Machine Learning

Gaussian Mixture Models

Hiroshi Shimodaira

March 2026

Ver. 1.0.1

Adapted from Kia Nazarpour's slides

Recap: K-means clustering

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2 \tag{1}$$

$$\boldsymbol{\mu}_k = \frac{\sum_n r_{nk}\mathbf{x}_n}{\sum_n r_{nk}} \tag{2}$$

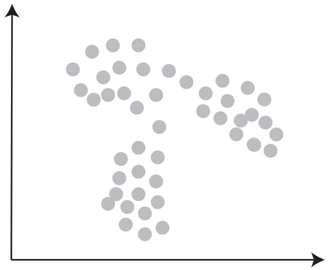
$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_{k'} \|\mathbf{x}_n - \boldsymbol{\mu}_{k'}\|^2 \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Recap: K-means clustering *(cont.)*

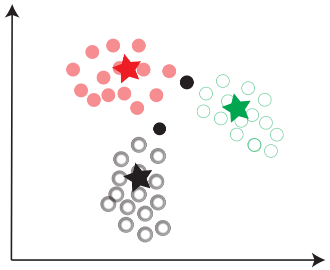
1. Too crude? Assumes that a cluster can be represented with a single point and a simple distance metric
2. A simple unsupervised method that enables clustering of data with no great computational complexity
3. Hard boundaries
4. Q: How to generalise it to models that can cluster data of various types and shapes?

Hard assignment vs. Soft assignment

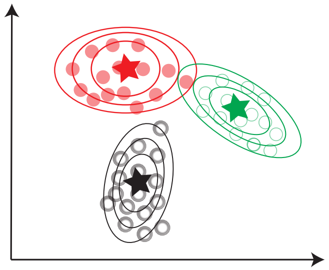
Original Data



Hard assignment



Soft assignment



Gaussian Mixture Model

Learning Outcomes

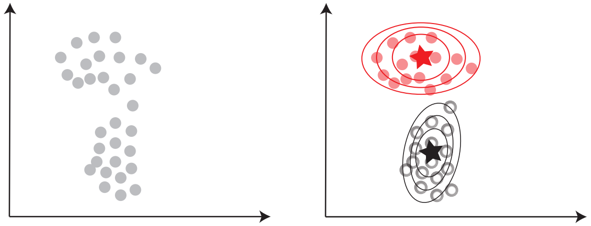
1. Understand the key motivations behind a Gaussian Mixture Model (GMM).
2. Understand the formulation of a GMM and the need for the Expectation Maximisation (EM) solver.
3. Analyse the solution to a GMM.

References:

1. [B] Bishop, *Pattern Recognition and Machine Learning*, Springer, 2008. (Section 9.1)
2. [DFO], (Chapter 11)
3. Rogers and Girolami, *A First Course in Machine Learning*, CRC Press, 2016. (Section 6.3)

Mixture Models

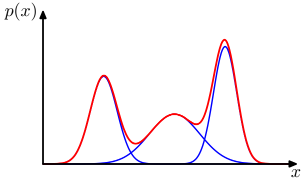
1. Models that can cluster data of various types and shapes
2. Simple to compute
3. Clustering with statistical mixture models, similar to k-means, but offers richer representation of the data



Mixture of Gaussians (Gaussian mixture model) - GMM

Complex probabilistic distributions can be approximated with a linear superposition of K Gaussian densities.

$$p(\mathbf{x} \,|\, \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(\mathbf{x} \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{4}$$



where π_k: mixing weight, $0 \leq \pi_k \leq 1$, and $\sum_{k=1}^{K} \pi_k = 1$

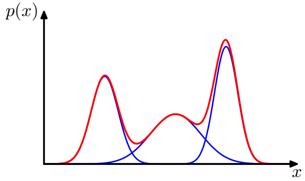
We introduce binary variables $\mathbf{z} = \begin{bmatrix} z_1 & \ldots & z_K \end{bmatrix}^\top$, where $z_k \in \{0, 1\}$ and $\sum_k z_k = 1$, such that:

- $p(z_k = 1) = \pi_k$

Mixture of Gaussians (Gaussian mixture model) - GMM

Complex probabilistic distributions can be approximated with a linear superposition of K Gaussian densities.

$$p(\mathbf{x} \,|\, \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(\mathbf{x} \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{4}$$



