MLE part 2

Gaussian Mixture Model

- Suppose data is drawn from k Gaussians, meaning

$$Y = j \sim \text{Discrete}(\pi_1, \ldots, \pi_k),$$
$$\boldsymbol{X} = \boldsymbol{x} | Y = j \sim \mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j),$$

and the parameters are $\boldsymbol{\theta} = ((\pi_1, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), \ldots, (\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k))$.
(Note: this is a **generative** model, and we have a way to sample.)

Gaussian Mixture Model

- Suppose data is drawn from k Gaussians, meaning

$$Y = j \sim \text{Discrete}(\pi_1, \ldots, \pi_k),$$
$$\boldsymbol{X} = \boldsymbol{x} | Y = j \sim \mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j),$$

  and the parameters are $\boldsymbol{\theta} = ((\pi_1, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), \ldots, (\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k))$.
  (Note: this is a **generative** model, and we have a way to sample.)

- The probability density (with parameters $\boldsymbol{\theta} = ((\pi_j, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j))_{j=1}^k$) at a given $\boldsymbol{x}$ is

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{j=1}^k p_{\boldsymbol{\theta}}(\boldsymbol{x}|y = j) p_{\boldsymbol{\theta}}(y = j) = \sum_{j=1}^k p_{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j}(\boldsymbol{x}|Y = j)\pi_j,$$

  and the likelihood problem is

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{i=1}^n \ln \sum_{j=1}^k \frac{\pi_j}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^\intercal \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)\right).$$

  The $\ln$ and the $\exp$ are no longer next to each other; we can't just

Lloyd's method for k-means

Original k-means formulation

$$\phi((\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k)) = \sum_{i=1}^{n} \min_j \|\boldsymbol{x}_i - \boldsymbol{\mu}_j\|^2.$$

Lloyd's method for k-means

Original k-means formulation

$$\phi((\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k)) = \sum_{i=1}^{n} \min_j \|\boldsymbol{x}_i - \boldsymbol{\mu}_j\|^2.$$

To make an algorithm, we introduced *assignment matrix* $\boldsymbol{A} \in \mathcal{A}_{n,k}$:

$$\phi((\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k); \boldsymbol{A}) = \sum_{i=1}^{n} \sum_{j=1}^{k} A_{ij} \|\boldsymbol{x}_i - \boldsymbol{\mu}_j\|^2.$$

Lloyd's method for k-means

Original k-means formulation

$$\phi((\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k)) = \sum_{i=1}^{n} \min_j \|\boldsymbol{x}_i - \boldsymbol{\mu}_j\|^2.$$

To make an algorithm, we introduced *assignment matrix* $\boldsymbol{A} \in \mathcal{A}_{n,k}$:

$$\phi((\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k); \boldsymbol{A}) = \sum_{i=1}^{n} \sum_{j=1}^{k} A_{ij} \|\boldsymbol{x}_i - \boldsymbol{\mu}_j\|^2.$$

Let's do the same thing with Gaussians!

Gaussian mixture likelihood with *responsibility matrix* $\boldsymbol{R}$

Let's replace $\sum_{i=1}^{n} \ln \sum_{j=1}^{k} \pi_j p_{\mu_j, \boldsymbol{\Sigma}_j}(\boldsymbol{x}_i)$ with

$$\sum_{i=1}^{n} \sum_{j=1}^{k} R_{ij} \ln \left(\pi_j p_{\mu_j, \boldsymbol{\Sigma}_j}(\boldsymbol{x}_i) \right)$$

where $\boldsymbol{R} \in \mathcal{R}_{n,k} := \{\boldsymbol{R} \in [0,1]^{n \times k} : \boldsymbol{R}\mathbf{1}_k = \mathbf{1}_n\}$ is a **responsibility matrix**.

Gaussian mixture likelihood with *responsibility matrix* $\boldsymbol{R}$

Let's replace $\sum_{i=1}^n \ln \sum_{j=1}^k \pi_j p_{\mu_j, \boldsymbol{\Sigma}_j}(\boldsymbol{x}_i)$ with

$$\sum_{i=1}^n \sum_{j=1}^k R_{ij} \ln \left(\pi_j p_{\mu_j, \boldsymbol{\Sigma}_j}(\boldsymbol{x}_i)\right)$$

where $\boldsymbol{R} \in \mathcal{R}_{n,k} := \{\boldsymbol{R} \in [0,1]^{n \times k} : \boldsymbol{R}\mathbf{1}_k = \mathbf{1}_n\}$ is a **responsibility matrix**.

Holding R fixed and optimizing θ gives

$$\begin{aligned}
\pi_j &:= \frac{\sum_{i=1}^n R_{ij}}{\sum_{i=1}^n \sum_{l=1}^k R_{il}} = \frac{\sum_{i=1}^n R_{ij}}{n}; \\
\boldsymbol{\mu}_j &:= \frac{\sum_{i=1}^n R_{ij} \boldsymbol{x}_i}{\sum_{i=1}^n R_{ij}} = \frac{\sum_{i=1}^n R_{ij} \boldsymbol{x}_i}{n\pi_j}, \\
\boldsymbol{\Sigma}_j &:= \frac{\sum_{i=1}^n R_{ij} (\boldsymbol{x}_i - \boldsymbol{\mu}_j)(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^\intercal}{n\pi_j}.
\end{aligned}$$

(Should use new mean in $\boldsymbol{\Sigma}_j$ so that all deriviatives 0.)

Updating $\boldsymbol{\mu}_j$

Recall our new likelihood with responsibilities $\boldsymbol{R}$:

$$\sum_{i=1}^{n} \sum_{j=1}^{k} R_{ij} \ln \pi_j p_{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j}(\boldsymbol{x}_i)$$

(In the literature, this quantity is **"expected complete data likelihood"**.)

Updating $\boldsymbol{\mu}_j$

Recall our new likelihood with responsibilities $\boldsymbol{R}$:

$$\sum_{i=1}^{n}\sum_{j=1}^{k} R_{ij} \ln \pi_j p_{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j}(\boldsymbol{x}_i)$$

(In the literature, this quantity is **"expected complete data likelihood"**.)

Taking derivative and setting to 0:

$$0 = \sum_{i=1}^{n} R_{ij} \nabla_{\boldsymbol{\mu}_j} \left(\ln \exp(-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^\intercal \boldsymbol{\Sigma}_j^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)) + \text{terms w/o } \boldsymbol{\mu}_j \right)$$

$$= \sum_{i=1}^{n} R_{ij} \boldsymbol{\Sigma}_j^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_j).$$

Rearranging, $\boldsymbol{\mu}_j = \frac{\sum_{i=1}^{n} R_{ij}\boldsymbol{x}_i}{n\pi_j}$.

Updating $\boldsymbol{\pi}$

Recall our new likelihood with responsibilities $\boldsymbol{R}$:

$$\sum_{i=1}^{n} \sum_{j=1}^{k} R_{ij} \ln \pi_j p_{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j}(\boldsymbol{x}_i)$$

Updating $\boldsymbol{\pi}$

Recall our new likelihood with responsibilities $\boldsymbol{R}$:

$$\sum_{i=1}^{n}\sum_{j=1}^{k} R_{ij} \ln \pi_j p_{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j}(\boldsymbol{x}_i)$$

Taking derivative and setting to 0:

$$0 = \sum_{i=1}^{n} \frac{R_{ij}}{\pi_j};$$

oops?

Updating $\boldsymbol{\pi}$

Recall our new likelihood with responsibilities $\boldsymbol{R}$:

$$\sum_{i=1}^{n}\sum_{j=1}^{k} R_{ij} \ln \pi_j p_{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j}(\boldsymbol{x}_i)$$

Taking derivative and setting to 0:

$$0 = \sum_{i=1}^{n} \frac{R_{ij}}{\pi_j};$$

oops?

Fix: we forgot the constraints on $\boldsymbol{\pi}$!

Updating $\boldsymbol{\pi}$

Include constraint $\sum_{j=1}^{k} \pi_j = 1$ with a Lagrangian:

$$\sum_{i=1}^{n} \sum_{j=1}^{k} R_{ij} \ln \pi_j p_{\mu_j, \boldsymbol{\Sigma}_j}(\boldsymbol{x}_i) + \lambda \left(1 - \sum_{j=1}^{k} \pi_j\right)$$

Updating $\boldsymbol{\pi}$

Include constraint $\sum_{j=1}^{k} \pi_j = 1$ with a Lagrangian:

$$\sum_{i=1}^{n} \sum_{j=1}^{k} R_{ij} \ln \pi_j p_{\mu_j, \boldsymbol{\Sigma}_j}(\boldsymbol{x}_i) + \lambda \left(1 - \sum_{j=1}^{k} \pi_j\right)$$

Differentiating and setting this Lagrangian to 0, we get

$$\lambda = \sum_{i=1}^{n} \frac{R_{ij}}{\pi_j}, \qquad \text{and} \qquad \sum_{j} \pi_j = 1.$$

Updating $\boldsymbol{\pi}$

Include constraint $\sum_{j=1}^{k} \pi_j = 1$ with a Lagrangian:

$$\sum_{i=1}^{n} \sum_{j=1}^{k} R_{ij} \ln \pi_j p_{\mu_j, \boldsymbol{\Sigma}_j}(\boldsymbol{x}_i) + \lambda \left(1 - \sum_{j=1}^{k} \pi_j\right)$$

Differentiating and setting this Lagrangian to 0, we get

$$\lambda = \sum_{i=1}^{n} \frac{R_{ij}}{\pi_j}, \qquad \text{and} \qquad \sum_{j} \pi_j = 1.$$

Together, $\pi_j = \sum_{i=1}^{n} R_{ij}/\lambda$, and

$$1 = \sum_{j=1}^{k} \pi_j = \sum_{j=1}^{k} \sum_{i=1}^{n} \frac{R_{ij}}{\lambda} = \frac{n}{\lambda},$$

so $\lambda = n$ and $\pi_j = \sum_{i=1}^{n} R_{ij}/n$.

Updating $\boldsymbol{\Sigma}_j$

Starting again from likelihood with responsibilities $\boldsymbol{R}$:

$$\sum_{i=1}^{n} \sum_{j=1}^{k} R_{ij} \ln \pi_j p_{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j}(\boldsymbol{x}_i).$$

Updating $\boldsymbol{\Sigma}_j$

Starting again from likelihood with responsibilities $\boldsymbol{R}$:

$$\sum_{i=1}^{n}\sum_{j=1}^{k} R_{ij} \ln \pi_j p_{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j}(\boldsymbol{x}_i).$$

Taking derivative and setting to 0,

$$0 = \sum_{i=1}^{n} R_{ij} \nabla_{\boldsymbol{\Sigma}_j} \left(-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^\intercal \boldsymbol{\Sigma}_j^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_j) - \frac{1}{2}\ln|\boldsymbol{\Sigma}_j| + \text{other stuff} \right).$$

Updating $\boldsymbol{\Sigma}_j$

Starting again from likelihood with responsibilities $\boldsymbol{R}$:

$$\sum_{i=1}^{n}\sum_{j=1}^{k} R_{ij} \ln \pi_j p_{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j}(\boldsymbol{x}_i).$$

Taking derivative and setting to 0,

$$0 = \sum_{i=1}^{n} R_{ij} \nabla_{\boldsymbol{\Sigma}_j} \left(-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^\intercal \boldsymbol{\Sigma}_j^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_j) - \frac{1}{2}\ln|\boldsymbol{\Sigma}_j| + \text{other stuff} \right).$$

By magic matrix derivative rules,
$\boldsymbol{\Sigma}_j^{-1} = \sum_{i=1}^{n} R_{ij}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^\intercal / (n\pi_j)$.

