 0$ and sums to 1

then,

$$f\left(\sum_i^n w_i x_i\right) \leq \sum_i^n w_i f(x_i)$$



If f is concave, flip the inequality.

Can view this as expectation

$$f(E[X]) \leq E[f(X)]$$

Jensen's inequality and ELBO

$\log \Sigma_k \, p(x, k|\theta)$

$$f\left(\sum_i^n w_i x_i\right) \leq \sum_i^n w_i f(x_i)$$

Maximize Evidence Lower Bound (ELBO) = $\Sigma_k \, Q(k) \log (p(x, k;\theta)/Q(k))$

Making the lower bound tight

We will make the bound tight for fixed θ

Jensen's inequality is tight when?

$$f\left(\sum_i^n w_i x_i\right) \leq \sum_i^n w_i f(x_i)$$

$$f(E[X]) \leq E[f(X)]$$

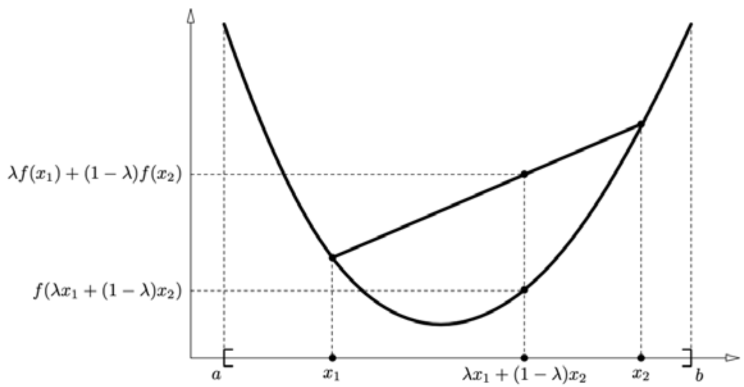


Figure 1: f is *convex* on $[a, b]$ if $f(\lambda x_1 + (1-\lambda) x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2)$ $\forall x_1, x_2 \in [a, b]$, $\lambda \in [0, 1]$.

Making the lower bound tight

We will make the bound tight for fixed θ

$$f\left(\sum_i^n w_i x_i\right) \leq \sum_i^n w_i f(x_i)$$

$$f(E[X]) \leq E[f(X)]$$

If f( ) is strictly convex, Jensen's inequality is tight IFF

x_i are all equal

E[X] = X = constant

Making the lower bound tight

We will make the bound tight for fixed θ

Jensen's inequality is tight when the inside of the expectation is a constant, c wrt the expectation

$p(x, k;\theta)/Q(k) = c$

or $Q(k) = p(k \mid x; \theta)$

Iterative algorithm (general)

- Goal of EM : $\log \sum_k p(x, k;\theta) >= \Sigma_k Q(k) \log (p(x, k;\theta)/Q(k))$
- Maximize the ELBO instead
- Initialize Θ
- Repeat till convergence
  - Expectation step (E-step) : estimate the conditional expectation $Q(k) = p(k|x;\theta)$ using the current θ.
  - Maximization step (M-step) : Estimate new Θ given by maximizing the ELBO given current $Q(k)$

EM on a simple example

- Grades in class $P(A) = 0.5$ $P(B) = 0.5-\theta$ $P(C) = \theta$
- We want to estimate θ from three known numbers
  - N_a N_b N_c
- Find the maximum likelihood estimate of θ

EM on a simple example

- Grades in class $P(A) = 0.5$ $P(B) = 0.5-\theta$ $P(C) = \theta$
- We want to estimate θ from ONE known number
  - N_c (we also know N the total number of students)
- Find θ using EM

Will this work?

For iteration i, with $\theta^{(i)}$

ELBO

$\log \sum_k p(x, k;\theta^{(i)}) >= \boxed{\Sigma_k Q(k) \log (p(x, k;\theta^{(i)})/Q(k))}$

E-step, making the bound tight by picking Q'(k) yields

$\log \sum_k p(x, k;\theta^{(i)}) = \Sigma_k Q'(k) \log (p(x, k;\theta^{(i)})/Q'(k))$

M-step, maximize ELBO by finding $\theta^{(i+1)}$

$\Sigma_k Q'(k) \log (p(x, k;\theta^{(i)})/Q'(k)) <= \Sigma_k Q'(k) \log (p(x, k;\theta^{(i+1)})/Q'(k))$

For iteration i+1, with $\theta^{(i+1)}$

$\log \sum_k p(x, k;\theta^{(i+1)}) >= \Sigma_k Q(k) \log (p(x, k;\theta^{(i+1)})/Q(k))$

Thus,

$\log \sum_k p(x, k;\theta^{(i+1)}) >= \log \sum_k p(x, k;\theta^{(i)})$

So EM improves the likelihood at every step!

Notes on ELBO

We set $Q(k) = p(k \mid x; \theta)$ to make the inequality tight.

What if we cannot compute $p(k \mid x; \theta)$?

Use a looser bound by picking any $Q(k)$

Estimate $p(k \mid x; \theta)$ with $q(k \mid x; \theta)$ that we can compute

This is called Variational Inference

We will revisit this.

Estimating latent variables and model parameters

- GMM $$p(x) = \sum_k p(k) N(\mu_k, \sigma_k)$$
- Observed $(x_1, x_2, \ldots, x_N)$
- Latent $(k_1, k_2, \ldots, k_N)$ from K possible mixtures
- Parameter for p(k) is ϕ , $p(k = 1) = \phi_1$, $p(k = 2) = \phi_2 \ldots$

$$l(\phi, \mu, \Sigma) = \Sigma_{n=1}^{N} log p(x^{(i)}; \phi, \mu, \sigma)$$

$$= \Sigma_{n=1}^{N} log \boxed{\Sigma_{l=1}^{K}} p(x_n | k_{n,l}; \mu, \sigma) p(k_{n,l}; \phi)$$

Make things hard to solve

Cannot be solved by differentiating

EM on GMM

- E-step
  - Set soft labels: $\boxed{w_{n,j}}$ = probability that nth sample comes from jth mixture p
  - Using Bayes rule
    - $\boxed{p(k|x\,;\,\mu,\sigma,\phi)} = \dfrac{p(x|k\,;\,\mu,\sigma,\phi)\,p(k;\,\mu,\sigma,\phi)}{p(x;\,\mu,\sigma,\phi)}$
    - $p(k|x\,;\,\mu,\sigma,\phi)$ is proportional to $p(x|k\,;\,\mu,\sigma,\phi)\,p(k;\,\phi)$

$$p(k_n = j | x_n; \phi, \mu, \Sigma) = \frac{\overbrace{p(x_n; \mu_j, \sigma_j)}^{N(\mu_j, \sigma_j)}\overbrace{p(k_n = j; \phi)}^{\phi_j}}{\Sigma_l p(x_n; \mu_l, \sigma_l) p(k_n = l; \phi)}$$

EM on GMM

- M-step (hard labels)

$$\phi_j = \frac{1}{N} \Sigma_{n=1}^{N} 1(k_n = j)$$

$$\mu_j = \frac{\Sigma_{n=1}^{N} 1(k_n = j) x_n}{\Sigma_{n=1}^{N} 1(k_n = j)}$$

$$\sigma_j^2 = \frac{\Sigma_{n=1}^{N} 1(k_n = j)(x_n - \mu_j)^2}{\Sigma_{n=1}^{N} 1(k_n = j)}$$