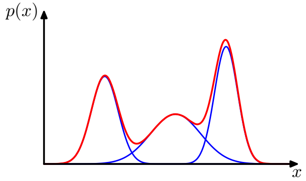
where π_k: mixing weight, $0 \leq \pi_k \leq 1$, and $\sum_{k=1}^{K} \pi_k = 1$

We introduce binary variables $\mathbf{z} = \begin{bmatrix} z_1 & \dots & z_K \end{bmatrix}^\top$, where $z_k \in \{0, 1\}$ and $\sum_k z_k = 1$, such that:

- $p(z_k = 1) = \pi_k$
- $p(\boldsymbol{z}) = \boldsymbol{\pi} = \begin{bmatrix} \pi_1 & \dots & \pi_K \end{bmatrix}^\top$

Mixture of Gaussians (Gaussian mixture model) - GMM

Complex probabilistic distributions can be approximated with a linear superposition of K Gaussian densities.



$$p(\mathbf{x} \,|\, \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(\mathbf{x} \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{4}$$

where π_k: mixing weight, $0 \leq \pi_k \leq 1$, and $\sum_{k=1}^{K} \pi_k = 1$

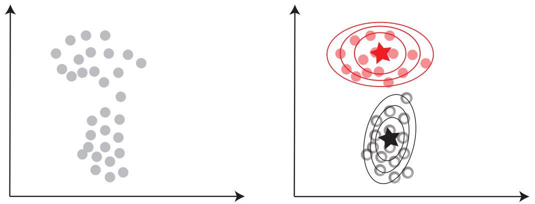
We introduce binary variables $\mathbf{z} = \begin{bmatrix} z_1 & \dots & z_K \end{bmatrix}^\top$, where $z_k \in \{0, 1\}$ and $\sum_k z_k = 1$, such that:

- $p(z_k = 1) = \pi_k$
- $p(\boldsymbol{z}) = \boldsymbol{\pi} = \begin{bmatrix} \pi_1 & \dots & \pi_K \end{bmatrix}^\top$
- $p(\boldsymbol{x} \,|\, z_k = 1, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{x} \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$

GMM - as a generative model

1. Let's assume we want to generate the below data with two Gaussians.
2. Select one of the Gaussians (based on $\boldsymbol{\pi}$) and set the indicator variable $z_k = 1$ (or label $y = k$) if Gaussian k is chosen.
3. Sample (generate) data $\boldsymbol{x}$ from this Gaussian k, and repeat from step 2 until you get N samples.

$$p(\boldsymbol{x} \mid z_k = 1, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{5}$$



GMM as a model with hidden variables

GMM as a model with *hidden (latent) variables* $\boldsymbol{z} = \begin{bmatrix} z_1 \; \ldots \; z_K \end{bmatrix}^\top$.

$$p(\boldsymbol{x} \mid \boldsymbol{\theta}) = \sum_{\boldsymbol{z}} p(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{\theta}) = \sum_{\boldsymbol{z}} p(\boldsymbol{x} \mid \boldsymbol{z}, \boldsymbol{\theta}) \, p(\boldsymbol{z} \mid \boldsymbol{\theta}) \tag{6}$$

GMM as a model with hidden variables

GMM as a model with *hidden (latent) variables* $\boldsymbol{z} = \begin{bmatrix} z_1 & \dots & z_K \end{bmatrix}^\top$.

$$\begin{aligned} p(\boldsymbol{x} \mid \boldsymbol{\theta}) &= \sum_{\boldsymbol{z}} p(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{\theta}) = \sum_{\boldsymbol{z}} p(\boldsymbol{x} \mid \boldsymbol{z}, \boldsymbol{\theta}) \, p(\boldsymbol{z} \mid \boldsymbol{\theta}) && (6) \\ &= \sum_{z_1=0}^{1} \cdots \sum_{z_K=0}^{1} p(\boldsymbol{x} \mid z_1, \dots, z_K, \boldsymbol{\theta}) \, p(z_1, \dots, z_K \mid \boldsymbol{\theta}) && (7) \end{aligned}$$

GMM as a model with hidden variables

GMM as a model with *hidden (latent) variables* $\boldsymbol{z} = \begin{bmatrix} z_1 & \dots & z_K \end{bmatrix}^\top$.