Summary of $\boldsymbol{\theta}$ optimization

Replace $\sum_{i=1}^n \ln \sum_{j=1}^k \pi_j p_{\mu_j, \boldsymbol{\Sigma}_j}(\boldsymbol{x}_i)$ with

$$\sum_{i=1}^n \sum_{j=1}^k R_{ij} \ln \left(\pi_j p_{\mu_j, \boldsymbol{\Sigma}_j}(\boldsymbol{x}_i) \right).$$

Hold R fixed and optimize θ:

$$\pi_j := \frac{\sum_{i=1}^n R_{ij}}{\sum_{i=1}^n \sum_{l=1}^k R_{il}} = \frac{\sum_{i=1}^n R_{ij}}{n};$$

$$\boldsymbol{\mu}_j := \frac{\sum_{i=1}^n R_{ij} \boldsymbol{x}_i}{\sum_{i=1}^n R_{ij}} = \frac{\sum_{i=1}^n R_{ij} \boldsymbol{x}_i}{n \pi_j},$$

$$\boldsymbol{\Sigma}_j := \frac{\sum_{i=1}^n R_{ij} (\boldsymbol{x}_i - \boldsymbol{\mu}_j)(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^\intercal}{n \pi_j}.$$

Summary of $\boldsymbol{\theta}$ optimization

Replace $\sum_{i=1}^n \ln \sum_{j=1}^k \pi_j p_{\mu_j, \boldsymbol{\Sigma}_j}(\boldsymbol{x}_i)$ with

$$\sum_{i=1}^n \sum_{j=1}^k R_{ij} \ln \left(\pi_j p_{\mu_j, \boldsymbol{\Sigma}_j}(\boldsymbol{x}_i) \right).$$

Hold R fixed and optimize θ:

$$\pi_j := \frac{\sum_{i=1}^n R_{ij}}{\sum_{i=1}^n \sum_{l=1}^k R_{il}} = \frac{\sum_{i=1}^n R_{ij}}{n};$$

$$\boldsymbol{\mu}_j := \frac{\sum_{i=1}^n R_{ij} \boldsymbol{x}_i}{\sum_{i=1}^n R_{ij}} = \frac{\sum_{i=1}^n R_{ij} \boldsymbol{x}_i}{n \pi_j},$$

$$\boldsymbol{\Sigma}_j := \frac{\sum_{i=1}^n R_{ij} (\boldsymbol{x}_i - \boldsymbol{\mu}_j)(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^\intercal}{n \pi_j}.$$

How to optimize R_{ij}?

- Likelihood lacks the $\min_j$ from the k-means cost.
- We'll now develop the E-M method, which picks $\boldsymbol{R}$ in a way that guarantees **likelihood** increases.

E-M (Expectation-Maximization)

Generalizing the assignment matrix to GMMs

We introduced an **assigment matrix** $\boldsymbol{A} \in \{0,1\}^{n \times k}$:

- For each $\boldsymbol{x}_i$, define $\boldsymbol{\mu}(\boldsymbol{x}_i)$ to be a closest center:

$$\|\boldsymbol{x}_i - \boldsymbol{\mu}(\boldsymbol{x}_i)\| = \min_j \|\boldsymbol{x}_i - \boldsymbol{\mu}_j\|.$$

- For each i, set $A_{ij} = \mathbb{1}[\boldsymbol{\mu}(\boldsymbol{x}_i) = \boldsymbol{\mu}_j]$.

Generalizing the assignment matrix to GMMs

We introduced an **assigment matrix** $\boldsymbol{A} \in \{0,1\}^{n\times k}$:

- For each $\boldsymbol{x}_i$, define $\boldsymbol{\mu}(\boldsymbol{x}_i)$ to be a closest center:

$$\|\boldsymbol{x}_i - \boldsymbol{\mu}(\boldsymbol{x}_i)\| = \min_j \|\boldsymbol{x}_i - \boldsymbol{\mu}_j\|.$$

- For each i, set $A_{ij} = \mathbb{1}[\boldsymbol{\mu}(\boldsymbol{x}_i) = \boldsymbol{\mu}_j]$.

- **Key property:** by this choice,

$$\phi(\boldsymbol{C};\boldsymbol{A}) = \sum_{i=1}^{n}\sum_{j=1}^{k} A_{ij}\|\boldsymbol{x}_i - \boldsymbol{\mu}_j\|^2 = \sum_{i=1}^{n}\min_j \|\boldsymbol{x}_i - \boldsymbol{\mu}_j\|^2 = \phi(\boldsymbol{C});$$

therefore we can decrase $\phi(\boldsymbol{C}) = \phi(\boldsymbol{C};\boldsymbol{A})$
first by optimizing $\boldsymbol{C}$ to get $\phi(\boldsymbol{C}';\boldsymbol{A}) \le \phi(\boldsymbol{C};\boldsymbol{A})$,
then setting $\boldsymbol{A}$ as above to get

$$\phi(\boldsymbol{C}') = \phi(\boldsymbol{C}';\boldsymbol{A}') \le \phi(\boldsymbol{C}';\boldsymbol{A}) \le \phi(\boldsymbol{C};\boldsymbol{A}) = \phi(\boldsymbol{C}).$$

In other words: we minimize $\phi(\boldsymbol{C})$ via $\phi(\boldsymbol{C};\boldsymbol{A})$.

Generalizing the assignment matrix to GMMs

We introduced an **assigment matrix** $\boldsymbol{A} \in \{0,1\}^{n\times k}$:

- For each $\boldsymbol{x}_i$, define $\boldsymbol{\mu}(\boldsymbol{x}_i)$ to be a closest center:

$$\|\boldsymbol{x}_i - \boldsymbol{\mu}(\boldsymbol{x}_i)\| = \min_j \|\boldsymbol{x}_i - \boldsymbol{\mu}_j\|.$$

- For each i, set $A_{ij} = \mathbb{1}[\boldsymbol{\mu}(\boldsymbol{x}_i) = \boldsymbol{\mu}_j]$.

- **Key property:** by this choice,

$$\phi(\boldsymbol{C};\boldsymbol{A}) = \sum_{i=1}^{n}\sum_{j=1}^{k} A_{ij}\|\boldsymbol{x}_i - \boldsymbol{\mu}_j\|^2 = \sum_{i=1}^{n}\min_j \|\boldsymbol{x}_i - \boldsymbol{\mu}_j\|^2 = \phi(\boldsymbol{C});$$

therefore we can decrase $\phi(\boldsymbol{C}) = \phi(\boldsymbol{C};\boldsymbol{A})$
first by optimizing $\boldsymbol{C}$ to get $\phi(\boldsymbol{C}';\boldsymbol{A}) \le \phi(\boldsymbol{C};\boldsymbol{A})$,
then setting $\boldsymbol{A}$ as above to get

$$\phi(\boldsymbol{C}') = \phi(\boldsymbol{C}';\boldsymbol{A}') \le \phi(\boldsymbol{C}';\boldsymbol{A}) \le \phi(\boldsymbol{C};\boldsymbol{A}) = \phi(\boldsymbol{C}).$$

In other words: we minimize $\phi(\boldsymbol{C})$ via $\phi(\boldsymbol{C};\boldsymbol{A})$.

What fulfills the same role for $\mathcal{L}$?

Latent variable models.

Since $1 = \sum_{j=1}^{k} p_{\boldsymbol{\theta}}(y_i = j|\boldsymbol{x}_i)$ and $p_{\boldsymbol{\theta}}(y_i = j|\boldsymbol{x}_i) = \dfrac{p_{\boldsymbol{\theta}}(y_i = j, \boldsymbol{x}_i)}{p_{\boldsymbol{\theta}}(\boldsymbol{x}_i)}$, then

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{i=1}^{n} \ln p_{\boldsymbol{\theta}}(x_i) = \sum_{i=1}^{n} 1 \cdot \ln p_{\boldsymbol{\theta}}(x_i) = \sum_{i=1}^{n} \sum_{j=1}^{k} p_{\boldsymbol{\theta}}(y_i = j|\boldsymbol{x}_i) \ln p_{\boldsymbol{\theta}}(\boldsymbol{x}_i)$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{k} p_{\boldsymbol{\theta}}(y_i = j|\boldsymbol{x}_i) \ln \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_i, y_i = j)}{p_{\boldsymbol{\theta}}(y_i = j|\boldsymbol{x}_i)}.$$

Latent variable models.

Since $1 = \sum_{j=1}^{k} p_{\boldsymbol{\theta}}(y_i = j|\boldsymbol{x}_i)$ and $p_{\boldsymbol{\theta}}(y_i = j|\boldsymbol{x}_i) = \dfrac{p_{\boldsymbol{\theta}}(y_i = j, \boldsymbol{x}_i)}{p_{\boldsymbol{\theta}}(\boldsymbol{x}_i)}$, then

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{i=1}^{n} \ln p_{\boldsymbol{\theta}}(x_i) = \sum_{i=1}^{n} 1 \cdot \ln p_{\boldsymbol{\theta}}(x_i) = \sum_{i=1}^{n}\sum_{j=1}^{k} p_{\boldsymbol{\theta}}(y_i = j|\boldsymbol{x}_i) \ln p_{\boldsymbol{\theta}}(\boldsymbol{x}_i)$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{k} p_{\boldsymbol{\theta}}(y_i = j|\boldsymbol{x}_i) \ln \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_i, y_i = j)}{p_{\boldsymbol{\theta}}(y_i = j|\boldsymbol{x}_i)}.$$

Therefore: define **augmented likelihood**

$$\mathcal{L}(\boldsymbol{\theta}; \boldsymbol{R}) := \sum_{i=1}^{n}\sum_{j=1}^{k} R_{ij} \ln \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_i, y_i = j)}{R_{ij}};$$

note that $R_{ij} := p_{\boldsymbol{\theta}}(y_i = j|\boldsymbol{x}_i)$ implies $\mathcal{L}(\boldsymbol{\theta}; \boldsymbol{R}) = \mathcal{L}(\boldsymbol{\theta})$.

E-M method for latent variable models

Define **augmented likelihood**

$$\mathcal{L}(\boldsymbol{\theta}; \boldsymbol{R}) := \sum_{i=1}^{n} \sum_{j=1}^{k} R_{ij} \ln \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_i, y_i = j)}{R_{ij}},$$

with **responsibility matrix** $\boldsymbol{R} \in \mathcal{R}_{n,k} := \{\boldsymbol{R} \in [0,1]^{n \times k} : \boldsymbol{R}\mathbf{1}_k = \mathbf{1}_n\}$.

Alternate two steps:

- E-step: set $(\boldsymbol{R}_t)_{ij} := p_{\boldsymbol{\theta}_{t-1}}(y_i = j | \boldsymbol{x}_i)$.
- M-step: set $\boldsymbol{\theta}_t = \arg\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathcal{L}(\boldsymbol{\theta}; \boldsymbol{R}_t)$.

E-M method for latent variable models

Define **augmented likelihood**

$$\mathcal{L}(\boldsymbol{\theta}; \boldsymbol{R}) := \sum_{i=1}^{n} \sum_{j=1}^{k} R_{ij} \ln \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_i, y_i = j)}{R_{ij}},$$

with **responsibility matrix** $\boldsymbol{R} \in \mathcal{R}_{n,k} := \{\boldsymbol{R} \in [0,1]^{n \times k} : \boldsymbol{R}\mathbf{1}_k = \mathbf{1}_n\}$.