EM on GMM

- M-step (soft labels)

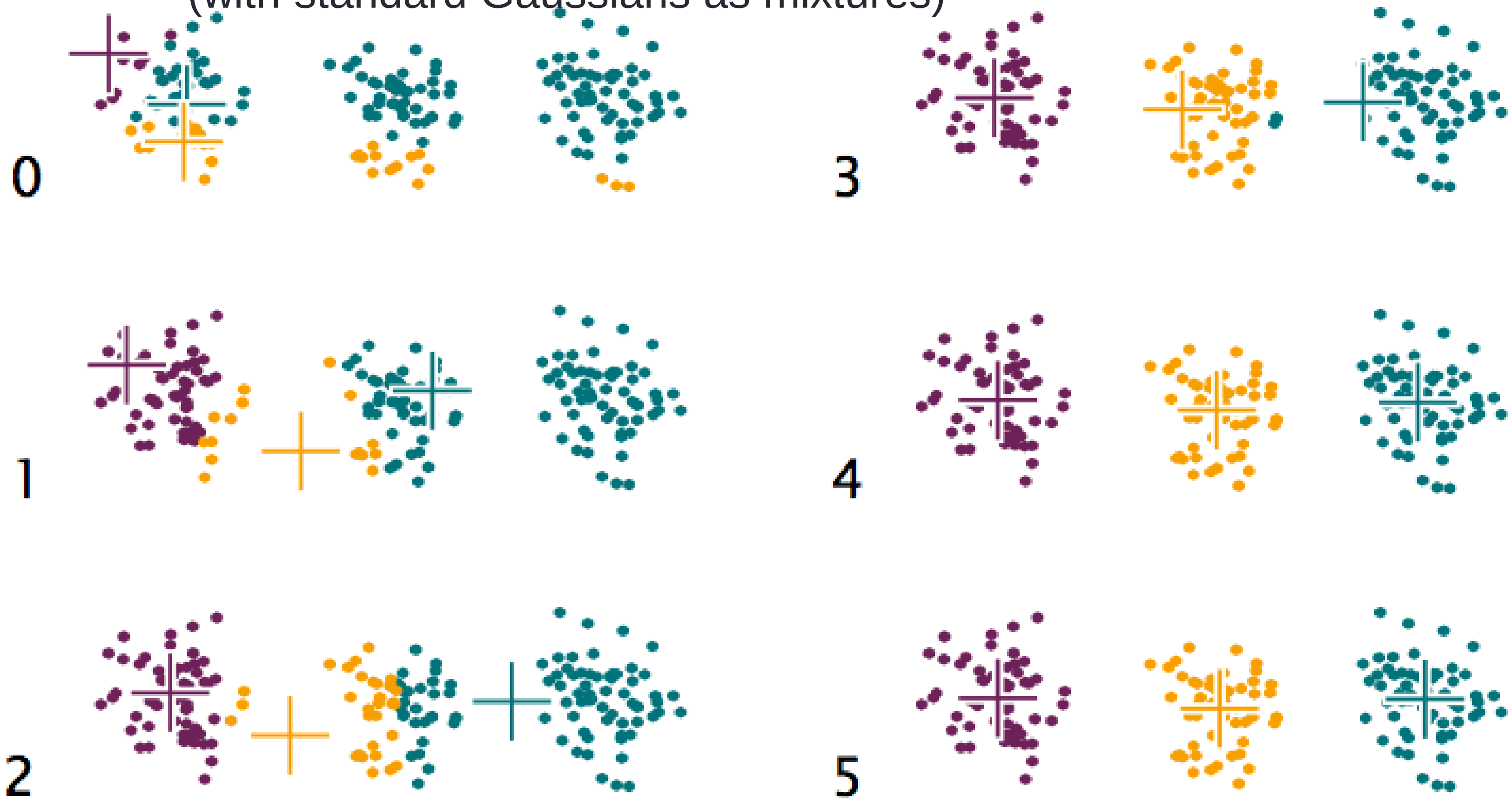
$$\phi_j = \frac{1}{N} \Sigma_{n=1}^{N} w_{n,j}$$

$$\mu_j = \frac{\Sigma_{n=1}^{N} w_{n,j} x_n}{\Sigma_{n=1}^{N} w_{n,j}}$$

$$\sigma_j^2 = \frac{\Sigma_{n=1}^{N} w_{n,j} (x_n - \mu_j)^2}{\Sigma_{n=1}^{N} w_{n,j}}$$

K-mean vs EM

EM on GMM can be considered as EM with soft labels
(with standard Gaussians as mixtures)



K-mean clustering

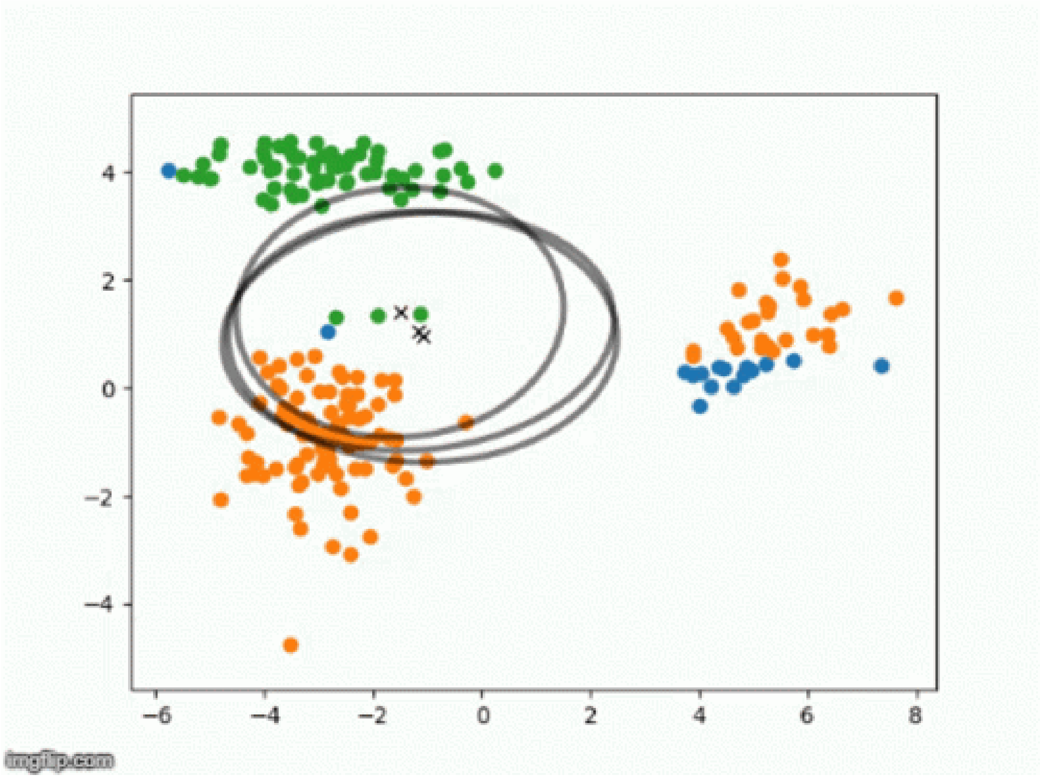
- Task: cluster data into groups
- K-mean algorithm
  - Initialization: Pick K data points as cluster centers
  - Assign: Assign data points to the closest centers
  - Update: Re-compute cluster center
  - Repeat: Assign and Update