$$p(\boldsymbol{x} \,|\, \boldsymbol{\theta}) = \sum_{\boldsymbol{z}} p(\boldsymbol{x}, \boldsymbol{z} \,|\, \boldsymbol{\theta}) = \sum_{\boldsymbol{z}} p(\boldsymbol{x} \,|\, \boldsymbol{z}, \boldsymbol{\theta})\, p(\boldsymbol{z} \,|\, \boldsymbol{\theta}) \tag{6}$$

$$= \sum_{z_1=0}^{1} \cdots \sum_{z_K=0}^{1} p(\boldsymbol{x} \,|\, z_1, \dots, z_K, \boldsymbol{\theta})\, p(z_1, \dots, z_K \,|\, \boldsymbol{\theta}) \tag{7}$$

$$= p(\boldsymbol{x} \,|\, z_1 = 1, z_{i \neq 1} = 0, \boldsymbol{\theta})\, p(z_1 = 1, z_{i \neq 1} = 0 \,|\, \boldsymbol{\theta}) + \cdots$$
$$+ p(\boldsymbol{x} \,|\, z_K = 1, z_{i \neq K} = 0, \boldsymbol{\theta})\, p(z_K = 1, z_{i \neq K} = 0 \,|\, \boldsymbol{\theta}) \tag{8}$$

GMM as a model with hidden variables

GMM as a model with *hidden (latent) variables* $\boldsymbol{z} = \begin{bmatrix} z_1 & \dots & z_K \end{bmatrix}^\top$.

$$
\begin{aligned}
p(\boldsymbol{x} \mid \boldsymbol{\theta}) &= \sum_{\boldsymbol{z}} p(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{\theta}) = \sum_{\boldsymbol{z}} p(\boldsymbol{x} \mid \boldsymbol{z}, \boldsymbol{\theta})\, p(\boldsymbol{z} \mid \boldsymbol{\theta}) && (6) \\
&= \sum_{z_1=0}^{1} \cdots \sum_{z_K=0}^{1} p(\boldsymbol{x} \mid z_1, \dots, z_K, \boldsymbol{\theta})\, p(z_1, \dots, z_K \mid \boldsymbol{\theta}) && (7) \\
&= p(\boldsymbol{x} \mid z_1 = 1, z_{i \neq 1} = 0, \boldsymbol{\theta})\, p(z_1 = 1, z_{i \neq 1} = 0 \mid \boldsymbol{\theta}) + \cdots \\
&\quad + p(\boldsymbol{x} \mid z_K = 1, z_{i \neq K} = 0, \boldsymbol{\theta})\, p(z_K = 1, z_{i \neq K} = 0 \mid \boldsymbol{\theta}) && (8) \\
&= \sum_{k=1}^{K} p(\boldsymbol{x} \mid z_k = 1, \boldsymbol{\theta})\, p(z_k = 1 \mid \boldsymbol{\theta}) && (9)
\end{aligned}
$$

GMM as a model with hidden variables

GMM as a model with *hidden (latent) variables* $\boldsymbol{z} = \begin{bmatrix} z_1 & \dots & z_K \end{bmatrix}^\top$.

$$p(\boldsymbol{x} \mid \boldsymbol{\theta}) = \sum_{\boldsymbol{z}} p(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{\theta}) = \sum_{\boldsymbol{z}} p(\boldsymbol{x} \mid \boldsymbol{z}, \boldsymbol{\theta})\, p(\boldsymbol{z} \mid \boldsymbol{\theta}) \tag{6}$$

$$= \sum_{z_1=0}^{1} \cdots \sum_{z_K=0}^{1} p(\boldsymbol{x} \mid z_1, \dots, z_K, \boldsymbol{\theta})\, p(z_1, \dots, z_K \mid \boldsymbol{\theta}) \tag{7}$$

$$\begin{aligned} &= p(\boldsymbol{x} \mid z_1 = 1, z_{i \neq 1} = 0, \boldsymbol{\theta})\, p(z_1 = 1, z_{i \neq 1} = 0 \mid \boldsymbol{\theta}) + \cdots \\ &\quad + p(\boldsymbol{x} \mid z_K = 1, z_{i \neq K} = 0, \boldsymbol{\theta})\, p(z_K = 1, z_{i \neq K} = 0 \mid \boldsymbol{\theta}) \end{aligned} \tag{8}$$

$$= \sum_{k=1}^{K} p(\boldsymbol{x} \mid z_k = 1, \boldsymbol{\theta})\, p(z_k = 1 \mid \boldsymbol{\theta}) \tag{9}$$

$$= \sum_{k=1}^{K} \pi_k \, \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \Sigma_k) \tag{10}$$

How to train a GMM?