Alternate two steps:

- E-step: set $(\boldsymbol{R}_t)_{ij} := p_{\boldsymbol{\theta}_{t-1}}(y_i = j | \boldsymbol{x}_i)$.
- M-step: set $\boldsymbol{\theta}_t = \arg\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathcal{L}(\boldsymbol{\theta}; \boldsymbol{R}_t)$.

Soon: we'll see this gives nondecreasing likelihood!

E-M for Gaussian mixtures

Initialization: a standard choice is $\pi_j = 1/k$, $\boldsymbol{\Sigma}_j = \boldsymbol{I}$, and $(\mu_j)_{j=1}^k$ given by k-means.

- **E-step:** Set $R_{ij} = p_{\boldsymbol{\theta}}(y_i = j | \boldsymbol{x}_i)$, meaning

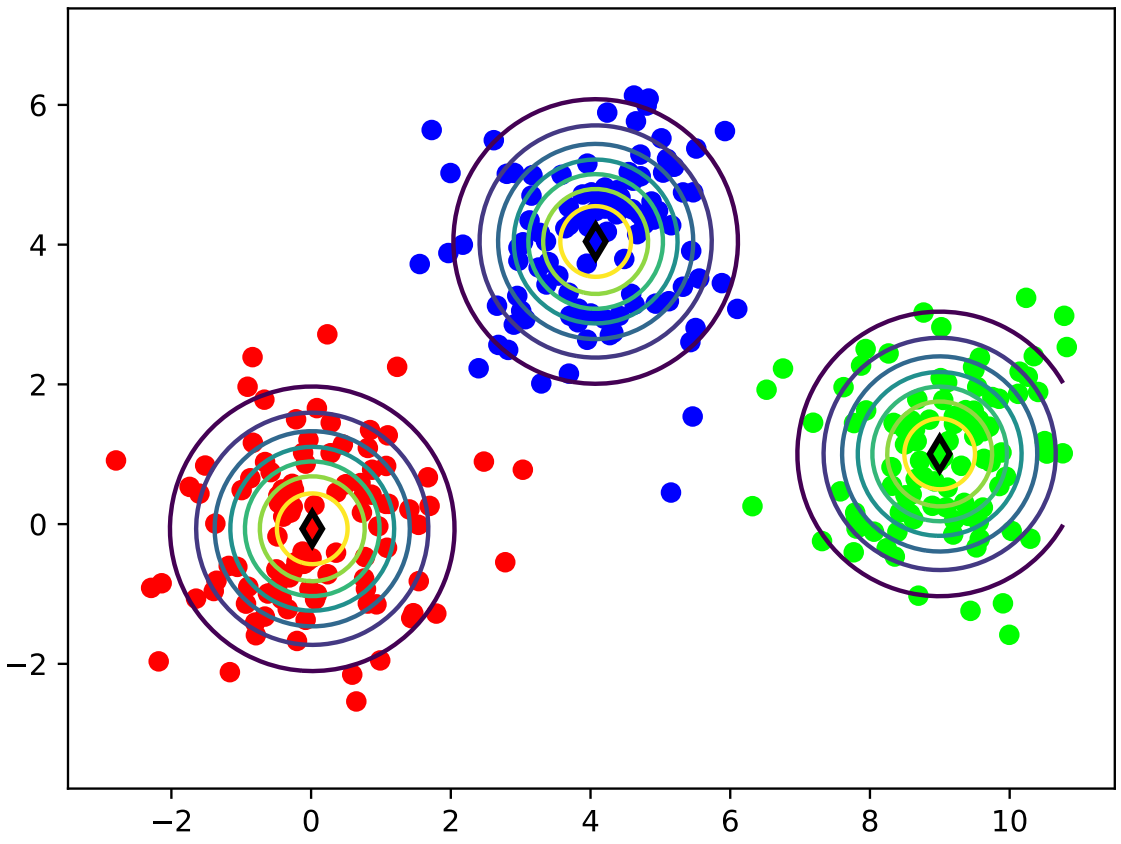
$$R_{ij} = p_{\boldsymbol{\theta}}(y_i = j | \boldsymbol{x}_i) = \frac{p_{\boldsymbol{\theta}}(y_i = j, \boldsymbol{x}_i)}{p_{\boldsymbol{\theta}}(\boldsymbol{x}_i)} = \frac{\pi_j p_{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j}(\boldsymbol{x}_i)}{\sum_{l=1}^k \pi_l p_{\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l}(\boldsymbol{x}_i)}.$$

- **M-step:** solve $\arg\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathcal{L}(\boldsymbol{\theta}; \boldsymbol{R})$, meaning

$$\begin{aligned}
\pi_j &:= \frac{\sum_{i=1}^n R_{ij}}{\sum_{i=1}^n \sum_{l=1}^k R_{il}} = \frac{\sum_{i=1}^n R_{ij}}{n}, \\
\boldsymbol{\mu}_j &:= \frac{\sum_{i=1}^n R_{ij} \boldsymbol{x}_i}{\sum_{i=1}^n R_{ij}} = \frac{\sum_{i=1}^n R_{ij} \boldsymbol{x}_i}{n \pi_j}, \\
\boldsymbol{\Sigma}_j &:= \frac{\sum_{i=1}^n R_{ij} (\boldsymbol{x}_i - \boldsymbol{\mu}_j)(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^\intercal}{n \pi_j}.
\end{aligned}$$

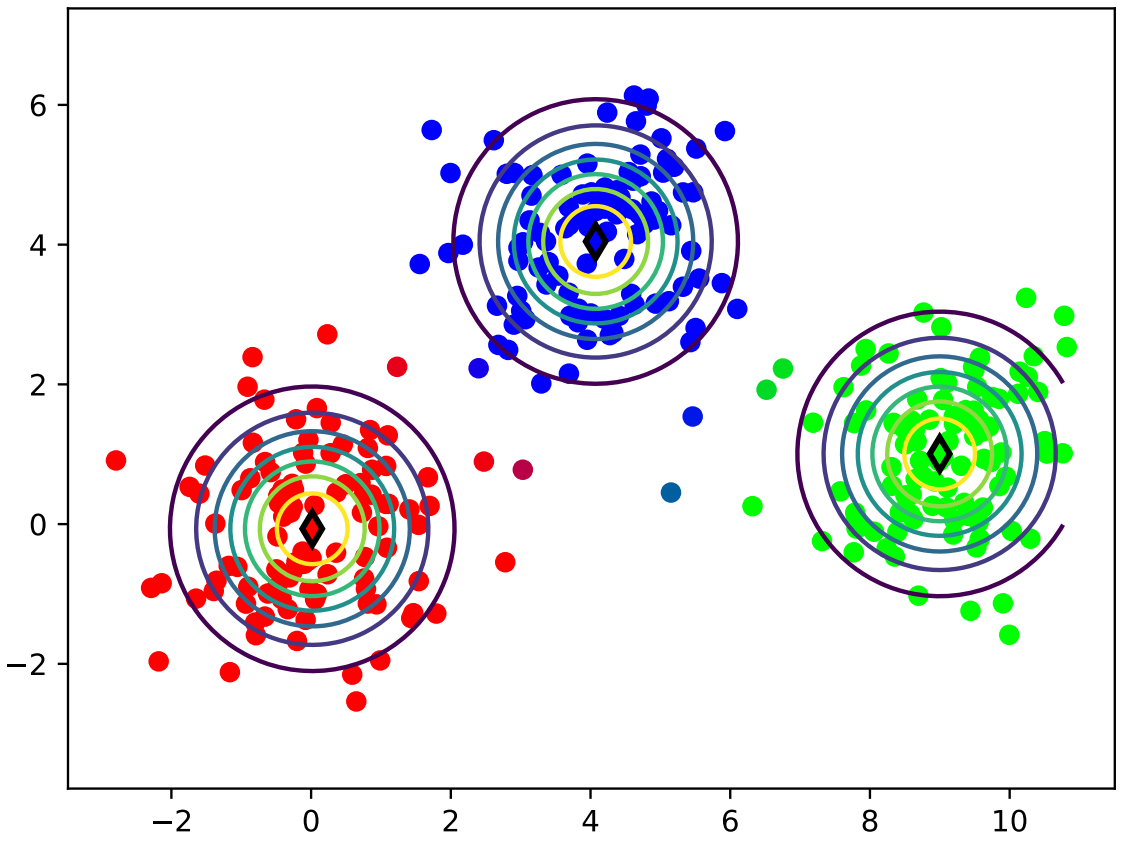
(These are as before.)

Demo: spherical clusters



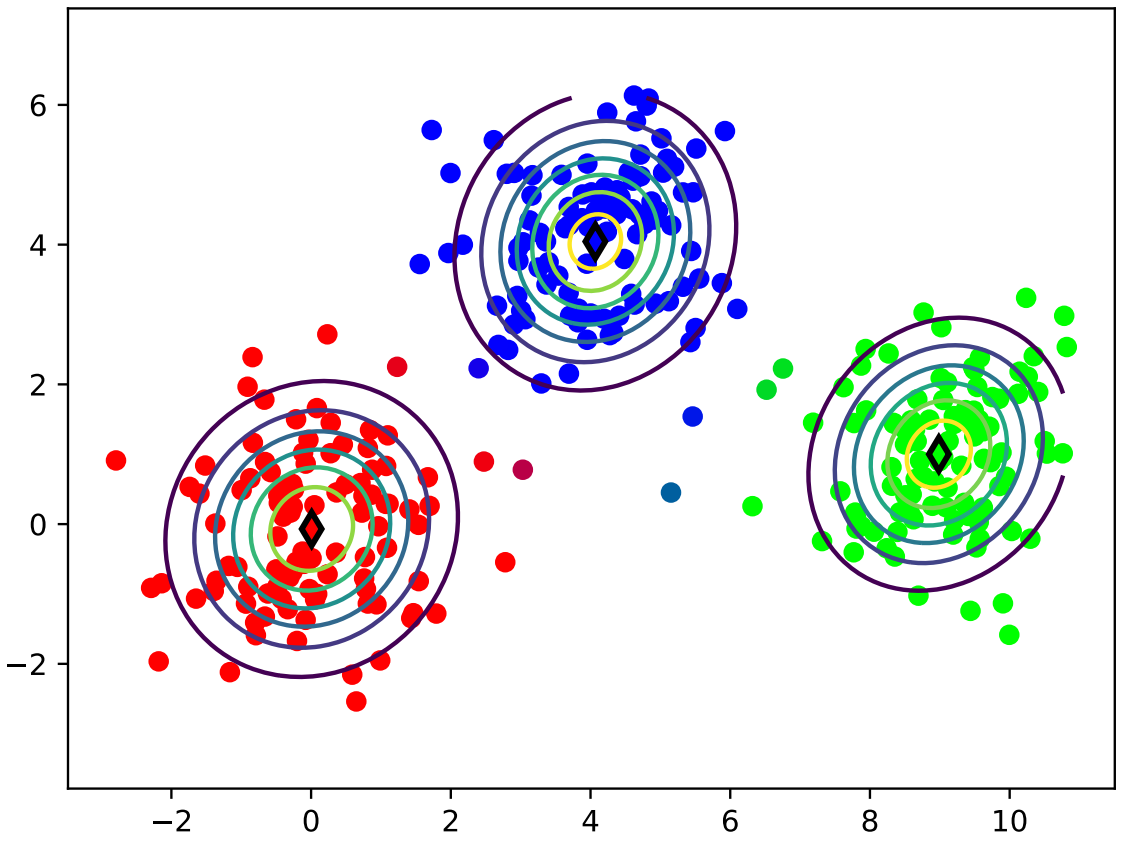
(Initialized with k-means, thus not so dramatic.)