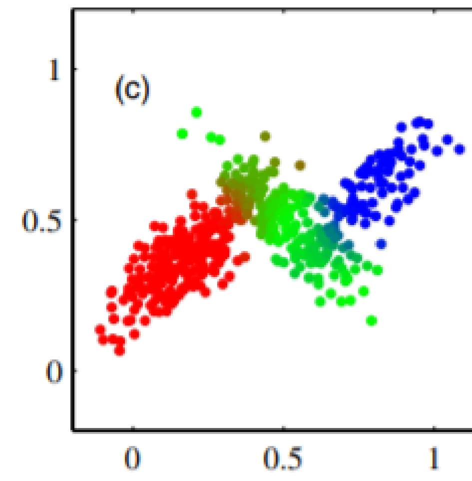
EM algorithm for GMM

- Task: cluster data into Gaussians
- EM algorithm
  - Initialization: Randomly initialize parameters Gaussians
  - Expectation: Assign data points to the closest Gaussians
  - Maximization: Re-compute Gaussians parameters according to assigned data points
  - Repeat: Expectation and Maximization

- Note: assigning data points is actually a soft assignment (with probability)

K-mean vs EM

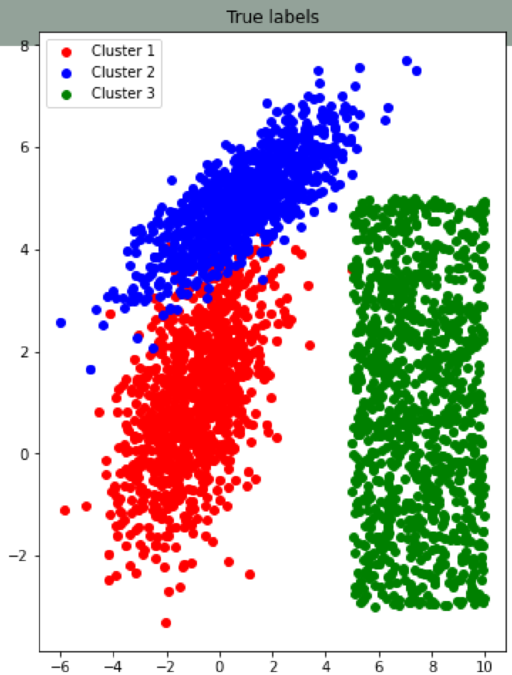


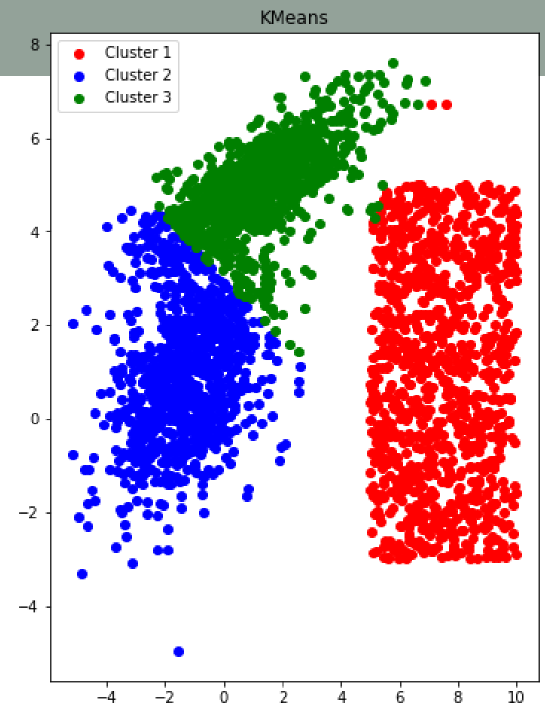


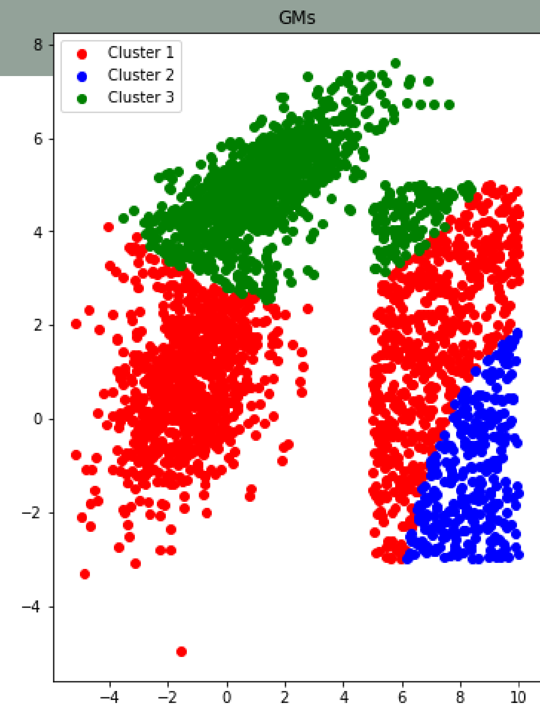
https://towardsdatascience.com/gaussian-mixture-models-vs-k-means-which-one-to-choose-62f2736025f0

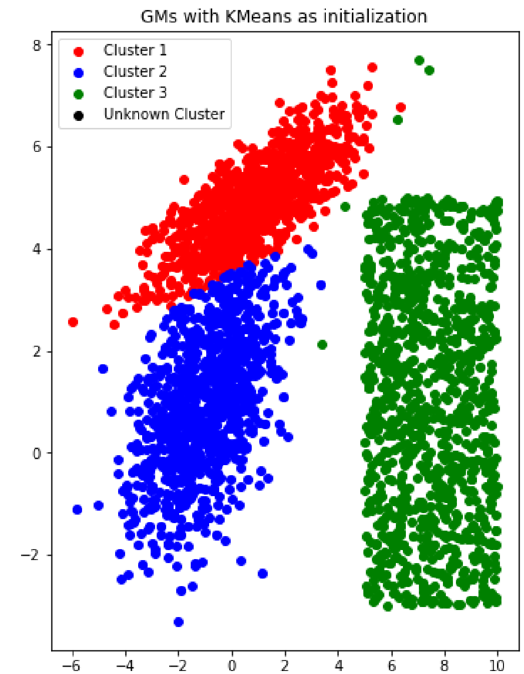
EM/GMM notes

- Converges to local maxima (maximizing likelihood)
  - Just like k-means, need to try different initialization points
- EM always improve the likelihood for each iteration
  - Stops EM when likelihood changes < threshold
- Just like k-means some centroid can get stuck with one sample point and no longer moves
  - For EM on GMM this cause variance to go to 0…
    - Introduce variance floor (minimum variance a Gaussian can have)
- Tricks to avoid bad local maxima
  - Starts with 1 Gaussian
  - Split the Gaussians according to the direction of maximum variance
  - Repeat until arrive at k Gaussians
  - Does not guarantee global maxima but works well in practice