1. We described our data with a generative process
2. In a clustering context all data points with $z_k = 1$ are in cluster k
3. But we need to learn/infer/calculate $\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$ from the observed data

BUT this is a circular argument

1. Trivial to calculate the component parameters $\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$
   if we knew data points with the assignment rule $z_k = 1$

How to train a GMM?

1. We described our data with a generative process
2. In a clustering context all data points with $z_k = 1$ are in cluster k
3. But we need to learn/infer/calculate $\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$ from the observed data

BUT this is a circular argument

1. Trivial to calculate the component parameters $\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$
   if we knew data points with the assignment rule $z_k = 1$
2. Trivial to work out data points with the assignment rule $z_k = 1$
   if we knew the component parameters $\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$

How to train a GMM?

1. We described our data with a generative process
2. In a clustering context all data points with $z_k = 1$ are in cluster k
3. But we need to learn/infer/calculate $\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$ from the observed data

BUT this is a circular argument

1. Trivial to calculate the component parameters $\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$
   if we knew data points with the assignment rule $z_k = 1$
2. Trivial to work out data points with the assignment rule $z_k = 1$
   if we knew the component parameters $\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$

$\rightarrow$ We could use the k-means clustering!?

K-means-like training of a GMM

$$p(\mathbf{x} \mid \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{12}$$

Step 1: Apply k-means clustering to the dataset to estimate $\{y_n\}$.

K-means-like training of a GMM

$$p(\mathbf{x} \mid \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{12}$$

Step 1: Apply k-means clustering to the dataset to estimate $\{y_n\}$.
Step 2: Estimate the model parameters based on $\{y_n\}$.

$$\pi_k = \frac{N_k}{N}, \quad \text{where } N_k = |\{y_n = k\}| \tag{13}$$

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{y_n = k} \mathbf{x}_n \tag{14}$$

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{y_n = k} (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^\top \tag{15}$$

K-means-like training of a GMM

$$p(\mathbf{x} \,|\, \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x} \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{12}$$

Step 1: Apply k-means clustering to the dataset to estimate $\{y_n\}$.
Step 2: Estimate the model parameters based on $\{y_n\}$.

$$\pi_k = \frac{N_k}{N}, \quad \text{where } N_k = |\,\{y_n = k\}\,| \tag{13}$$

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{y_n = k} \mathbf{x}_n \tag{14}$$

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{y_n = k} (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^\top \tag{15}$$

Step 3: Update $\{y_n\}$ – find the index of the Gaussian for each $\mathbf{x}_n$:

$$y_n = \arg\max_k \mathcal{N}(\mathbf{x}_n \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{16}$$

K-means-like training of a GMM

$$p(\mathbf{x} \mid \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{12}$$

Step 1: Apply k-means clustering to the dataset to estimate $\{y_n\}$.
Step 2: Estimate the model parameters based on $\{y_n\}$.

$$\pi_k = \frac{N_k}{N}, \quad \text{where } N_k = |\{y_n = k\}| \tag{13}$$

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{y_n = k} \mathbf{x}_n \tag{14}$$

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{y_n = k} (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^\top \tag{15}$$

Step 3: Update $\{y_n\}$ – find the index of the Gaussian for each $\mathbf{x}_n$:

$$y_n = \arg\max_k \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{16}$$

Step 4: Repeat from Step 2 until a convergence criterion is met.

A better way to train a GMM

Instead, let's use $p(y_n = k \mid \mathbf{x}_n)$, i.e., $p(z_{nk} = 1 \mid \mathbf{x}_n)$, denoted by r_{nk} :

$$r_{nk} = p(z_{nk} = 1 \mid \mathbf{x}_n, \boldsymbol{\theta}) = \frac{p(z_{nk} = 1)\, p(\mathbf{x}_n \mid z_{nk} = 1)}{\sum_{k'=1}^{K} p(z_{nk'} = 1)\, p(\mathbf{x}_n \mid z_{nk'} = 1)} \tag{17}$$

$$= \frac{\pi_k \, \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k'=1}^{K} \pi_{k'} \, \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'})} \tag{18}$$

r_{nk} is the *responsibility* that component k takes in explaining the observation $\mathbf{x}_n$.