Demo: spherical clusters



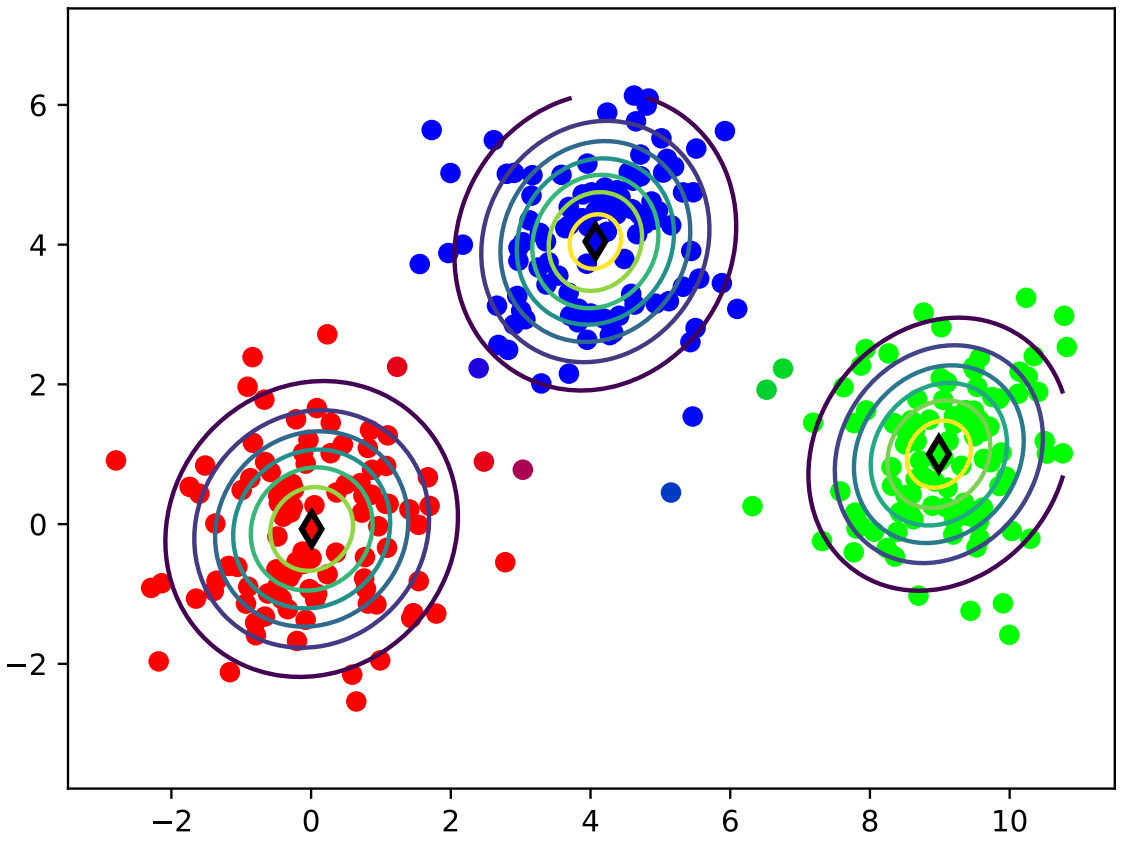
(Initialized with k-means, thus not so dramatic.)

Demo: spherical clusters



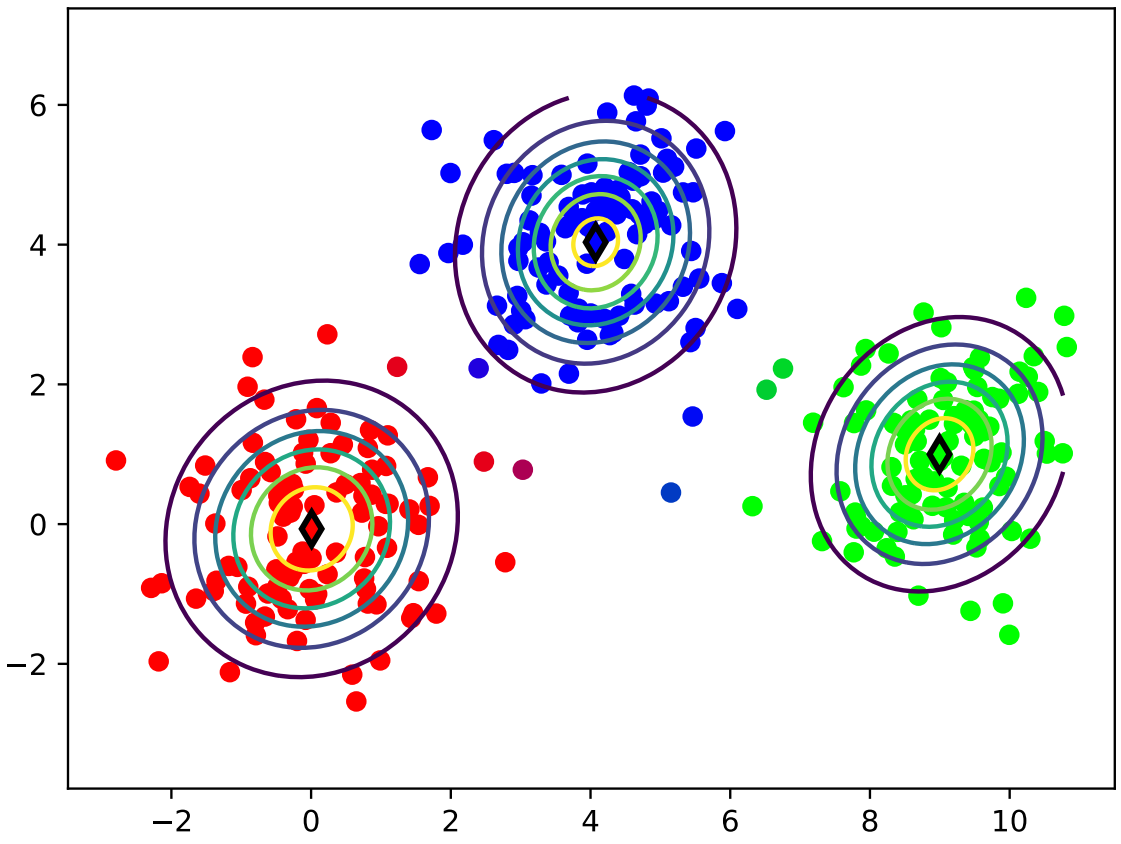
(Initialized with k-means, thus not so dramatic.)

Demo: spherical clusters



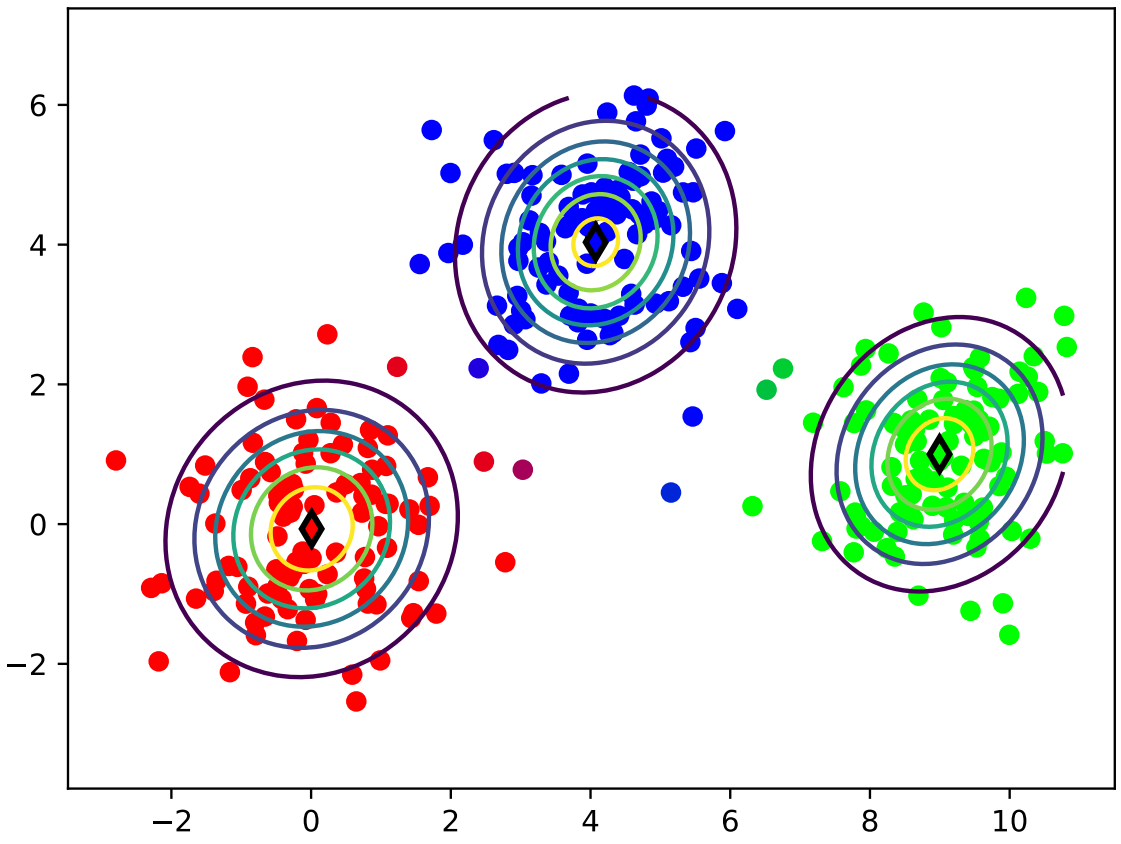
(Initialized with k-means, thus not so dramatic.)

Demo: spherical clusters



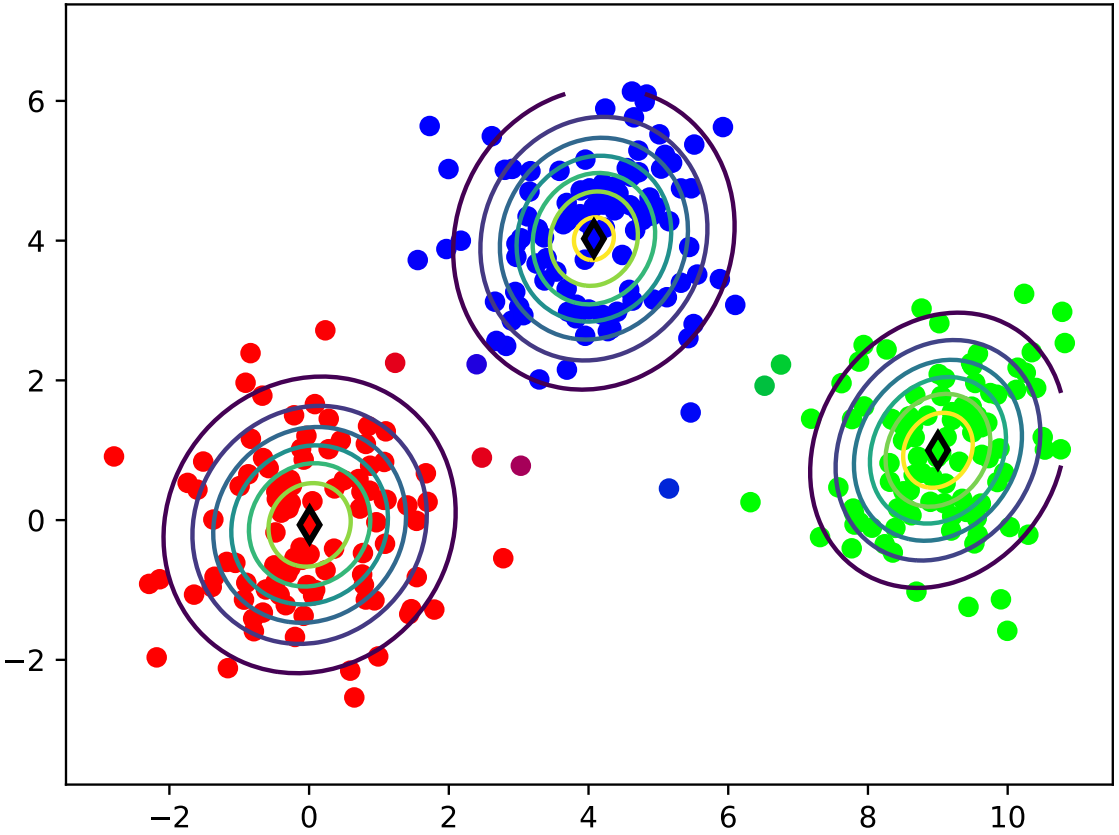
(Initialized with k-means, thus not so dramatic.)