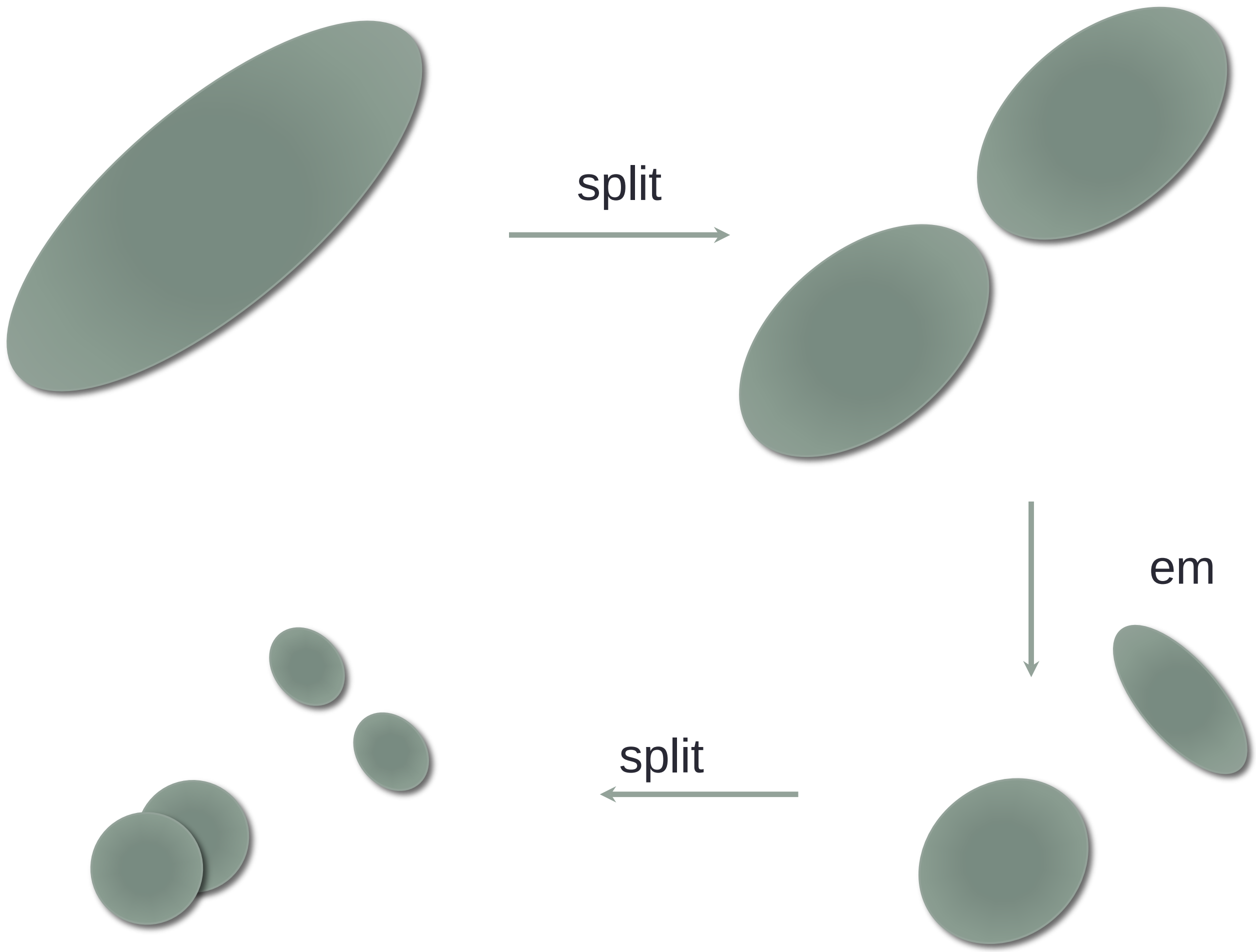






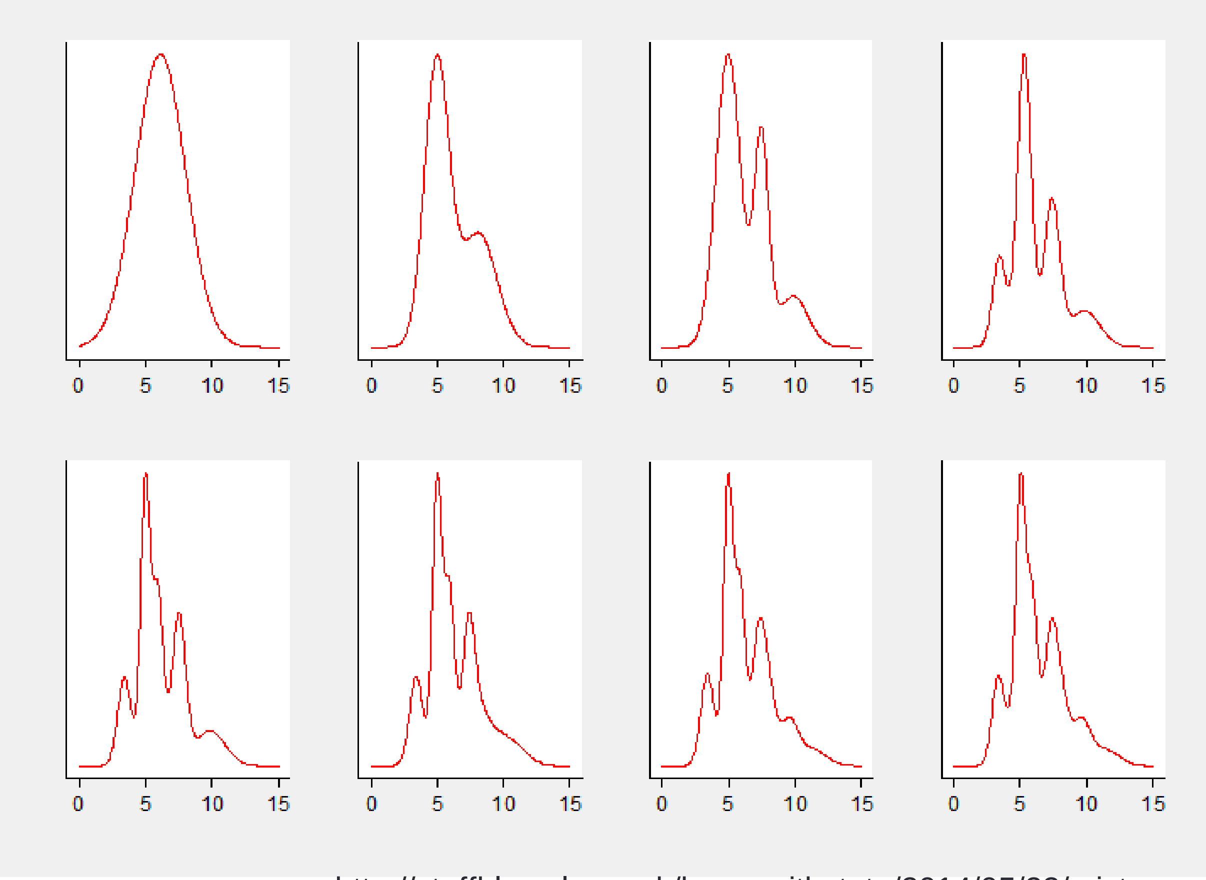
https://towardsdatascience.com/gaussian-mixture-models-vs-k-means-which-one-to-choose-62f2736025f0