Obviously, $r_{nk} \geq 0$ and

$$\sum_{k=1}^{K} r_{nk} = 1. \tag{19}$$

Training of a GMM with Expectation Maximisation

1. Initialise the model parameters: $K, \{\pi_k\}, \{\boldsymbol{\mu}_k\}, \{\boldsymbol{\Sigma}_k\}$

Training of a GMM with Expectation Maximisation

1. Initialise the model parameters: $K, \{\pi_k\}, \{\boldsymbol{\mu}_k\}, \{\boldsymbol{\Sigma}_k\}$
2. Calculate $\{r_{nk}\}$ based on the current model [Expectation].

$$r_{nk} = \frac{\pi_k \, \mathcal{N}(\mathbf{x}_n \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k'=1}^{K} \pi_{k'} \, \mathcal{N}(\mathbf{x}_n \,|\, \boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'})}$$

Training of a GMM with Expectation Maximisation

1. Initialise the model parameters: $K, \{\pi_k\}, \{\boldsymbol{\mu}_k\}, \{\boldsymbol{\Sigma}_k\}$
2. Calculate $\{r_{nk}\}$ based on the current model [Expectation].

$$r_{nk} = \frac{\pi_k \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k'=1}^{K} \pi_{k'} \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'})}$$

3. Update the model parameters [Maximisation]

$$\pi_k^{(\text{new})} = \frac{1}{N} \sum_{n=1}^{N} r_{nk} \tag{20}$$

$$\boldsymbol{\mu}_k^{(\text{new})} = \frac{\sum_{n=1}^{N} r_{nk} \mathbf{x}_n}{\sum_{n=1}^{N} r_{nk}} \tag{21}$$

$$\boldsymbol{\Sigma}_k^{(\text{new})} = \frac{\sum_{n=1}^{N} r_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^\top}{\sum_{n=1}^{N} r_{nk}} \tag{22}$$

Training of a GMM with Expectation Maximisation

1. Initialise the model parameters: $K, \{\pi_k\}, \{\boldsymbol{\mu}_k\}, \{\boldsymbol{\Sigma}_k\}$
2. Calculate $\{r_{nk}\}$ based on the current model [Expectation].

$$r_{nk} = \frac{\pi_k \, \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k'=1}^{K} \pi_{k'} \, \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'})}$$

3. Update the model parameters [Maximisation]

$$\pi_k^{(\text{new})} = \frac{1}{N} \sum_{n=1}^{N} r_{nk} \tag{20}$$

$$\boldsymbol{\mu}_k^{(\text{new})} = \frac{\sum_{n=1}^{N} r_{nk} \, \mathbf{x}_n}{\sum_{n=1}^{N} r_{nk}} \tag{21}$$

$$\boldsymbol{\Sigma}_k^{(\text{new})} = \frac{\sum_{n=1}^{N} r_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^\top}{\sum_{n=1}^{N} r_{nk}} \tag{22}$$

4. Repeat from step 2 until a stooping condition is met.

Training of a GMM with Expectation Maximisation *(cont.)*

- The EM method can be used to overcome challenges of using the maximum likelihood estimation method.
- EM derives a *lower bound* $\mathcal{B}$ on the likelihood L, that is $\mathcal{B} \leq L$.
- Instead of maximising L directly, EM maximises $\mathcal{B}$.
- EM does not guarantee a global optimum.
- We will discuss EM in the next lecture.

Why not using a gradient-based optimisation?

Suppose we observe $\mathbf{X}_{N\times D} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$. Assuming that the data points are drawn independently, the likelihood function of all N data points is

$$p(\mathbf{X}\ \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{23}$$

and so the log-likelihood will be

$$L = \log p(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \log \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\} \tag{24}$$

Why not using a gradient-based optimisation?

Suppose we observe $\mathbf{X}_{N \times D} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$. Assuming that the data points are drawn independently, the likelihood function of all N data points is

$$p(\mathbf{X}\ \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{23}$$

and so the log-likelihood will be

$$L = \log p(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \log \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\} \tag{24}$$

We can differentiate L with respect to the model parameters, π_k, $\boldsymbol{\mu}_k$, and $\boldsymbol{\Sigma}_k$, but there is no closed-form solution.

