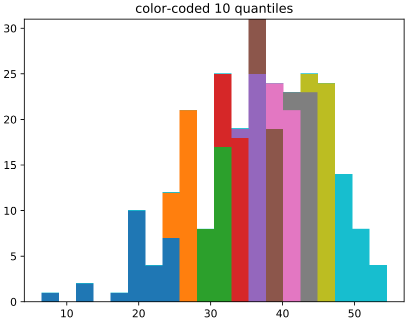
Lecture 18 — **G**aussian **M**ixture **M**odels.

Alex Schwing and Matus Telgarsky

March 29, 2018

Announcements.

- Midterm has been graded;
  grades are in compass;
  midterms handed back in TA office hours.
  This week there **is** a Friday TA office hour.
- Midterm grade histogram, with 10 quantiles:



Schedule for today.

- k-means review.
- (Almost) Deriving GMMs by extending k-means.
- Probability model behind GMMs.
- GMMs Examples.
- Ancillary topics.

k-means review.

k-means review.

Key points with k-means.

- The objective function.
- The standard algorithm (Lloyd's method).
- Standard application: vector quantization.

k-means review: objective function.

The **exemplar-based**, **hard-assignment** k-means clustering objective.

$$\min_{\mu_1,\ldots,\mu_k} \sum_{i=1}^{n} \min_{j} \|x_i - \mu_j\|_2^2 = \min_{\mu_1,\ldots,\mu_k} \min_{\substack{A\in\{0,1\}^{n\times k} \\ A\mathbf{1}_k=\mathbf{1}_n}} \sum_{i=1}^{n} \sum_{j=1}^{k} A_{ij} \|x_i - \mu_j\|_2^2.$$

Remarks.

- Second form makes assignment of data to centers explicit.
- NP-hard even when $d = 2$; methods do not globally optimize.

k-means review: standard algorithm.

1. Let initial clusters $(C_1, \ldots, C_k)$ be given.
2. Alternate the following two steps:
   - 2.1 **(Recentering.)** Hold $(C_1, \ldots C_k)$ fixed, optimally update centers: for all j, $\mu_j := \text{mean}(C_j)$.
   - 2.2 **(Reassignment.)** Hold $(\mu_1, \ldots, \mu_k)$ fixed, optimally update clusters: put x_i in C_j iff $\|x_i - \mu_j\| = \min_l \|x_i - \mu_l\|$ (breaking ties arbitrarily).

Remarks.

- This is **alternating minimization**.
- Initialization is crucial; standard initialization now is "`kmeans++`" (see k-means lecture).
- With good initialization, in practice method quickly finds good clusters; in theory, not so much.

k-means review: vector quantization.

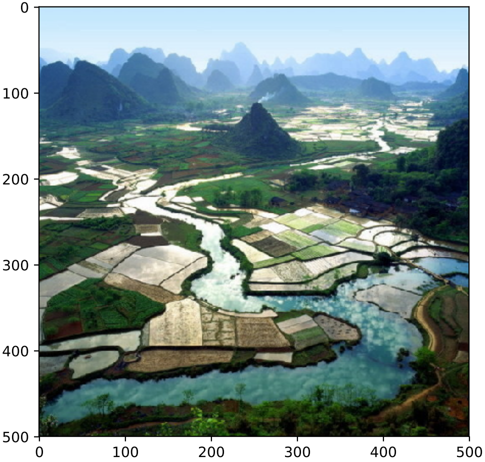
A standard application of k-means is **vector quantization**.

1. Obtain data $(x_i)_{i=1}^n$.
2. Run k-means on $(x_i)_{i=1}^n$, obtain $(\mu_1, \ldots, \mu_k)$.
3. Output new data $(\mu(x_i))_{i=1}^n$ where $\mu(x_i) = \arg\min_{\mu_j} \|x_i - \mu_j\|_2$, the center closest to x_i.

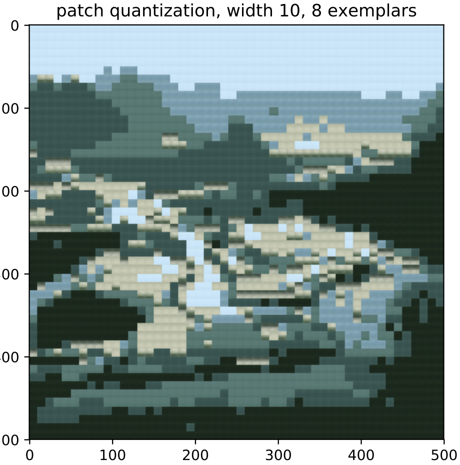
Remarks.

- In words: replace data points with their closest means.
- This gives a quick way to "compress" data.
- This is useful in speech and vision data (amongst others).

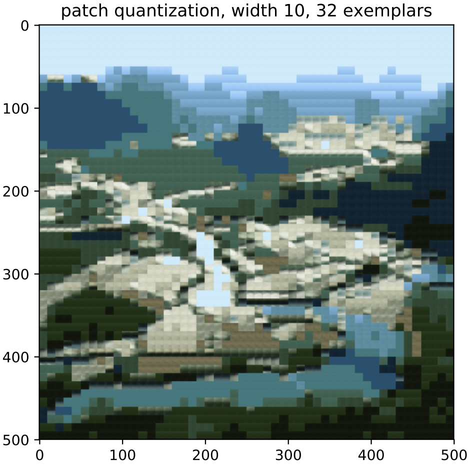
k-means review: vector quantization.



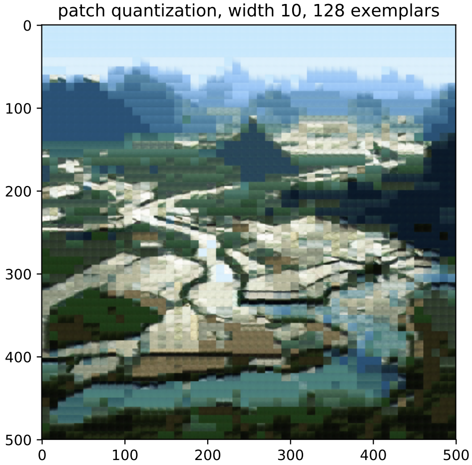
k-means review: vector quantization.