Gaussian splitting



Picking the amount of Gaussians

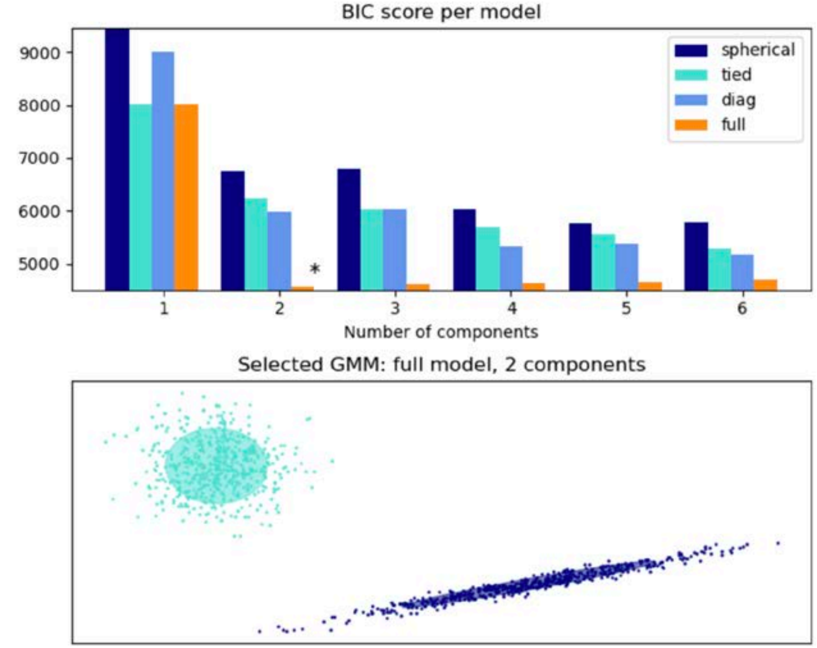
- As we increase K, the likelihood will keep increasing
- More mixtures -> more parameters -> overfits



http://staffblogs.le.ac.uk/bayeswithstata/2014/05/22/mixture-models-how-many-components/

Picking the amount of Gaussians

- Need a measure of goodness (like Elbow method in k-mean)
- Bayesian Information Criterion (BIC)
- Penalize the log likelihood from the data by the number of parameters in the model
  - $-2 \log L + t \log (n)$
  - t = number of parameters in the model
  - n = number of data points
- We want to minimize BIC



https://scikit-learn.org/stable/auto_examples/mixture/plot_gmm_selection.html

BIC is bad use cross validation!

- Just like how I don’t recommend using elbow method for clustering
- BIC is bad use cross validation!
- Test on the goal of your model

Latent variables?

EM is all about problem formulation. You can solve the same task with different formulations.

Latent variable considerations

- Imaginary quantity meant to provide a simplified view of the process
  - GMM mixtures. Speech recognizer states. Customer segmentation.
- Real-world thing, but impossible to directly measure
  - Cause of a disease. Temperature of a star.
- Real-world thing, that is not measured because of noise/faulty sensors

Latent variables?

- Discrete latent variables: clusters/partitions data into subgroups
- Continuous latent variables: can be used for dimensionality reduction (factor analysis, etc)

EM usage examples

Image segmentation with GMM EM

- D - {r,g,b} value at each pixel
- Latent : segment where each pixel comes from
- Hyperparameters: number of mixtures (K), initial values

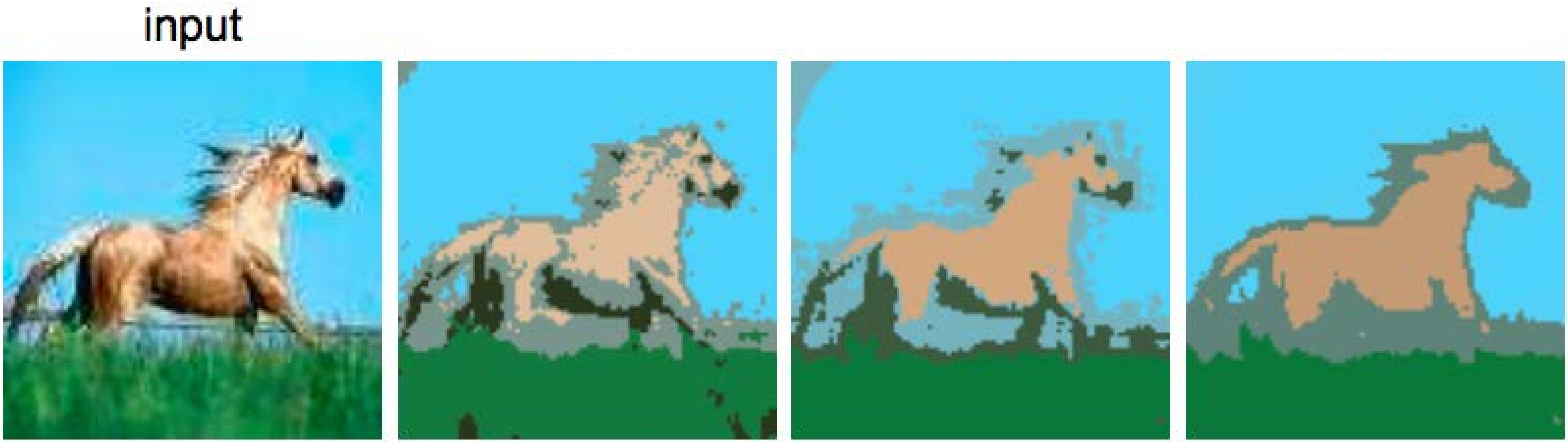


Image segmentation with GMM EM

(a) (b) (c)