Why not using a gradient-based optimisation?

Suppose we observe $\mathbf{X}_{N\times D} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$. Assuming that the data points are drawn independently, the likelihood function of all N data points is

$$p(\mathbf{X}\ \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{23}$$

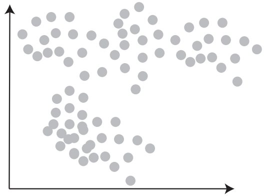
and so the log-likelihood will be

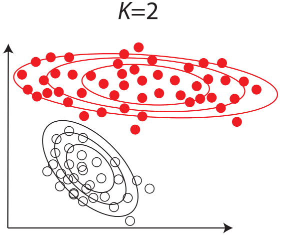
$$L = \log p(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \log \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\} \tag{24}$$

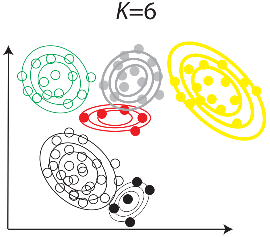
We can differentiate L with respect to the model parameters, π_k, $\boldsymbol{\mu}_k$, and $\boldsymbol{\Sigma}_k$, but there is no closed-form solution.
We can still use gradient-based optimisation. However, the EM method is more efficient, faster, and more widely used.

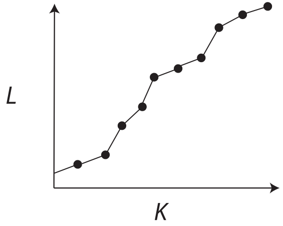
Practical issues

How to choose the number of components K?









Practical issues *(cont.)*

- How to initialise the parameters, $\{\pi_k\}, \{\boldsymbol{\mu}_k\}, \{\boldsymbol{\Sigma}_k\}$?

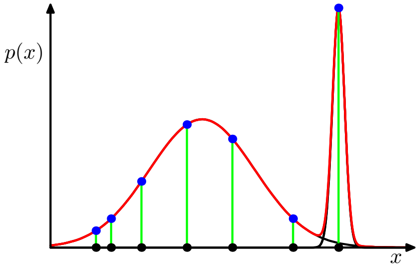
Practical issues *(cont.)*

- How to initialise the parameters, $\{\pi_k\}, \{\boldsymbol{\mu}_k\}, \{\boldsymbol{\Sigma}_k\}$?

  $\rightarrow$ We can use the k-means clustering as we saw in Eqs.(13)–(15).

Practical issues *(cont.)*

The singularity problem with GMMs.



(Credit: Bishop, Figure 9.7)

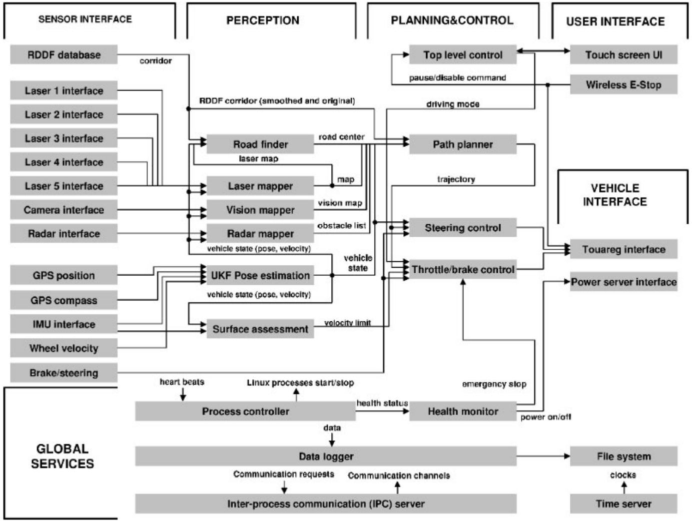
$\rightarrow$ Use the minimum number of instances or minimum variance (flooring) for each mixture component.