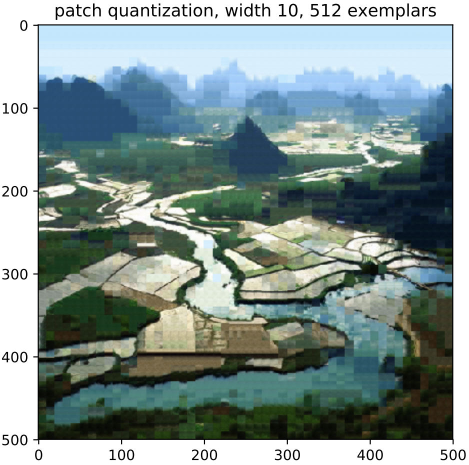
k-means review: vector quantization.



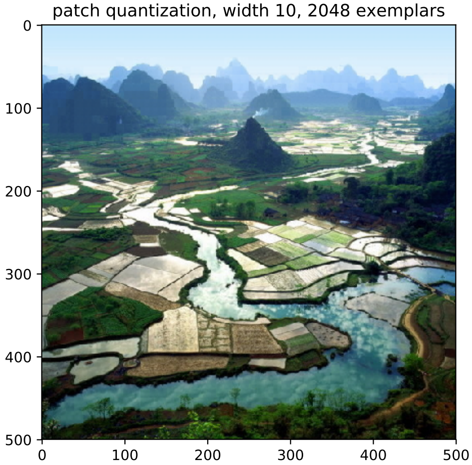
k-means review: vector quantization.



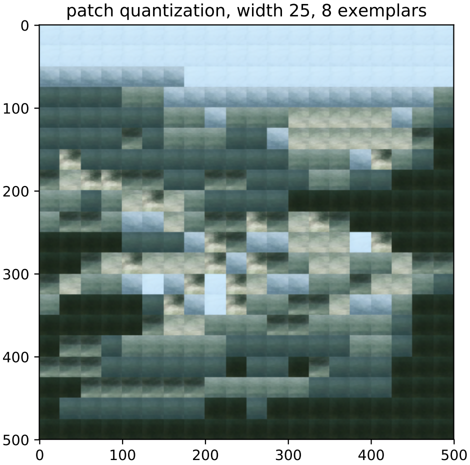
k-means review: vector quantization.



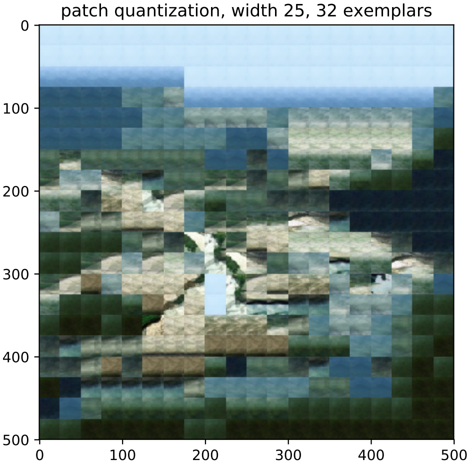
k-means review: vector quantization.



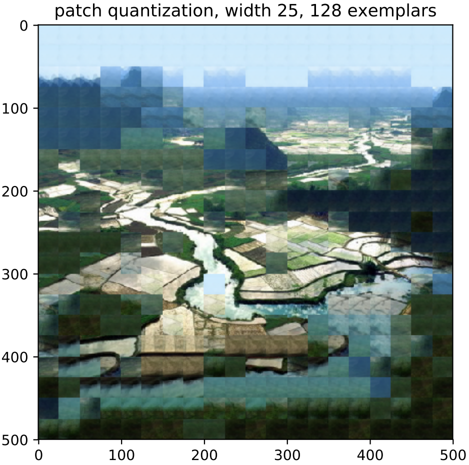
k-means review: vector quantization.



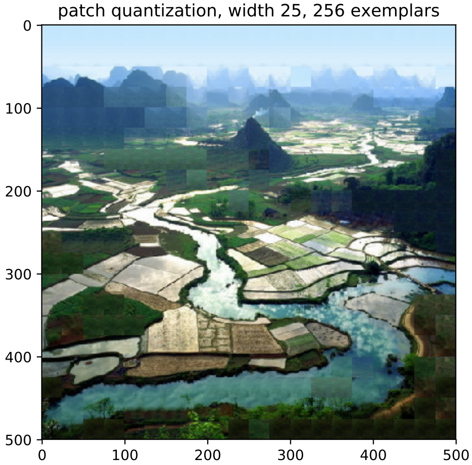
k-means review: vector quantization.



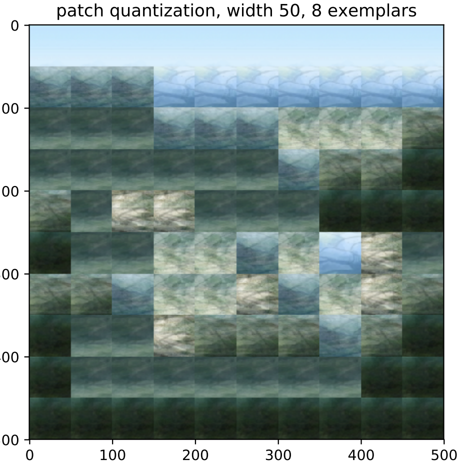
k-means review: vector quantization.



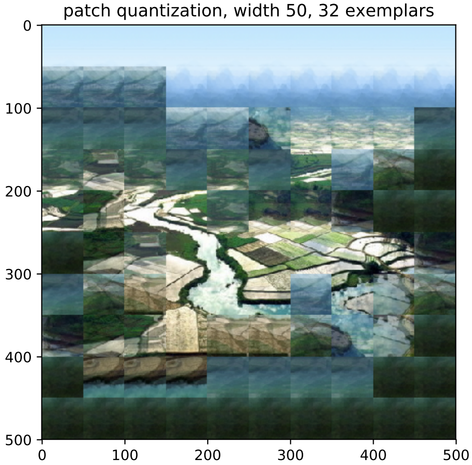
k-means review: vector quantization.



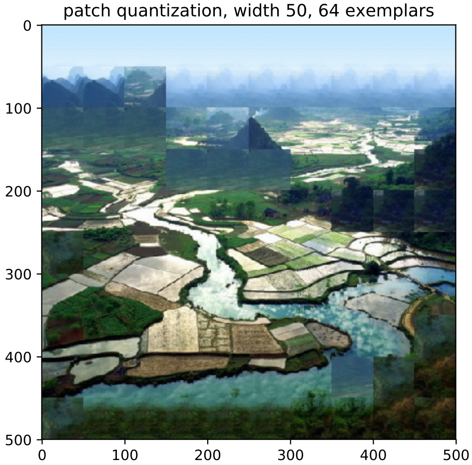
k-means review: vector quantization.



k-means review: vector quantization.



k-means review: vector quantization.