Demo: spherical clusters



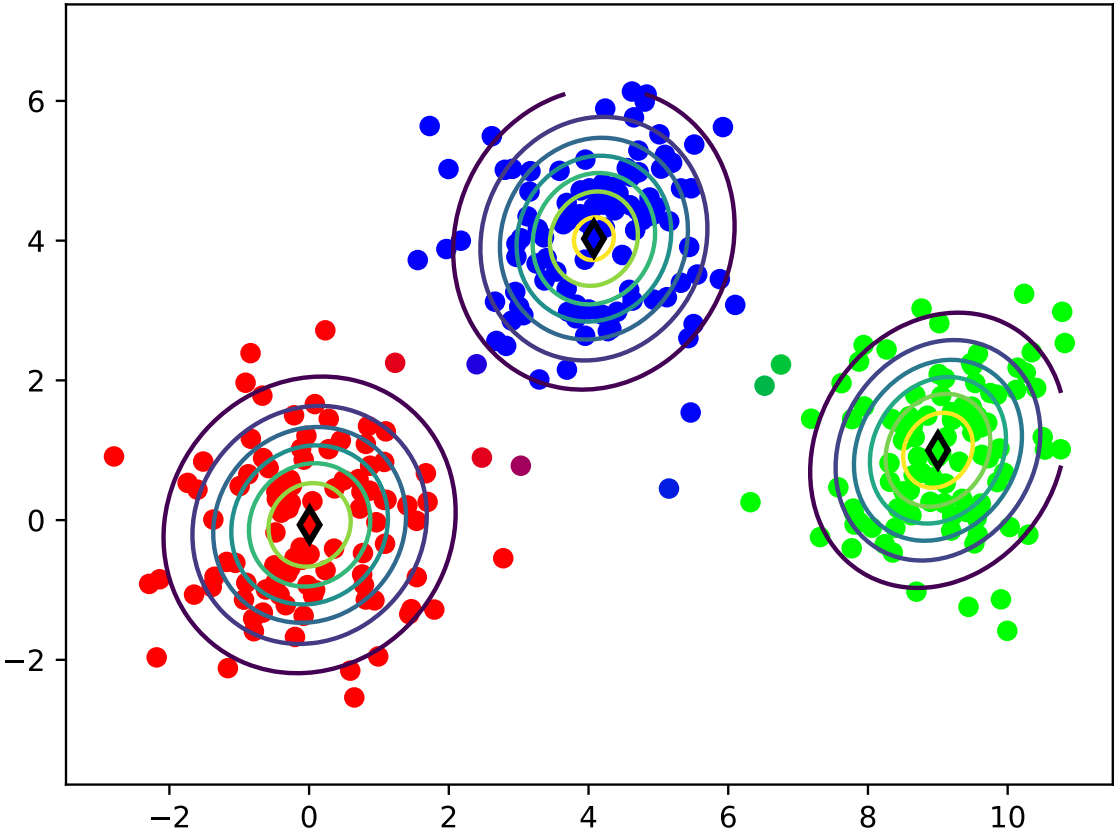
(Initialized with k-means, thus not so dramatic.)

Demo: spherical clusters



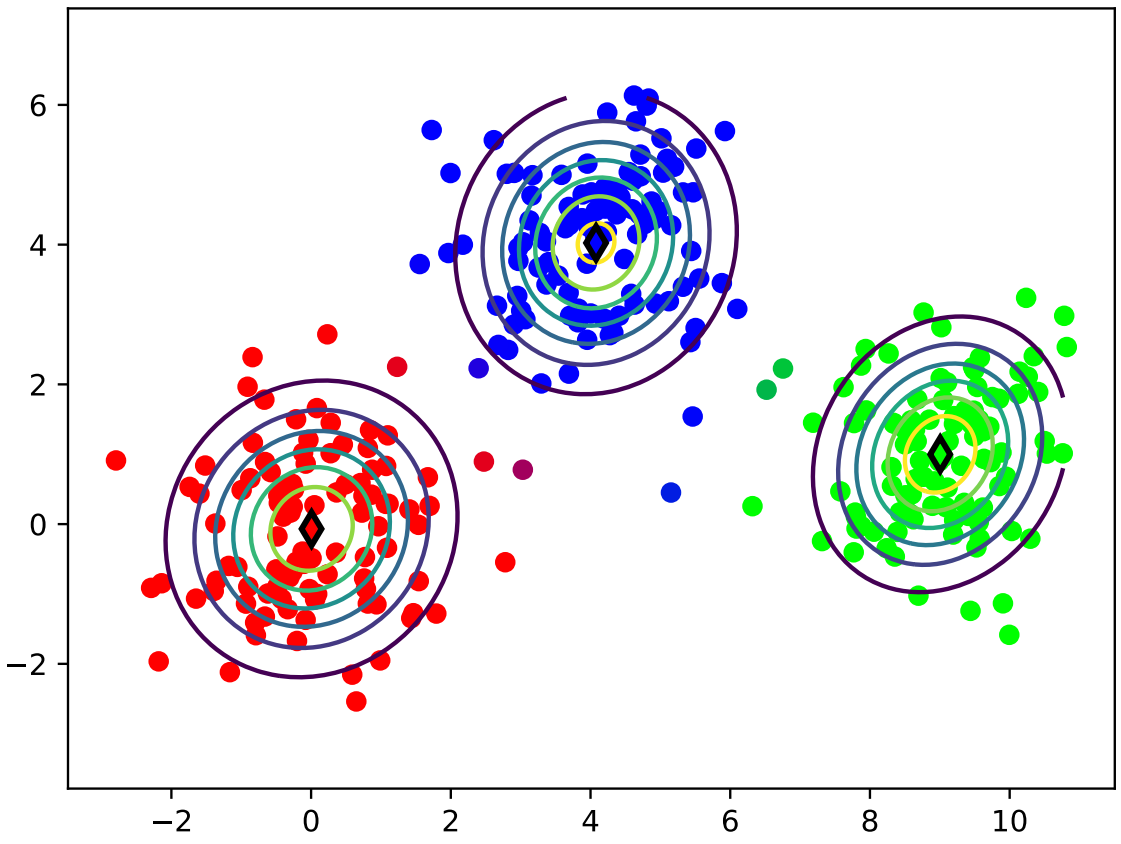
(Initialized with k-means, thus not so dramatic.)

Demo: spherical clusters



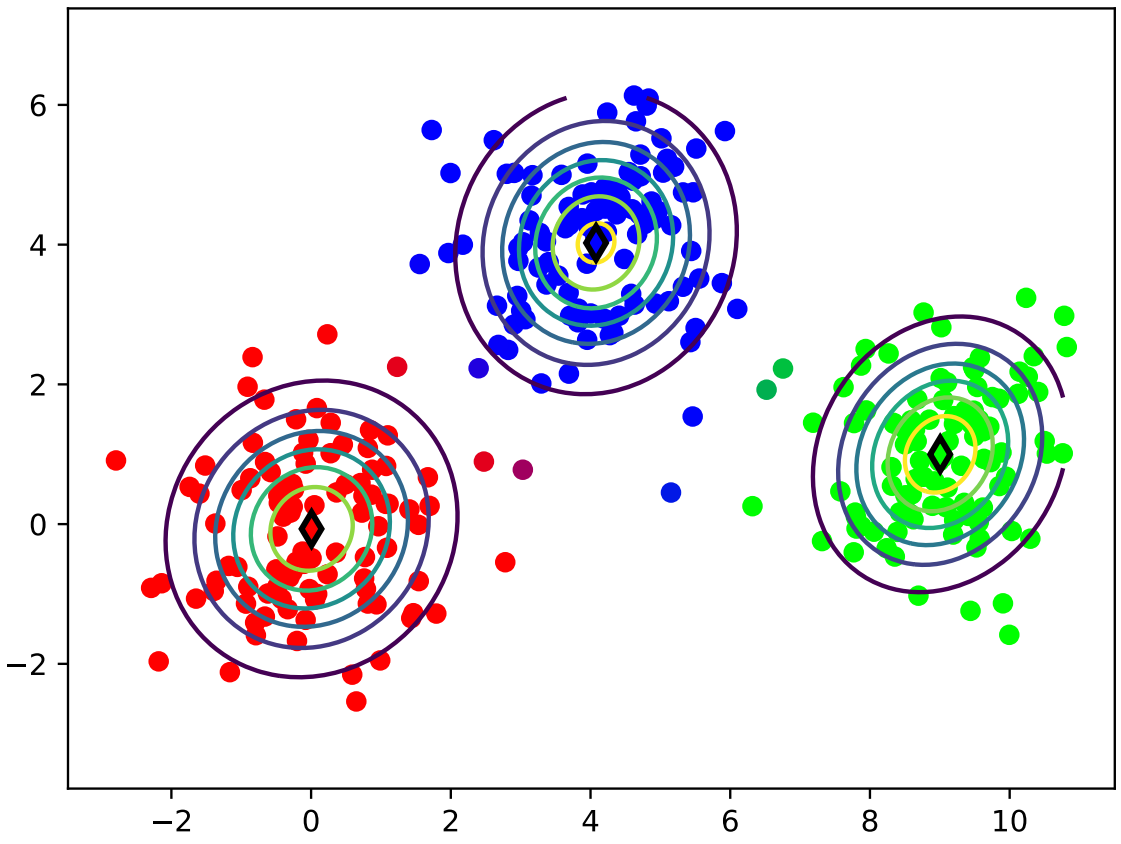
(Initialized with k-means, thus not so dramatic.)

Demo: spherical clusters



(Initialized with k-means, thus not so dramatic.)