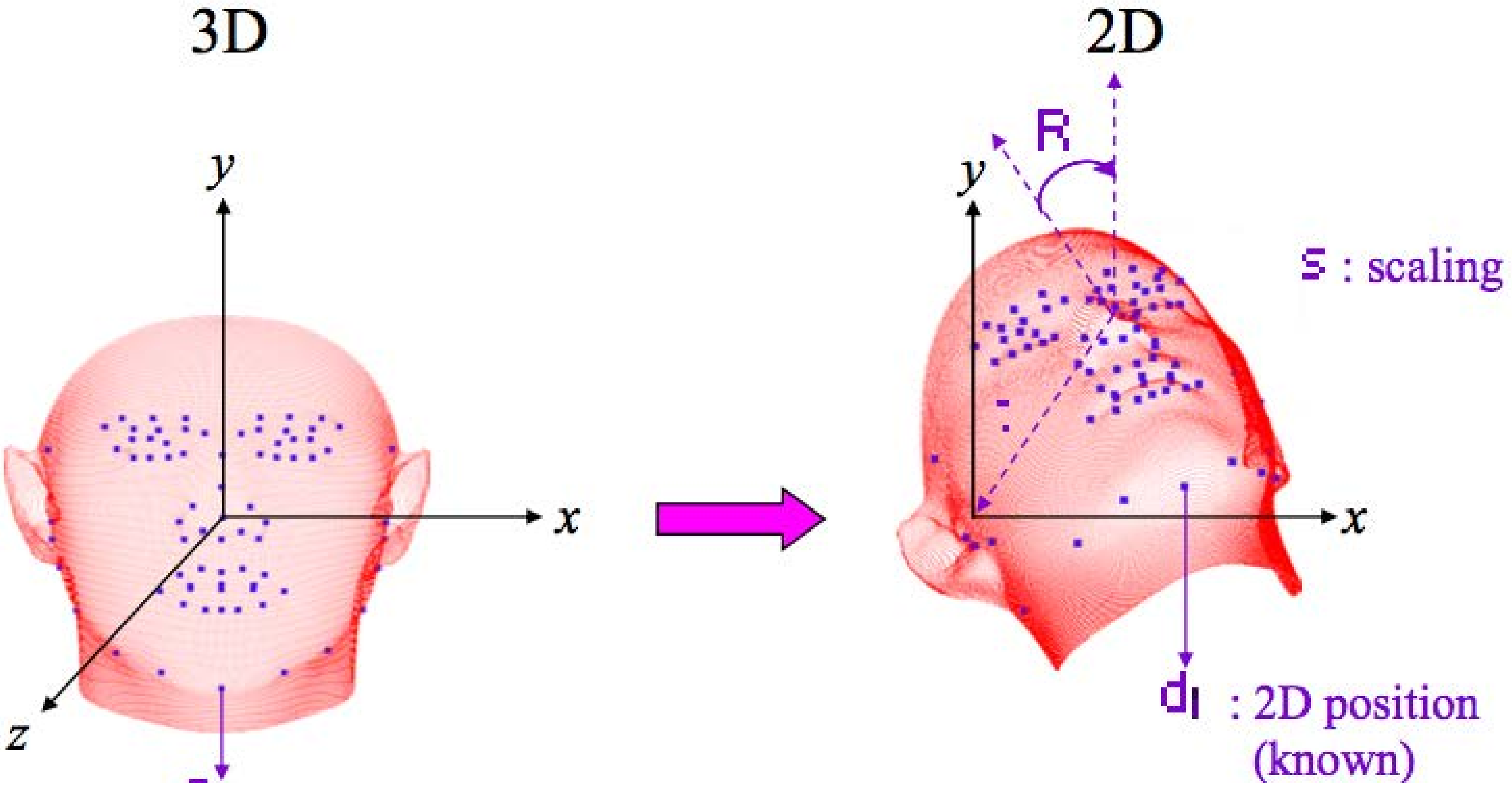
Fig. 1. Original images: (a) flower, (b) tiger, (c) bear

Fig. 2. Segmentation results ($M = 2$)

Fig. 3. Segmentation results ($M = 5$)

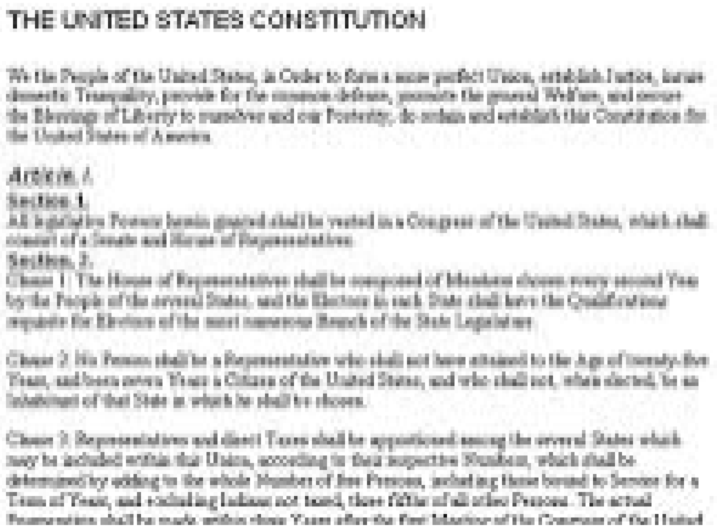
Zhaoxia Fu Color Image Segmentation Using Gaussian Mixture Model and EM Algorithm

Face pose estimation (estimate 3d coordinates from 2d picture)



https://www.researchgate.net/publication/2489713_Estimating_3D_Facial_Pose_using_the_EM_Algorithm

Language modeling


THE UNITED STATES CONSTITUTION

Article I.

Section 1.

Section 2.

Latent variable:
Topic
P(word|topic)

For examples: see Probabilistic latent semantic analysis

MEME

Multiple EM for Motif Elicitation

From Wikipedia, the free encyclopedia

For other uses, see MEME (disambiguation).

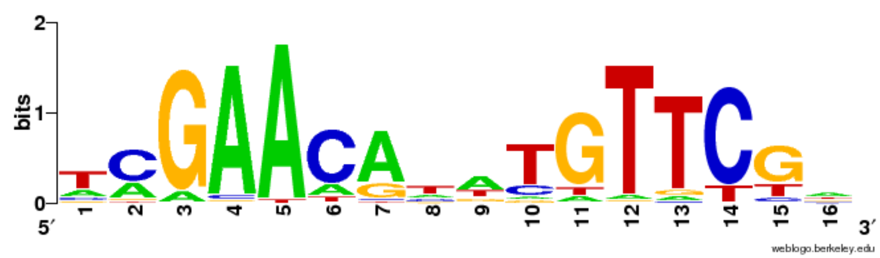
Multiple Expectation maximizations for Motif Elicitation (MEME) is a tool for discovering motifs in a group of related DNA or protein sequences.[1]

A motif is a sequence pattern that occurs repeatedly in a group of related protein or DNA sequences and is often associated with some biological function. MEME represents motifs as position-dependent letter-probability matrices which describe the probability of each possible letter at each position in the pattern. Individual MEME motifs do not contain gaps. Patterns with variable-length gaps are split by MEME into two or more separate motifs.

MEME takes as input a group of DNA or protein sequences (the training set) and outputs as many motifs as requested. It uses statistical modeling techniques to automatically choose the best width, number of occurrences, and description for each motif.

MEME is the first of a collection of tools for analyzing motifs called the MEME suite.

Contents [hide]



https://en.wikipedia.org/wiki/Multiple_EM_for_Motif_Elicitation
https://en.wikipedia.org/wiki/Position_weight_matrix

Summary

- GMM
  - Mixture of Gaussians
- EM
  - Expectation
  - Maximization

More info and exact proofs

https://www.cs.utah.edu/~piyush/teaching/EM_algorithm.pdf

http://cs229.stanford.edu/summer2019/cs229-notes8.pdf

Homework notes

- T8
- RoC

Prediction and thresholds

Beer	Grass	Rice	Flood	Prediction
100	3	3	**Yes**	**0.8**
20	1	1	**Yes**	**0.3**
80	3	2	**No**	**0.6**
40	1	1	**No**	**0.2**
40	1	1	**No**	**0.1**

What happens if I set my threshold at 0.5?