(Almost) Deriving GMMs by extending k-means.

(Almost) Deriving GMMs by extending k-means.

Let's extend k-means in two ways:

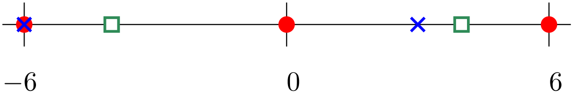
- Soft assignments.
- Non-spherical clusters.

k-means with soft assignments.

Suppose assignment matrix $A \in [0,1]^{n\times k}$ has *probability vectors* for rows:

$$\min_{\mu_1,\ldots,\mu_k} \min_{\substack{A\in[0,1]^{n\times k}\\ A\mathbf{1}_k=\mathbf{1}_n}} \sum_{i=1}^{n}\sum_{j=1}^{k} A_{ij}\|x_i-\mu_j\|_2^2.$$

Example:



Soft assignments give a “more pleasing” fit?

- red = data, blue = hard centers, green = soft centers.
- Soft clustering allows a symmetric solution; hard does not!

k-means with soft assignments.

Directly (min over a larger set),

$$\min_{\mu_1,\ldots,\mu_k} \min_{\substack{A\in[0,1]^{n\times k}\\ A\mathbf{1}_k=\mathbf{1}_n}} \sum_{i=1}^{n}\sum_{j=1}^{k} A_{ij}\|x_i-\mu_j\|_2^2 \leq \min_{\mu_1,\ldots,\mu_k} \min_{\substack{A\in\{0,1\}^{n\times k}\\ A\mathbf{1}_k=\mathbf{1}_n}} \sum_{i=1}^{n}\sum_{j=1}^{k} A_{ij}\|x_i-\mu_j\|_2^2.$$

On the other hand,

k-means with soft assignments.

Directly (min over a larger set),

$$\min_{\mu_1,\ldots,\mu_k} \min_{\substack{A\in[0,1]^{n\times k}\\ A\mathbf{1}_k=\mathbf{1}_n}} \sum_{i=1}^{n}\sum_{j=1}^{k} A_{ij}\|x_i-\mu_j\|_2^2 \leq \min_{\mu_1,\ldots,\mu_k} \min_{\substack{A\in\{0,1\}^{n\times k}\\ A\mathbf{1}_k=\mathbf{1}_n}} \sum_{i=1}^{n}\sum_{j=1}^{k} A_{ij}\|x_i-\mu_j\|_2^2.$$

On the other hand,

$$\begin{aligned}
&\min_{\mu_1,\ldots,\mu_k} \min_{\substack{A\in[0,1]^{n\times k}\\ A\mathbf{1}_k=\mathbf{1}_n}} \sum_{i=1}^{n}\sum_{j=1}^{k} A_{ij}\|x_i-\mu_j\|_2^2 \\
&\geq \min_{\mu_1,\ldots,\mu_k} \min_{\substack{A\in[0,1]^{n\times k}\\ A\mathbf{1}_k=\mathbf{1}_n}} \sum_{i=1}^{n}\sum_{j=1}^{k} A_{ij}\min_{l}\|x_i-\mu_l\|_2^2 \\
&\geq \min_{\mu_1,\ldots,\mu_k} \min_{\substack{A\in[0,1]^{n\times k}\\ A\mathbf{1}_k=\mathbf{1}_n}} \sum_{i=1}^{n}\left(\sum_{j=1}^{k} A_{ij}\right)\min_{l}\|x_i-\mu_l\|_2^2 \\
&= \min_{\mu_1,\ldots,\mu_k} \min_{\substack{A\in\{0,1\}^{n\times k}\\ A\mathbf{1}_k=\mathbf{1}_n}} \sum_{i=1}^{n}\sum_{j=1}^{k} A_{ij}\|x_i-\mu_j\|_2^2.
\end{aligned}$$

k-means with soft assignments.

Therefore

$$\min_{\mu_1,\ldots,\mu_k} \min_{\substack{A\in[0,1]^{n\times k} \\ A\mathbf{1}_k=\mathbf{1}_n}} \sum_{i=1}^{n}\sum_{j=1}^{k} A_{ij}\|x_i-\mu_j\|_2^2 = \min_{\mu_1,\ldots,\mu_k} \min_{\substack{A\in\{0,1\}^{n\times k} \\ A\mathbf{1}_k=\mathbf{1}_n}} \sum_{i=1}^{n}\sum_{j=1}^{k} A_{ij}\|x_i-\mu_j\|_2^2,$$

and even soft assignment has a globally optimal **hard assignment**!

- Therefore: minimization alone won't give soft A_{ij} choice.
- In earlier example, symmetric local optimum was not global!

k-means with spherical clusters.

Consider a single cluster C_j; the cost is

$$\sum_{x_i \in C_j} \|x_i - \mu_j\|_2^2 = (x_i - \mu_j)^\top (x_i - \mu_j).$$

How can this be made this non-spherical?

k-means with spherical clusters.

Consider a single cluster C_j; the cost is

$$\sum_{x_i \in C_j} \|x_i - \mu_j\|_2^2 = (x_i - \mu_j)^\top (x_i - \mu_j).$$

How can this be made this non-spherical?
Introduce a positive definite matrix $M = Q\Lambda Q^\top$:

$$\sum_{x_i \in C_j} (x_i - \mu_j)^\top M (x_i - \mu_j) = \sum_{x_i \in C_j} \left(Q^\top (x_i - \mu_j)\right)^\top \Lambda \left(Q^\top (x_i - \mu_j)\right).$$

"Non-spherical" because $\{x \in \mathbb{R}^d : x^\top M x = 1\}$ is an ellipse.

k-means with spherical clusters.

Great. Let's optimize

$$\min_{\substack{\mu_1,\ldots,\mu_k \\ M_1,\ldots,M_k \\ M_j \succ 0}} \min_{\substack{A \in [0,1]^{n \times k} \\ A\mathbf{1}_k = \mathbf{1}_n}} \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij}(x_i - \mu_j)^\top M_j (x_i - \mu_j)$$

...

k-means with spherical clusters.

Great. Let's optimize

$$\min_{\substack{\mu_1,\ldots,\mu_k \\ M_1,\ldots,M_k \\ M_j \succ 0}} \min_{\substack{A \in [0,1]^{n \times k} \\ A\mathbf{1}_k = \mathbf{1}_n}} \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij}(x_i - \mu_j)^\top M_j (x_i - \mu_j)$$

... not great! Solve this by taking $M = cI$ with $c \downarrow 0$...

k-means with spherical clusters.