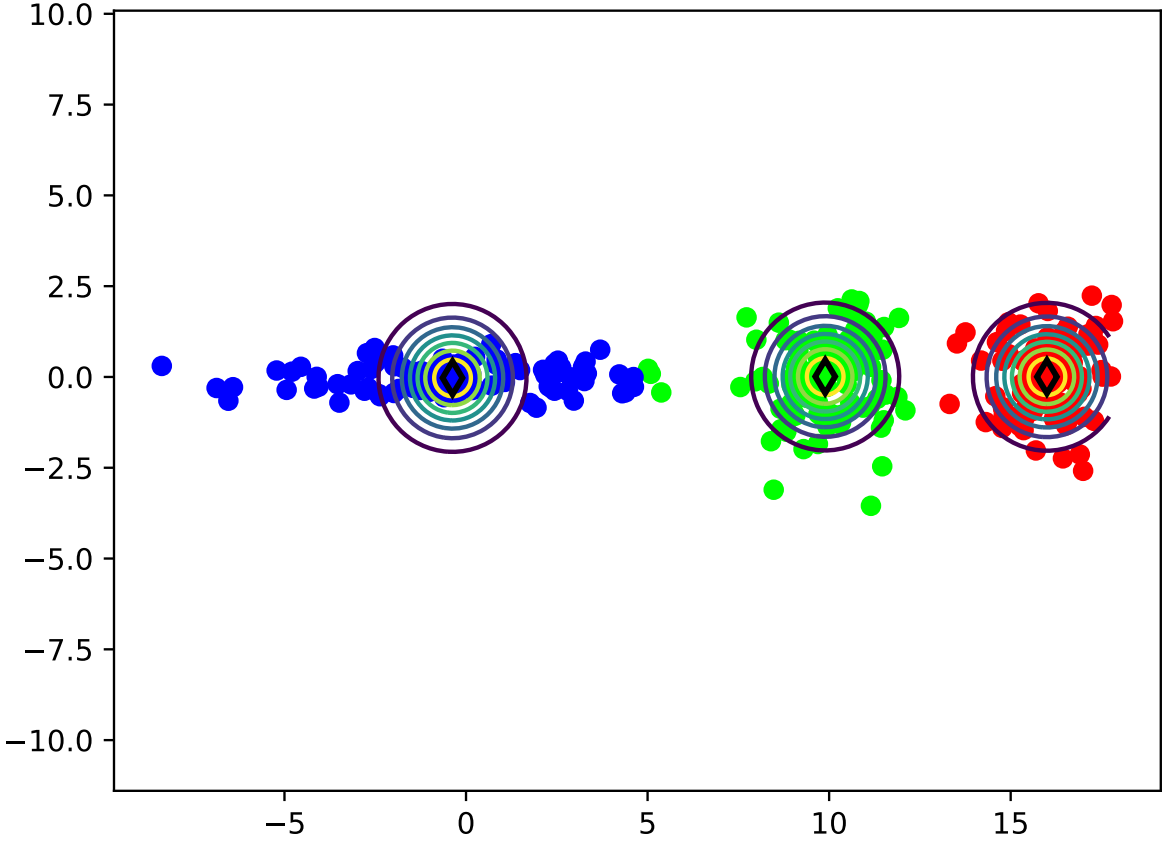
Demo: spherical clusters



(Initialized with k-means, thus not so dramatic.)

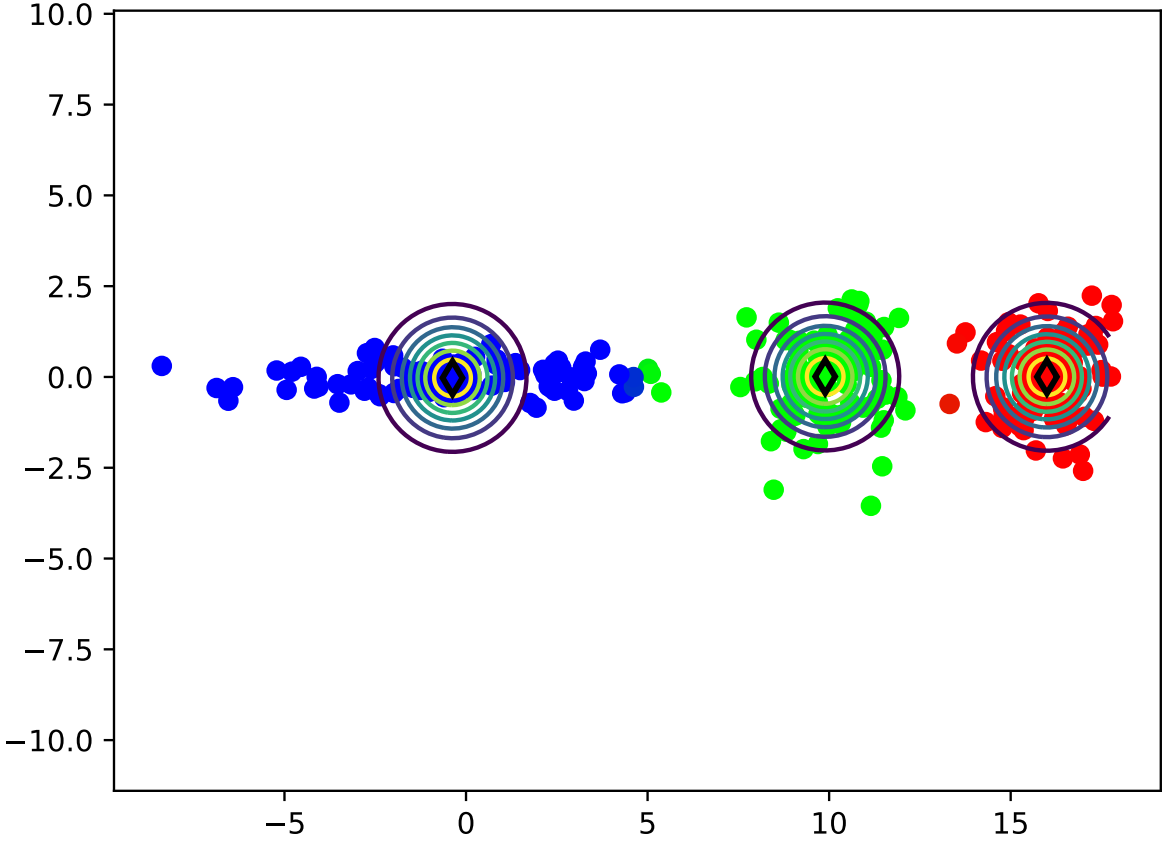
Demo: elliptical clusters

E. . .



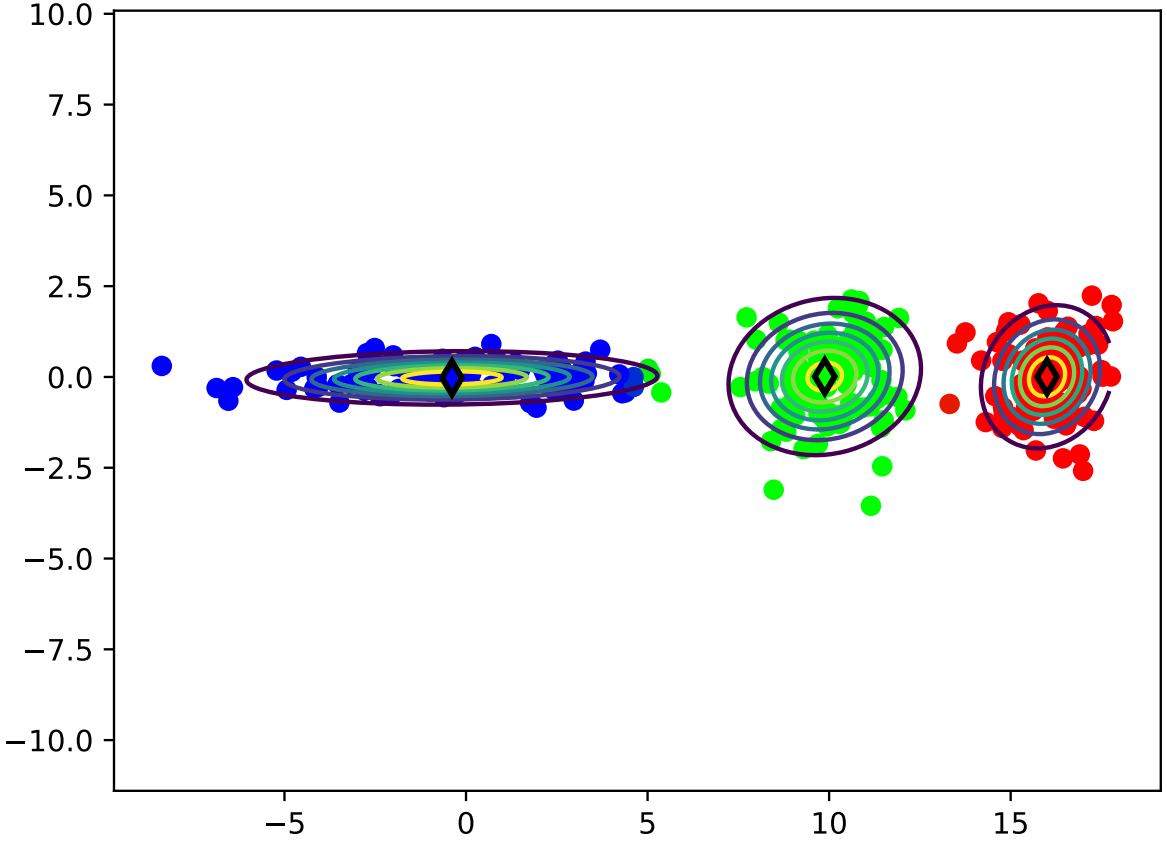
Demo: elliptical clusters

E... M...



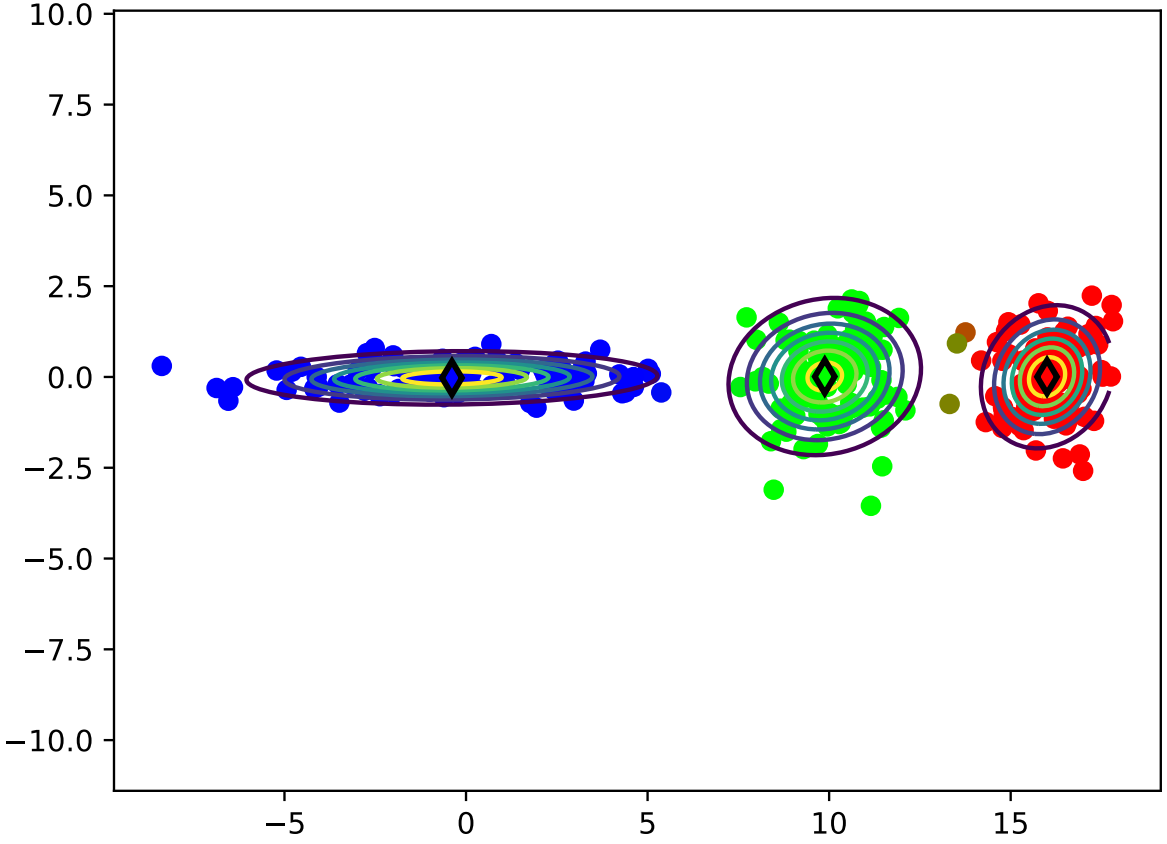
Demo: elliptical clusters

E... M... E...