GMM application - Autonomous vehicle

Stanley

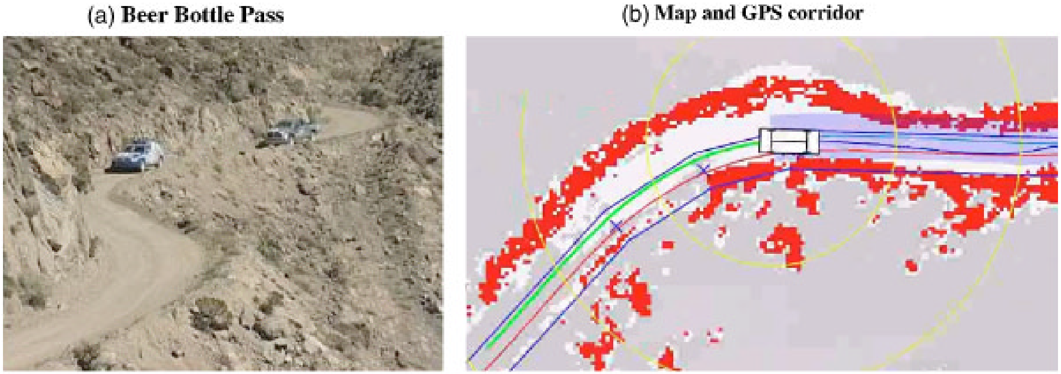
Stanford Racing Team; 2005 DARPA Grand Challenge winner
http://robots.stanford.edu/papers/thrun.stanley05.pdf

Inside Stanley



Stanley figures from Thrun et al., J. Field Robotics 23(9):661, 2006.

Perception and intelligence



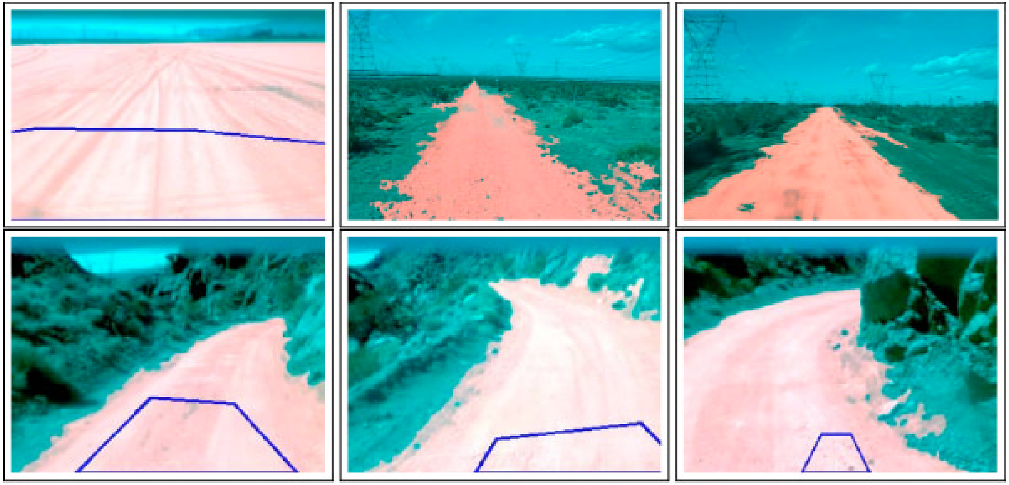
It would look pretty stupid to run off the road,
just because the trip planner said so.

Adapted from Iain Murray's Inf2b slides

How to stay on the road?

Classifying road seems hard. Colours and textures change: road appearance in one place may match ditches elsewhere.

Clustering to stay on the road



Stanley used a Gaussian mixture model.
The cluster just in front is road (unless we already failed).

GMM application - spoken language identification

GMM as a probabilistic density estimator $\rightarrow$ classification

$$p(y = \text{language}_i \mid \boldsymbol{x}, \text{GMM}_i) \tag{25}$$

References:

- A. Dustor and P. Szwarc, "Spoken language identification based on GMM models," ICSES 2010 International Conference on Signals and Electronic Circuits, Gliwice, Poland, 2010, pp. 105-108.
- Torres-Carrasquillo, P.A., Singer, E., Kohler, M.A., Greene, R.J., Reynolds, D.A., Deller Jr., J.R. (2002) Approaches to language identification using Gaussian mixture models and shifted delta cepstral features. Proc. 7th International Conference on Spoken Language Processing (ICSLP 2002), 89-92, doi: 10.21437/ICSLP.2002-74

Quizzes

- Check the derivations from Eq.(6) to Eq.(10).
- Find $\frac{\partial L}{\partial \boldsymbol{\mu}_k}$ in Eq.(24).