Great. Let's optimize

$$\min_{\substack{\mu_1,\ldots,\mu_k \\ M_1,\ldots,M_k \\ M_j \succ 0}} \min_{\substack{A \in [0,1]^{n \times k} \\ A\mathbf{1}_k = \mathbf{1}_n}} \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij}(x_i - \mu_j)^\top M_j (x_i - \mu_j)$$

... not great! Solve this by taking $M = cI$ with $c \downarrow 0$...

Fix: *regularize M.*

k-means with spherical clusters.

To determine regularization, consider a single cluster:

$$\sum_{x_i \in C_j} (x_i - \mu_j)^\top M_j (x_i - \mu_j) + \mathrm{Reg}(M).$$

Apply ∇_M and set to zero:

$$\sum_{x_i \in C} (x_i - \mu_j)(x_i - \mu_j)^\top = -\nabla_M \mathrm{Reg}(M).$$

k-means with spherical clusters.

To determine regularization, consider a single cluster:

$$\sum_{x_i \in C_j} (x_i - \mu_j)^\top M_j (x_i - \mu_j) + \mathrm{Reg}(M).$$

Apply ∇_M and set to zero:

$$\sum_{x_i \in C} (x_i - \mu_j)(x_i - \mu_j)^\top = -\nabla_M \mathrm{Reg}(M).$$

Natural choice: $\mathrm{Reg}(M) := -\ln\det M$, so $\nabla_M \mathrm{Reg}(M) = -M^{-1}$, and

$$\sum_{x_i \in C} (x_i - \mu_j)(x_i - \mu_j)^\top = -\nabla_M \mathrm{Reg}(M) = M^{-1},$$

the *inverse sample covariance*!

k-means with spherical clusters.

To determine regularization, consider a single cluster:

$$\sum_{x_i \in C_j} (x_i - \mu_j)^\top M_j (x_i - \mu_j) + \mathrm{Reg}(M).$$

Apply ∇_M and set to zero:

$$\sum_{x_i \in C} (x_i - \mu_j)(x_i - \mu_j)^\top = -\nabla_M \mathrm{Reg}(M).$$

Natural choice: $\mathrm{Reg}(M) := -\ln\det M$, so $\nabla_M \mathrm{Reg}(M) = -M^{-1}$, and

$$\sum_{x_i \in C} (x_i - \mu_j)(x_i - \mu_j)^\top = -\nabla_M \mathrm{Reg}(M) = M^{-1},$$

the *inverse sample covariance*!

Remark. If $\ln\det$ seems weird, try diagonal covariances.

Extensions to k-means.

Soft assignment: objective function

$$\min_{\mu_1,\ldots,\mu_k} \min_{\substack{A\in[0,1]^{n\times k}\\ A\mathbf{1}_k=\mathbf{1}_n}} \sum_{i=1}^{n}\sum_{j=1}^{k} A_{ij}\|x_i-\mu_j\|_2^2.$$

- Gives some symmetric solutions, but doesn't decrease cost; Unclear how to optimize.

Elliptical clusters: replace cluster cost $\sum_{x_i\in C_j}\|x_i-\mu_j\|^2$ with

$$\sum_{x_i\in C_j}(x_i-\mu_j)^\top M_j(x_i-\mu_j) - \ln\det M.$$

- Setting gradient to zero, get $M^{-1}=\sum_{x_i\in C_j}(x_i-\mu_j)(x_i-\mu_j)^\top$.

Probability model behind GMMs.

Probability model behind GMMs.

First recall *maximum likelihood*: choose parameters according to

$$\arg\max_{\theta\in\Theta} \prod_{x\in S} p_\theta(x) = \arg\min_{\theta\in\Theta} -\ln\left(\prod_{x\in S} p_\theta(x)\right) = \arg\min_{\theta\in\Theta} \sum_{x\in S} \ln\frac{1}{p_\theta(x)}.$$

Probability model behind GMMs.

First recall *maximum likelihood*: choose parameters according to

$$\arg\max_{\theta\in\Theta} \prod_{x\in S} p_\theta(x) = \arg\min_{\theta\in\Theta} -\ln\left(\prod_{x\in S} p_\theta(x)\right) = \arg\min_{\theta\in\Theta} \sum_{x\in S} \ln \frac{1}{p_\theta(x)}.$$

If $\theta = (\mu, \Sigma)$ and $\Theta = \{(\mu, \Sigma) \in \mathbb{R}^d \times \mathbb{R}^{d^2} : \Sigma \succ 0\}$ and

$$p_\theta(x) = \left((2\pi)^d \det \Sigma\right)^{-1/2} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right)$$

(Gaussian), so

$$\arg\max_{\theta\in\Theta} \prod_{x\in S} p_\theta(x) = \arg\min_{\theta\in\Theta} \sum_{x\in S} (x-\mu)^\top \Sigma^{-1}(x-\mu) + \ln\det\Sigma.$$

Familiar?

Probability model behind GMMs.

What we have so far: letting p_θ denote Gaussian density,

$$\underset{(\theta_1,\ldots,\theta_k)\in\Theta^k}{\arg\max} \min_{\substack{A\in[0,1]^{n\times k}\\ A\mathbf{1}_k=\mathbf{1}_n}} \sum_{i=1}^{n}\sum_{j=1}^{k} A_{ij} \ln p_\theta(x_i).$$

- This is just a reinterpretation of the previous section.
- We still don't have a way to choose soft $A \in [0,1]^{n\times k}$! For this, need a *complete* likelihood model.

Complete likelihood model for GMMS.

Data points are created in two steps:

$$\begin{aligned} Y &\sim \text{Discrete}(\pi_1, \ldots, \pi_k) && \text{(choose cluster component)}, \\ (X|Y = j) &\sim \mathcal{N}(\mu_j, \Sigma_j) && \text{(sample Gaussian } j\text{)}. \end{aligned}$$

- Random variable Y corresponds to a row of $A \in [0,1]^{n \times k}$.
- Given full parameters

  $$\theta = ((\pi_1, \theta_1), \ldots, (\pi_k, \theta_k)) = ((\pi_1, \mu_1, \Sigma_1), \ldots, (\pi_k, \mu_k, \Sigma_k)),$$

  then

  $$p_\theta(x) = \sum_{j=1}^{k} \pi_j p_{\theta_j}(x).$$

Complete likelihood model for GMMS.