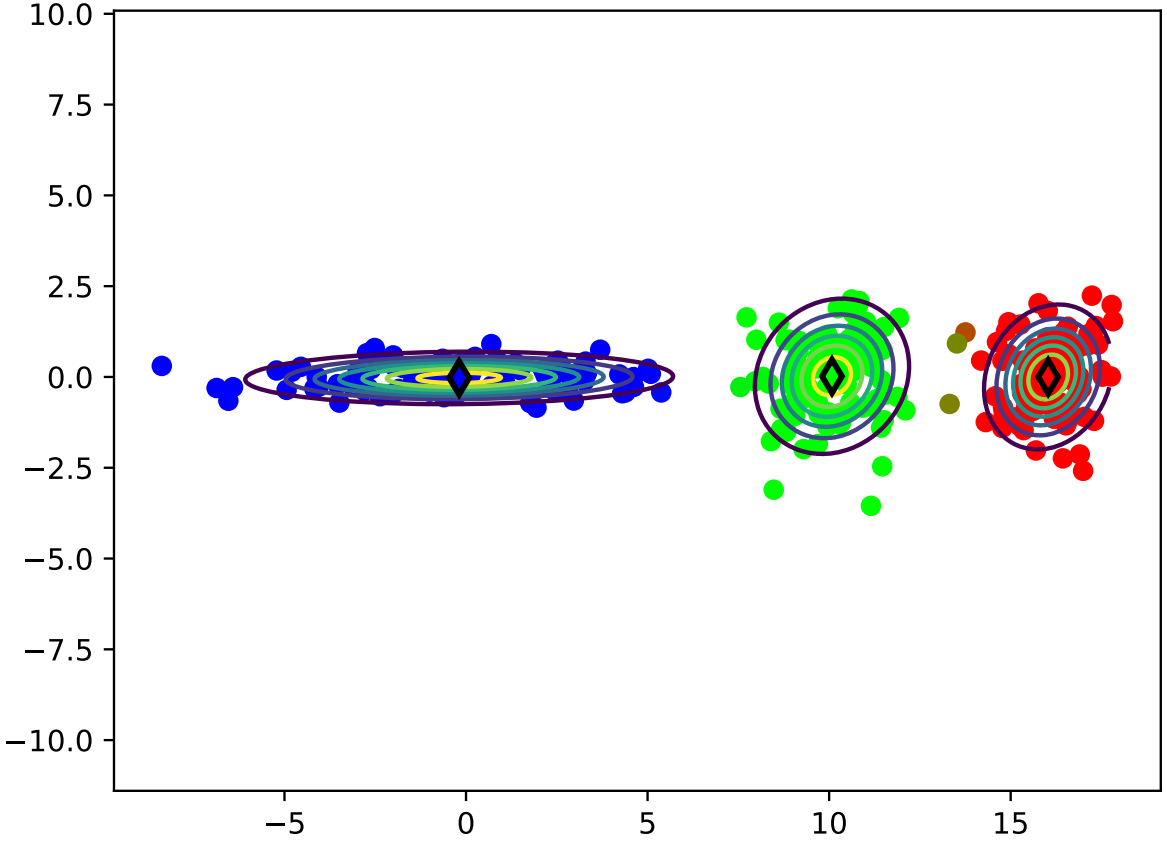
Demo: elliptical clusters

E... M... E... M...



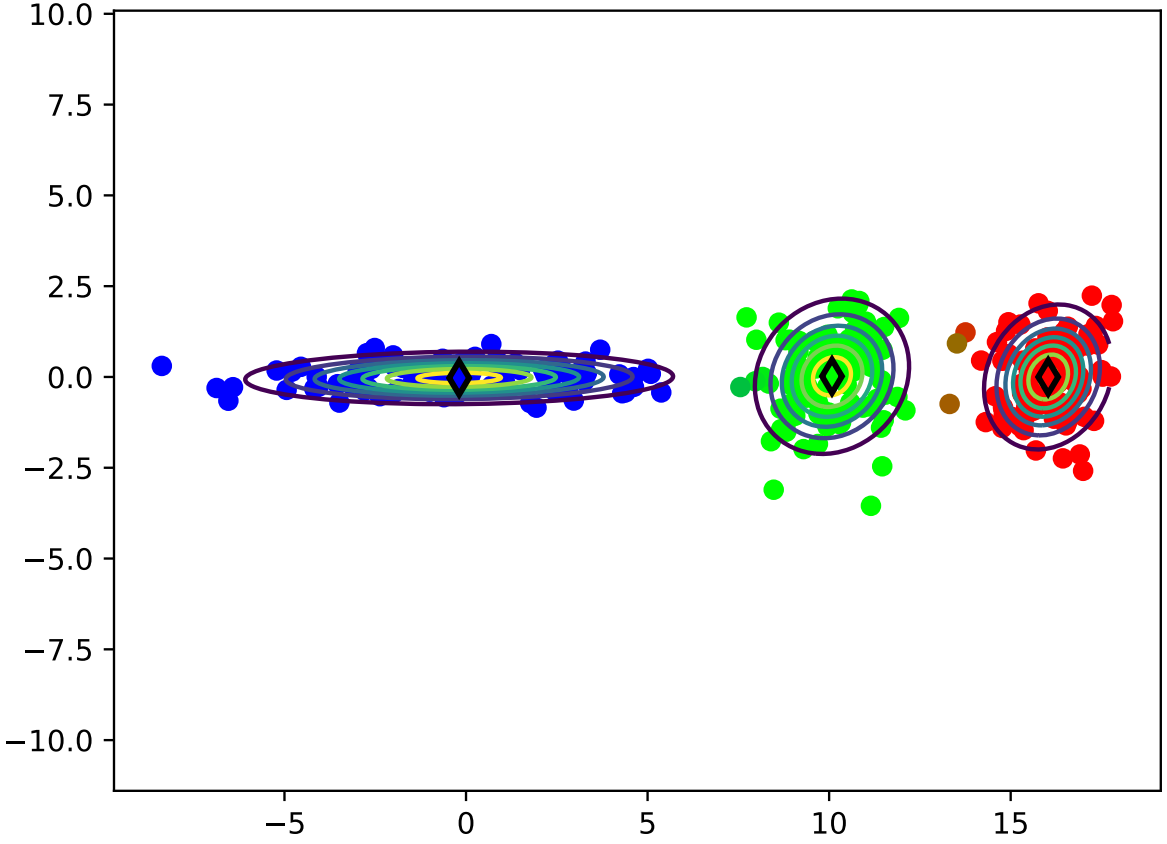
Demo: elliptical clusters

E... M... E... M... E...



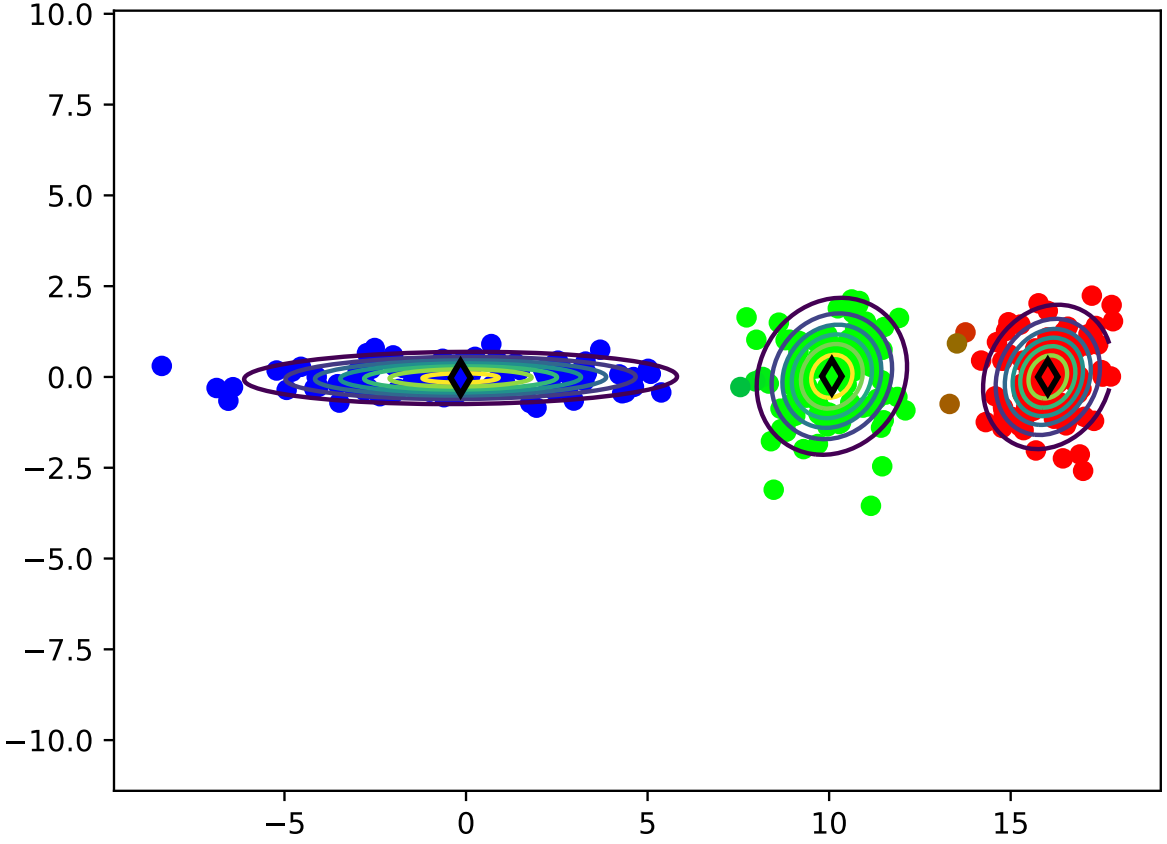
Demo: elliptical clusters

E... M... E... M... E... M...