Prediction and thresholds

Beer	Grass	Rice	Flood	Prediction	Metric
100	3	3	**Yes**	**0.8**	**TP**
20	1	1	**Yes**	**0.3**	**FN**
80	3	2	**No**	**0.6**	**FA**
40	1	1	**No**	**0.2**	**TN**
40	1	1	**No**	**0.1**	**TN**

What happens if I set my threshold at 0.5?

True positive rate =

False alarm rate =

Precision =

Recall =

Prediction and thresholds

Beer	Grass	Rice	Flood	Prediction	Metric
100	3	3	**Yes**	**0.8**	**TP**
20	1	1	**Yes**	**0.3**	**FN**
80	3	2	**No**	**0.6**	**FA**
40	1	1	**No**	**0.2**	**TN**
40	1	1	**No**	**0.1**	**TN**

What happens if I set my threshold at 0.5?

True positive rate = ½

False alarm rate = ⅓

Precision = ½

Recall = ½

Prediction and thresholds

Beer	Grass	Rice	Flood	Prediction	Metric
100	3	3	**Yes**	**0.8**	
20	1	1	**Yes**	**0.3**	
80	3	2	**No**	**0.6**	
40	1	1	**No**	**0.2**	
40	1	1	**No**	**0.1**	

What happens if I set my threshold at 0.15?

Prediction and thresholds

Beer	Grass	Rice	Flood	Prediction	Metric
100	3	3	**Yes**	**0.8**	**TP**
20	1	1	**Yes**	**0.3**	**TP**
80	3	2	**No**	**0.6**	**FA**
40	1	1	**No**	**0.2**	**FA**
40	1	1	**No**	**0.1**	**TN**

What happens if I set my threshold at 0.15?

True positive rate = 1

False alarm rate = 2/3

Precision = 2/4

Recall = 2/2

Prediction and thresholds

Beer	Grass	Rice	Flood	Prediction	Metric
100	3	3	**Yes**	**0.8**	
20	1	1	**Yes**	**0.3**	
80	3	2	**No**	**0.6**	
40	1	1	**No**	**0.2**	
40	1	1	**No**	**0.1**	

What happens if I set my threshold at 0.5?

- True positive rate = ½
- False alarm rate = ⅓
- Precision = ½
- Recall = ½

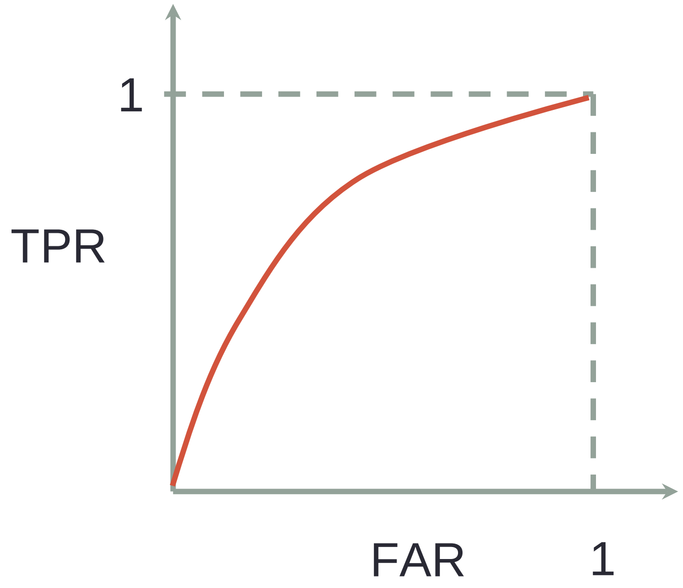
What happens if I set my threshold at 0.15?

- True positive rate = 1
- False alarm rate = 2/3
- Precision = 2/4
- Recall = 2/2

Receiver operating Characteristic (RoC) curve

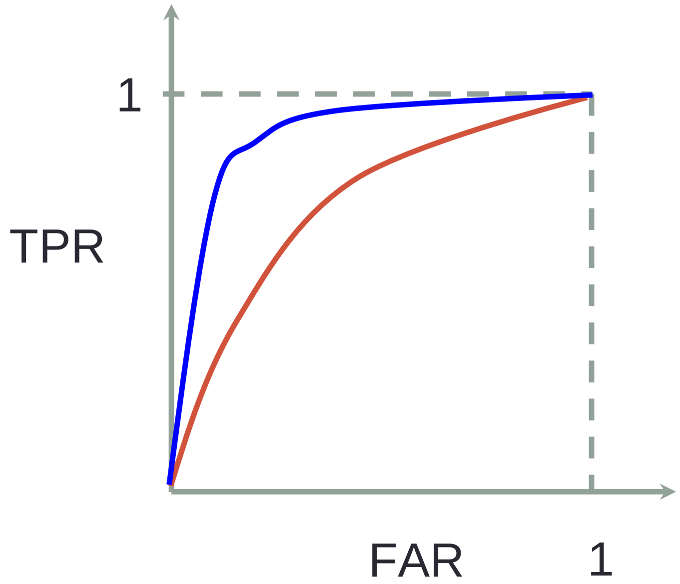
- What if we change the threshold
- FA TP is a tradeoff

  This is why we need to think of the application when thinking of metrics.
- Plot FA rate and TP rate as threshold changes



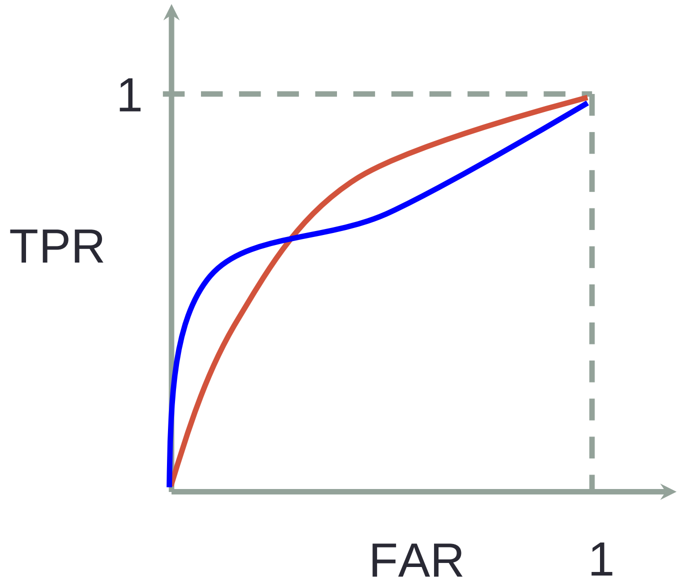
Comparing detectors

- Which is better?