Suppose we knew the true assignments $A \in [0,1]^{n \times k}$.
Then fitting the parameters to data means

$$\arg\max_{\theta \in \Theta} \sum_{i=1}^{n} \sum_{j=1}^{k} A_{ij} \ln p_{\theta_j}(x_i)$$

$$= \arg\max_{\theta \in \Theta} \sum_{i=1}^{n} \sum_{j=1}^{k} A_{ij} \left(2 \ln \pi_j - (x_i - \mu_j)^\top \Sigma_j^{-1} (x - \mu_j) - \ln \det(\Sigma_j) \right).$$

(For each i, A_{ij} isolates the single true mixture component.)

Complete likelihood model for GMMS.

Suppose we knew the true assignments $A \in [0,1]^{n\times k}$.
Then fitting the parameters to data means

$$\arg\max_{\theta\in\Theta} \sum_{i=1}^{n}\sum_{j=1}^{k} A_{ij} \ln p_{\theta_j}(x_i)$$

$$= \arg\max_{\theta\in\Theta} \sum_{i=1}^{n}\sum_{j=1}^{k} A_{ij} \left(2\ln \pi_j - (x_i - \mu_j)^\top \Sigma_j^{-1}(x - \mu_j) - \ln\det(\Sigma_j)\right).$$

(For each i, A_{ij} isolates the single true mixture component.)
As before, *sequentially* setting derivatives to zero gives update

$$\pi_j' := \frac{\sum_i A_{ij}}{\sum_{i,j} A_{ij}} = \frac{\sum_i A_{ij}}{n} \propto \sum_i A_{ij},$$

$$\mu_j' := \frac{\sum_i A_{ij} x_i}{\sum_i A_{ij}} = \frac{\sum_i A_{ij} x_i}{n\pi_j'},$$

$$\Sigma_j' := \frac{\sum_i A_{ij}(x_i - \mu_j')(x_i - \mu_j')^\top}{\sum_i A_{ij}} = \frac{\sum_i A_{ij}(x_i - \mu_j')(x_i - \mu_j')^\top}{n\pi_j'}.$$

("*Sequentially*" is why μ_j' not μ_j for Σ_j'.)

Complete likelihood model for GMMS.

- If we fixed the data and model parameters, we can update A_{ij} by expectations:

$$A'_{ij} := \Pr[Y_i = j | X_i = x_i] \quad = \frac{\Pr[Y_i = j]\Pr[X_i = x_i | Y_i = j]}{\Pr[X_i = x_i]}$$

$$= \frac{\pi_j p_{\theta_j}(x_i)}{\sum_l \pi_l p_{\theta_l}(x_i)}.$$

- For now this merely seems reasonable; we will justify it next lecture.

Full update rule for GMMs.

Initialize in some way; then alternate the following two steps.

1. Fix model parameters, update assignments:

$$A_{ij} := \frac{\pi_j p_{\theta_j}(x_i)}{\sum_l \pi_l p_{\theta_l}(x_i)},$$

   where p_{θ_j} is Gaussian density with parameters $\theta_j = (\mu_j, \Sigma_j)$.

2. Fix assignments, update parameters:

$$\pi'_j := \frac{\sum_i A_{ij}}{n},$$

$$\mu'_j := \frac{\sum_i A_{ij} x_i}{n\pi'_j},$$

$$\Sigma'_j := \frac{\sum_i A_{ij}(x_i - \mu'_j)(x_i - \mu'_j)^\top}{n\pi'_j}.$$

Expectation-maximization.

- This method is called Expectation-Maximization (E-M).
- Matrix $A \in [0,1]^{n\times k}$ is often called "responsibilities".
- Next lecture we will justify the method,
  in particular the choice of A_{ij},
  and establish it increases likelihood.

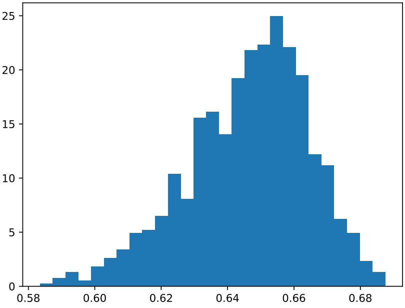
Examples.

Examples.

- Univariate example: Pearson's crabs!
- Bivariate example: synthetic data.

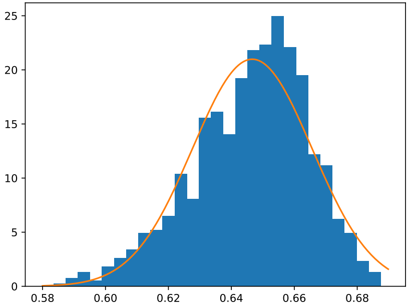
Pearson's crabs.

Statistician Karl Pearson wanted to understand the distribution of "forehead breadth to body length" for 1000 crabs



Pearson's crabs.

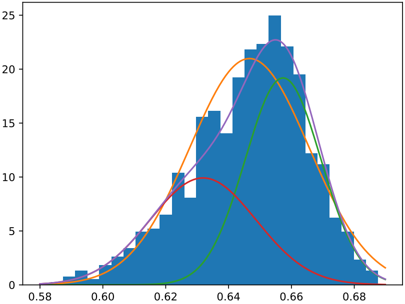
Statistician Karl Pearson wanted to understand the distribution of "forehead breadth to body length" for 1000 crabs



Doesn't look Gaussian!

Pearson's crabs.

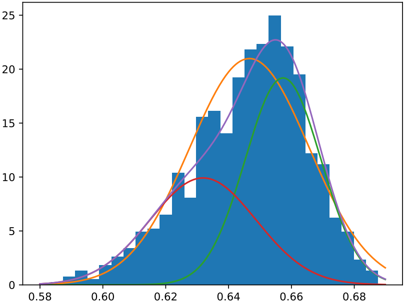
Statistician Karl Pearson wanted to understand the distribution of "forehead breadth to body length" for 1000 crabs



Pearson fit a **mixture of two Gaussians**.

Pearson's crabs.

Statistician Karl Pearson wanted to understand the distribution of "forehead breadth to body length" for 1000 crabs



Pearson fit a **mixture of two Gaussians**.

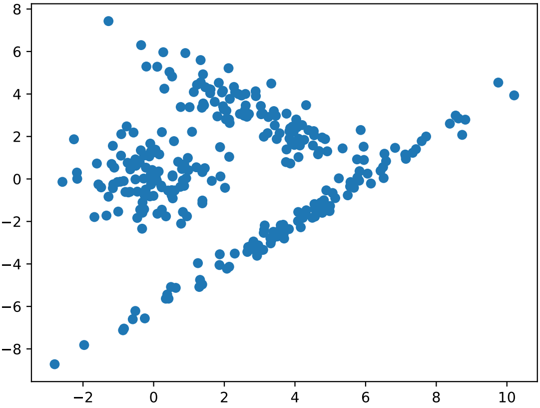
Remark. Pearson did *not* use E-M. For this he invented the "method of moments" and obtained a solution by hand.

Aside: why Gaussians at all?

- You can argue Gaussian is a good model for *single* populations thanks to the CLT (Central Limit Theorem).
- Pearson, seeing the skewed distribution, felt there are two populations.
- Treating these populations as independent, one gets a mixture of Gaussians.

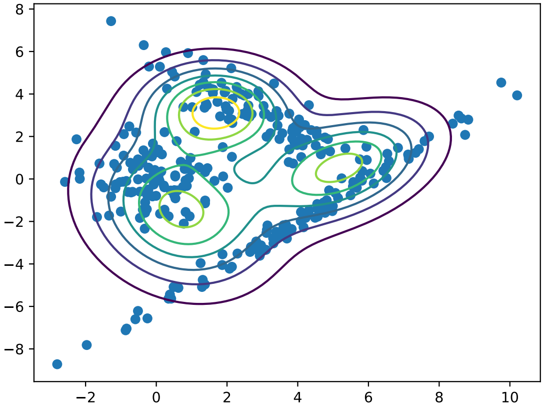
Example: synthetic bivariate data.

Plot of contours of $p_\theta(x) = \sum_{i=1}^{k} \pi_j p_{\theta_j}(x)$.



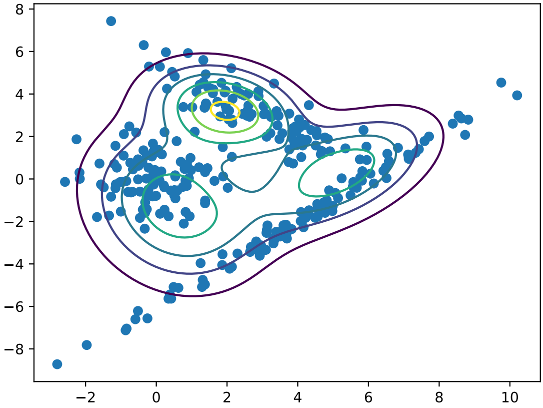
Example: synthetic bivariate data.

Plot of contours of $p_\theta(x) = \sum_{i=1}^{k} \pi_j p_{\theta_j}(x)$.



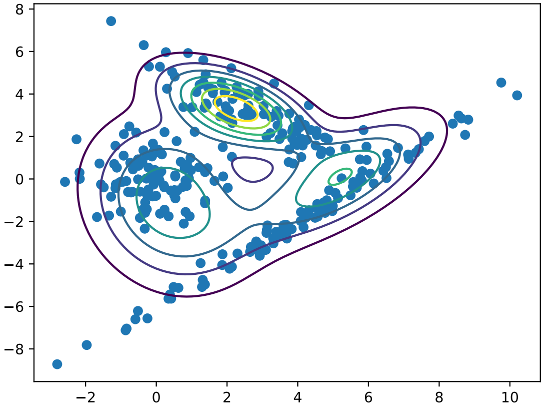
Example: synthetic bivariate data.

Plot of contours of $p_\theta(x) = \sum_{i=1}^{k} \pi_j p_{\theta_j}(x)$.



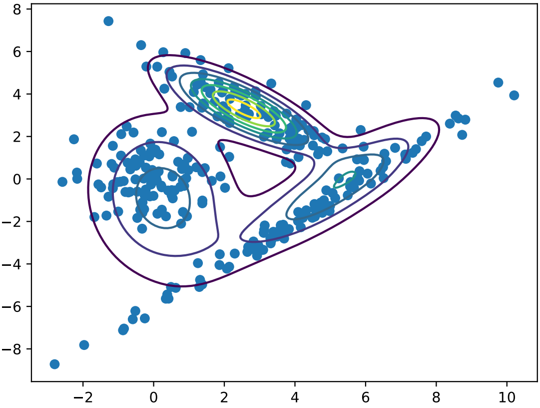
Example: synthetic bivariate data.

Plot of contours of $p_\theta(x) = \sum_{i=1}^{k} \pi_j p_{\theta_j}(x)$.



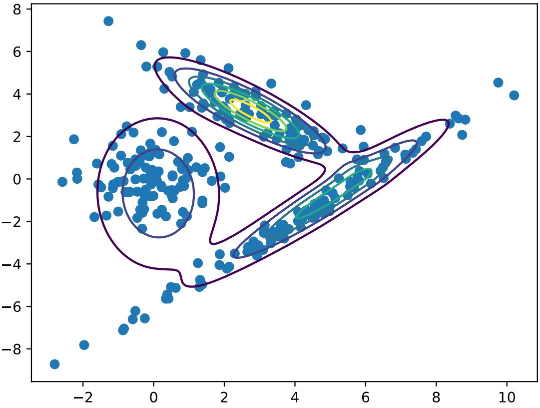
Example: synthetic bivariate data.

Plot of contours of $p_\theta(x) = \sum_{i=1}^{k} \pi_j p_{\theta_j}(x)$.



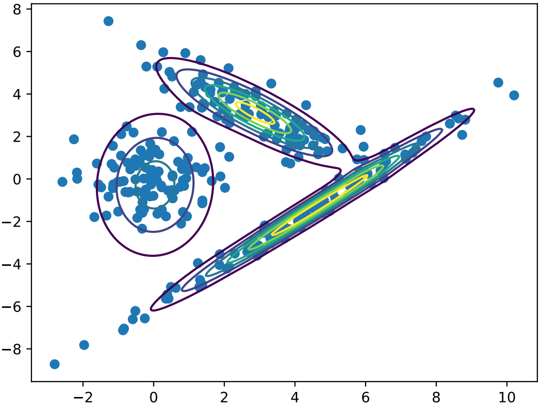
Example: synthetic bivariate data.

Plot of contours of $p_\theta(x) = \sum_{i=1}^{k} \pi_j p_{\theta_j}(x)$.



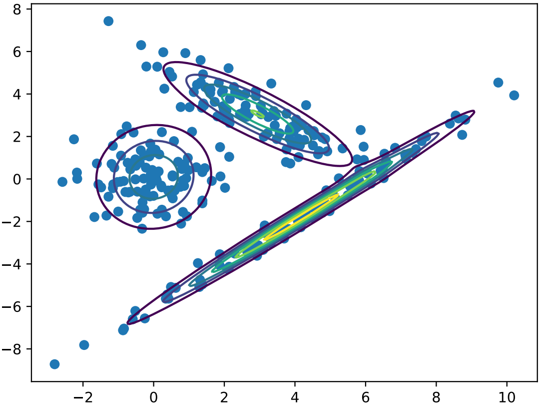
Example: synthetic bivariate data.