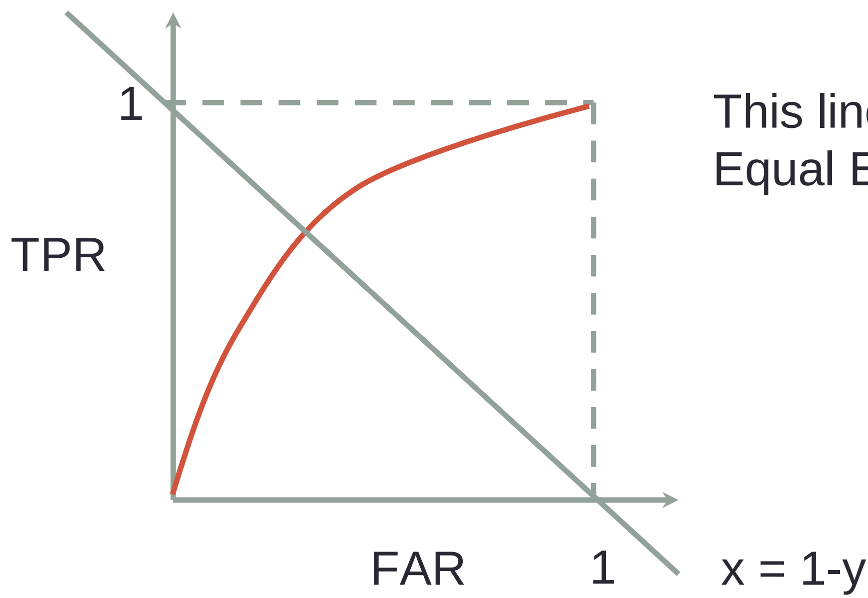
Comparing detectors

- Which is better?



Selecting the threshold

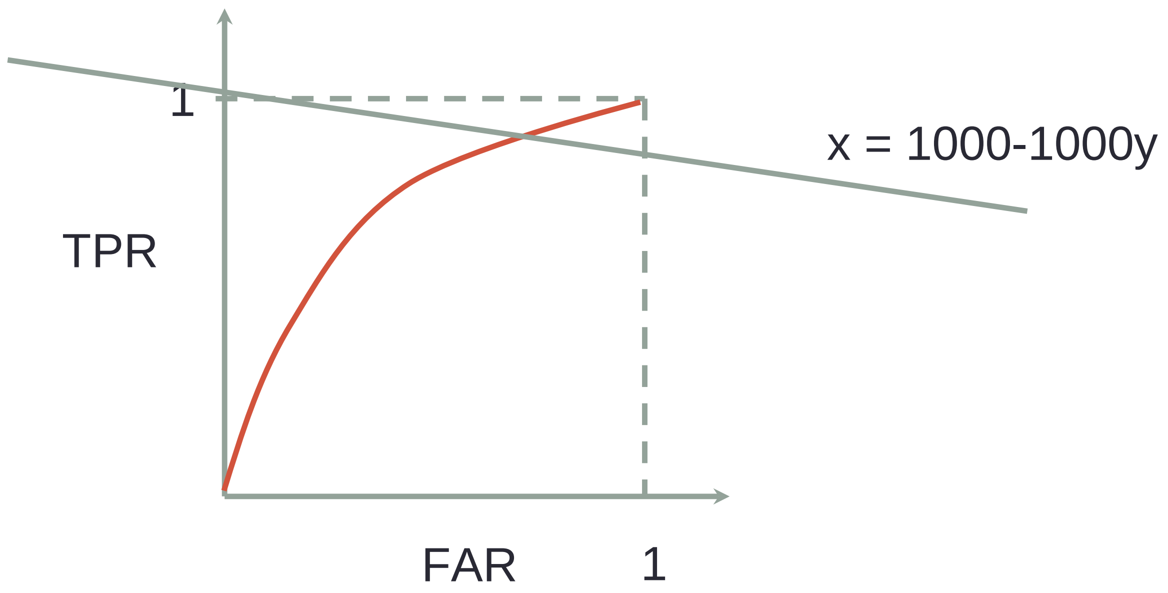
- Select based on the application
- Trade off between TP and FA. Know your application, know your users.
  - A miss is as bad as a false alarm FAR = 1-TPR => x = 1-y



This line has a special name
Equal Error Rate (EER)

Selecting the threshold

- Select based on the application
- Trade off between TP and FA. Know your application, know your users. Is the application about safety?
  - A miss is 1000 times more costly than false alarm.
    - FAR = 1000(1-TPR) => x = 1000-1000y



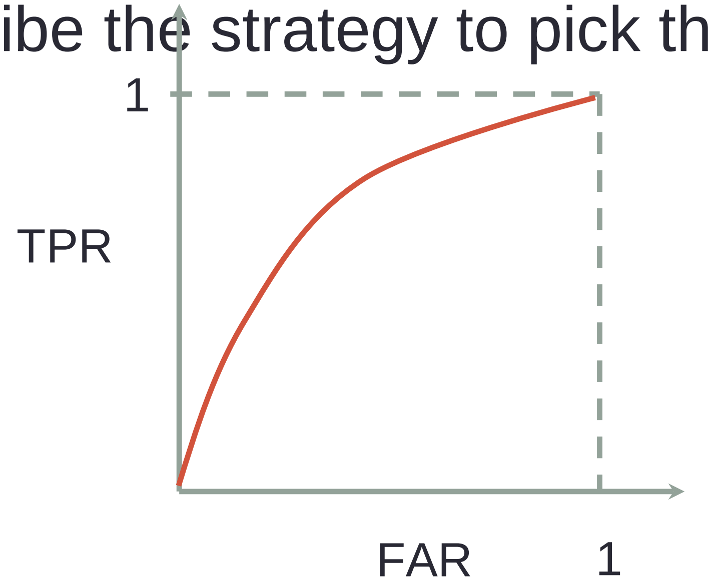
Churn prediction

Predict whether a customer will stop subscription, so we can send a promotional ad.

Usual subscription fee 50

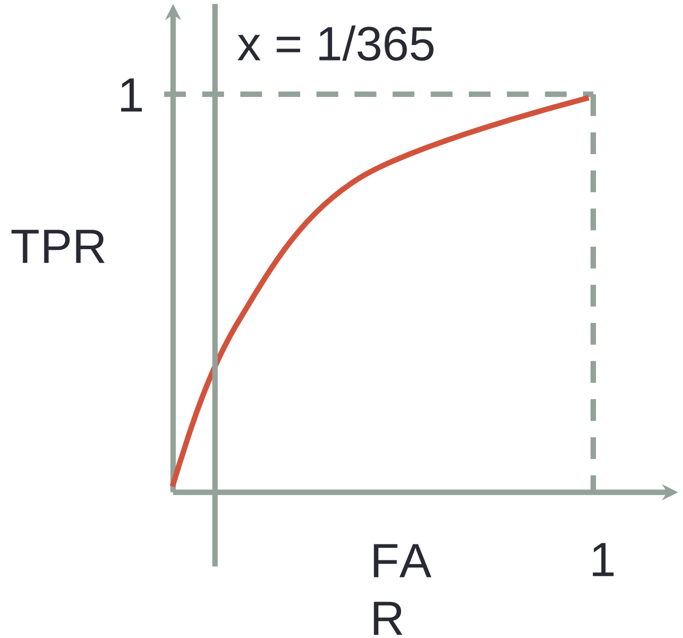
Cost of calling the customer 5

- Promotional subscription fee 25
  - Describe the strategy to pick the threshold



Selecting the threshold

- Select based on the application
- Trade off between TP and FA.
  - Regulation or hard threshold
  - Cannot exceed 1 False alarm per year
    - If 1 decision is made everyday, FAR = 1/365



Notes about RoC

- Ways to compress RoC to just a number for easier comparison -- use with care!!
  - EER
  - Area under the curve
  - F score
- Other similar curve - Detection Error Tradeoff (DET) curve
  - Plot False alarm vs Miss rate
- Can plot on log scale for clarity