Plot of contours of $p_\theta(x) = \sum_{i=1}^{k} \pi_j p_{\theta_j}(x)$.



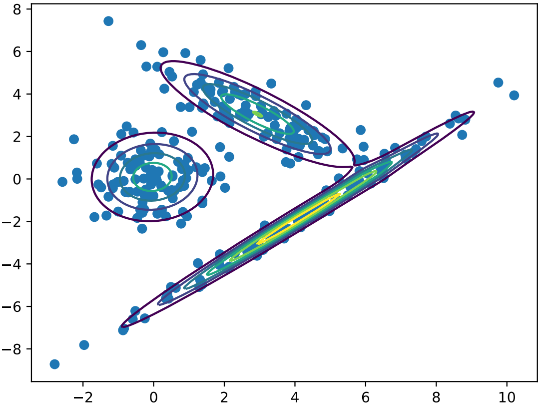
Example: synthetic bivariate data.

Plot of contours of $p_\theta(x) = \sum_{i=1}^{k} \pi_j p_{\theta_j}(x)$.



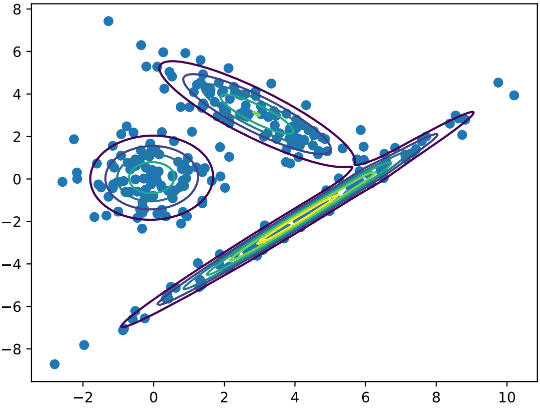
Example: synthetic bivariate data.

Plot of contours of $p_\theta(x) = \sum_{i=1}^{k} \pi_j p_{\theta_j}(x)$.



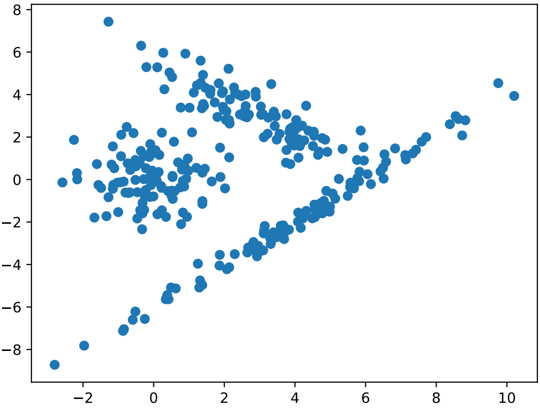
Example: synthetic bivariate data.

Plot of contours of $p_\theta(x) = \sum_{i=1}^{k} \pi_j p_{\theta_j}(x)$.



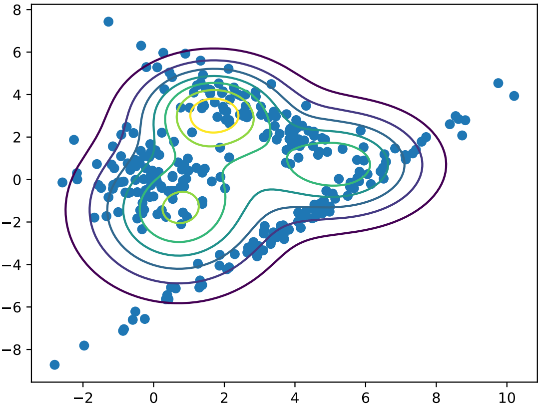
Example: synthetic bivariate data.

Common simplification: *diagonal* covariance matrices.
(Now $\mathcal{O}(kd)$ parameters, not $\mathcal{O}(kd^2)$.)



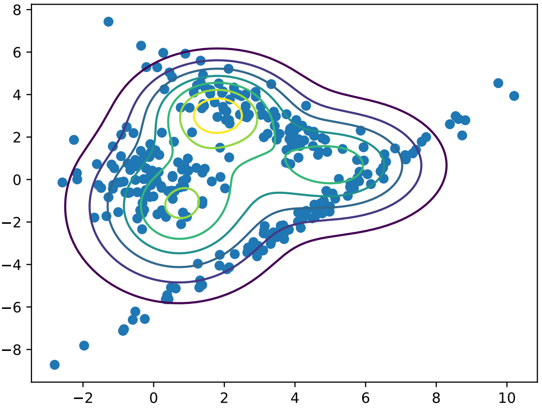
Example: synthetic bivariate data.

Common simplification: *diagonal* covariance matrices.
(Now $\mathcal{O}(kd)$ parameters, not $\mathcal{O}(kd^2)$.)



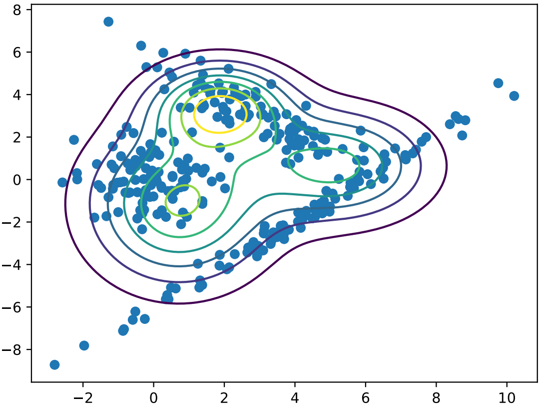
Example: synthetic bivariate data.

Common simplification: *diagonal* covariance matrices.
(Now $\mathcal{O}(kd)$ parameters, not $\mathcal{O}(kd^2)$.)



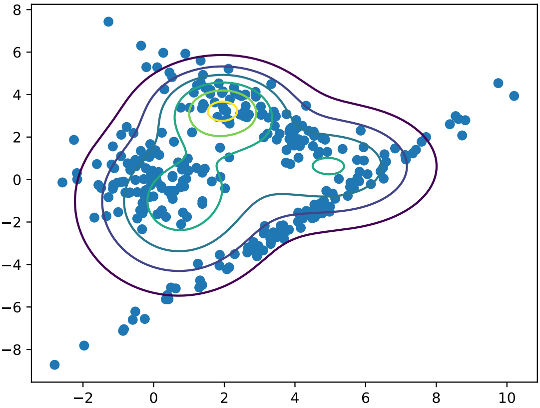
Example: synthetic bivariate data.

Common simplification: *diagonal* covariance matrices.
(Now $\mathcal{O}(kd)$ parameters, not $\mathcal{O}(kd^2)$.)



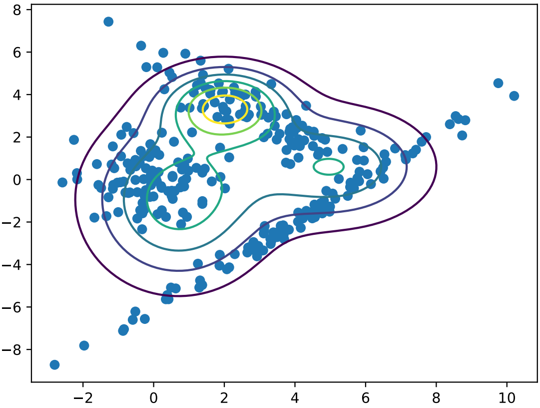
Example: synthetic bivariate data.

Common simplification: *diagonal* covariance matrices.
(Now $\mathcal{O}(kd)$ parameters, not $\mathcal{O}(kd^2)$.)



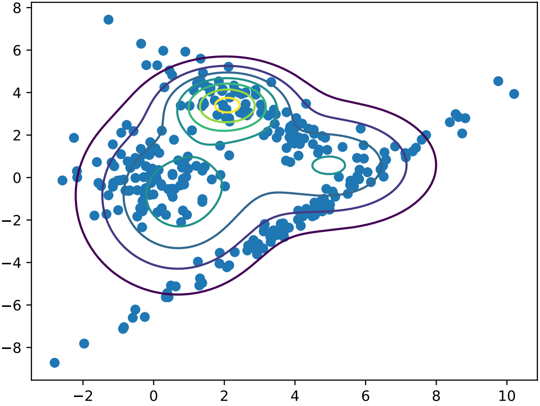
Example: synthetic bivariate data.

Common simplification: *diagonal* covariance matrices.
(Now $\mathcal{O}(kd)$ parameters, not $\mathcal{O}(kd^2)$.)



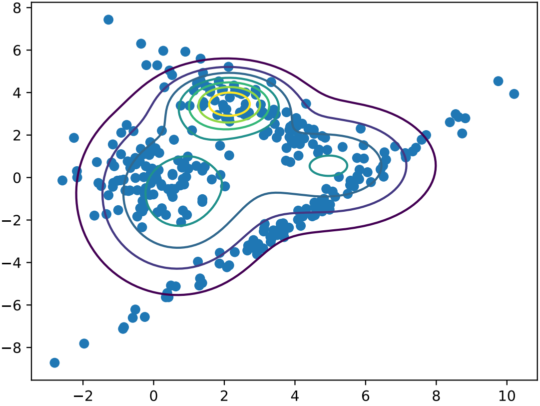
Example: synthetic bivariate data.

Common simplification: *diagonal* covariance matrices.
(Now $\mathcal{O}(kd)$ parameters, not $\mathcal{O}(kd^2)$.)



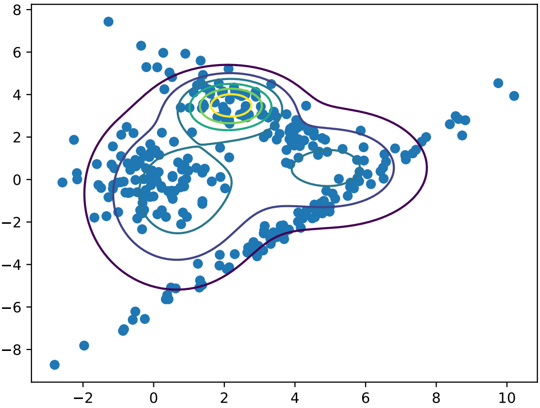
Example: synthetic bivariate data.

Common simplification: *diagonal* covariance matrices.
(Now $\mathcal{O}(kd)$ parameters, not $\mathcal{O}(kd^2)$.)



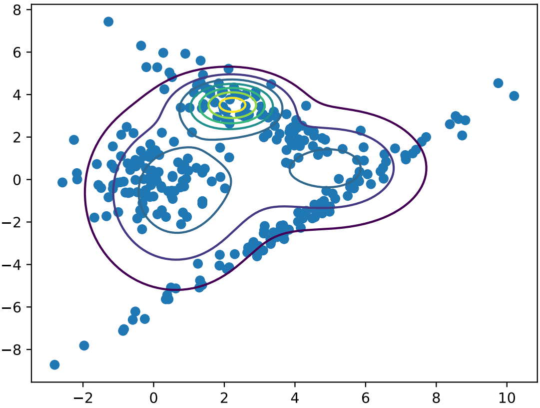
Example: synthetic bivariate data.

Common simplification: *diagonal* covariance matrices.
(Now $\mathcal{O}(kd)$ parameters, not $\mathcal{O}(kd^2)$.)



Example: synthetic bivariate data.

Common simplification: *diagonal* covariance matrices.
(Now $\mathcal{O}(kd)$ parameters, not $\mathcal{O}(kd^2)$.)



Ancillary topics.

Ancillary topics.

- Choosing k: use *elbow* as before.
  (Or some “Bayesian Information Criterion”.)
- Initialization: random, or k-means.
- “Singularaties” / collapsing components.
- Deriving k-means from GMMs.

“Singularities”.

Consider two data points in $\mathbb{R}$, two centers.
E-M can shrink a cluster onto one point and delete it!

(More details in lecture.)

In practice, implementations impose lower bounds on covariance eigenvalues, and sometimes randomly re-initialize small clusters.

Deriving k-means from GMMs.

Consider $\Sigma_j := \sigma^2 I$ with $\sigma \to 0$,
make mixture proportions $(\pi_1, \ldots, \pi_k) = (1/k, \ldots, 1/k)$ uniform,
otherwise leave algorithm as before.

1. **(Re-assignment.)** Since $A_{ij} \propto p_{\theta_j}(x_i)$, soft assignments become hard assignments as $\sigma \to 0$.
2. **(Parameter maximization.)** as in k-means, mean update is just sample mean!

Key points from this lecture.

Key points from this lecture.

1. The GMM likelihood model: $p_\theta(x) = \sum_j \pi_j p_{\theta_j}(x)$ where p_{θ_j} is a multivariate Gaussian density.
2. The E-M algorithm for GMMs.