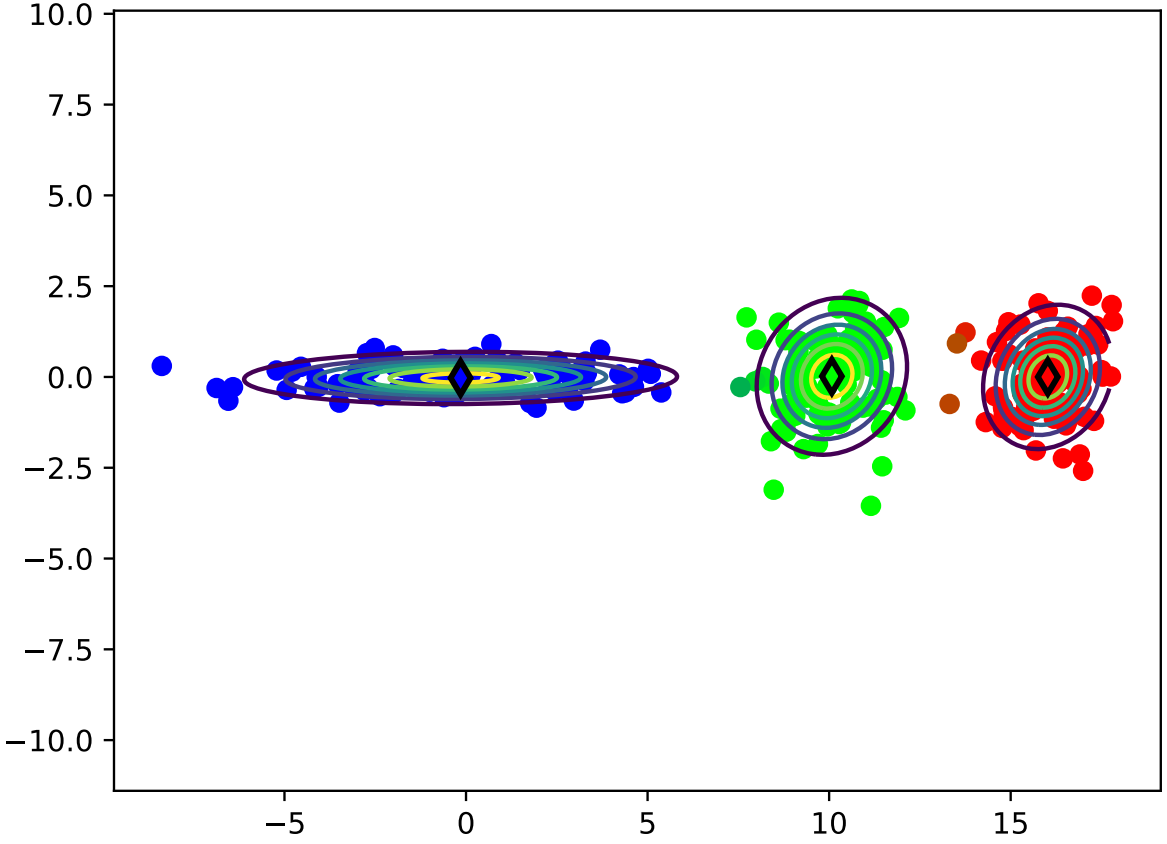
Demo: elliptical clusters

E... M... E... M... E... M... E...



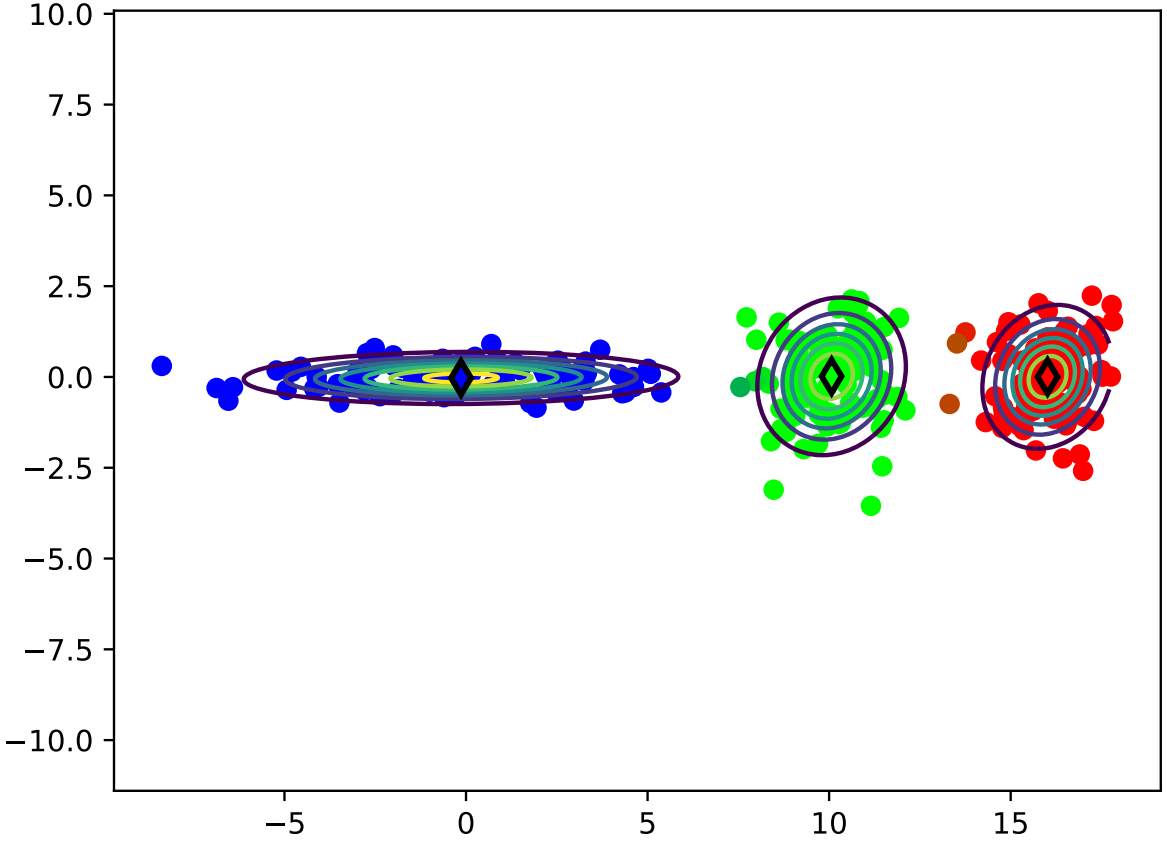
Demo: elliptical clusters

E... M... E... M... E... M... E... M...



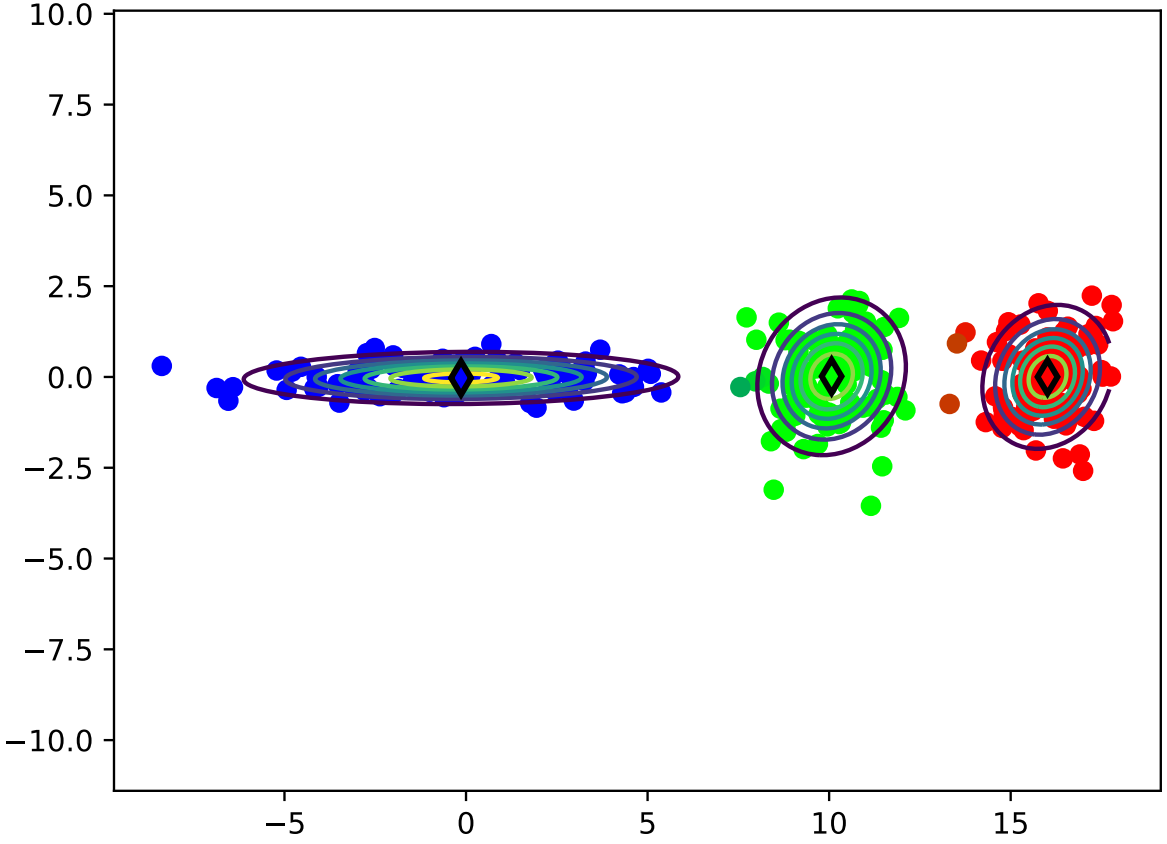
Demo: elliptical clusters

E... M... E... M... E... M... E... M...



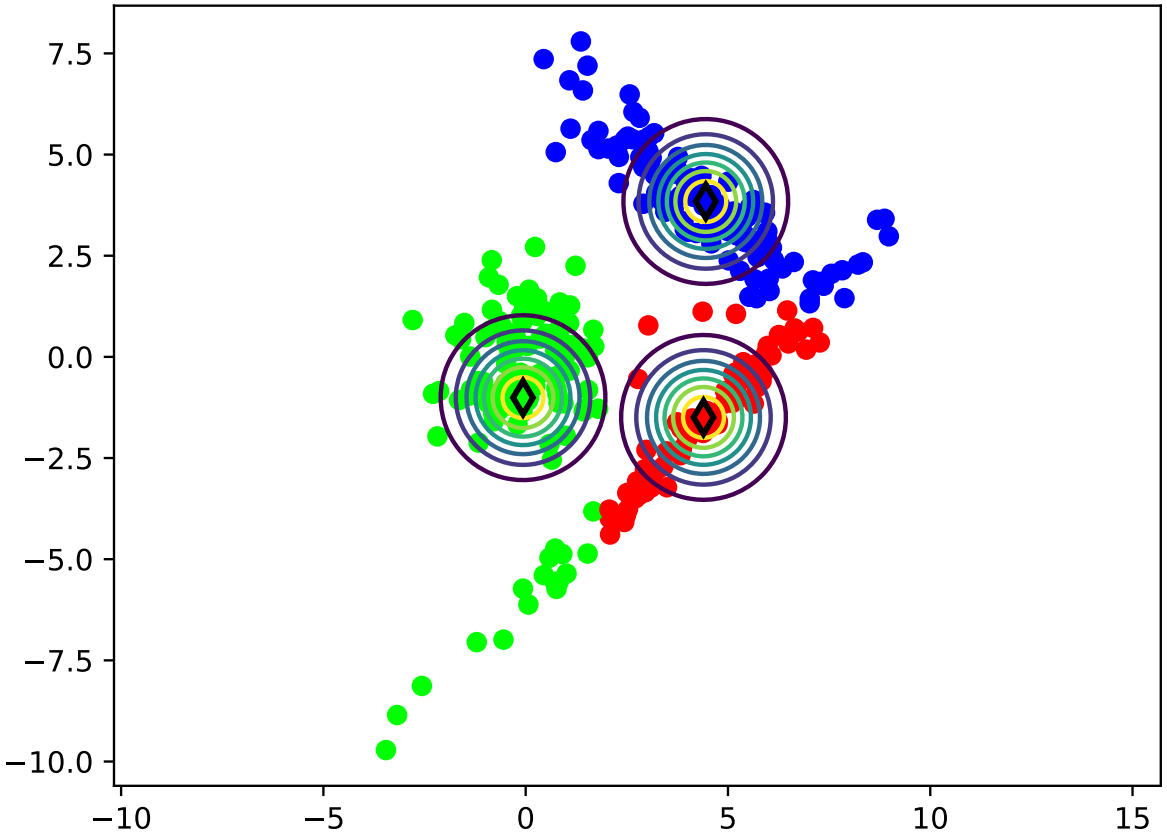
Demo: elliptical clusters

E... M... E... M... E... M... E... M...



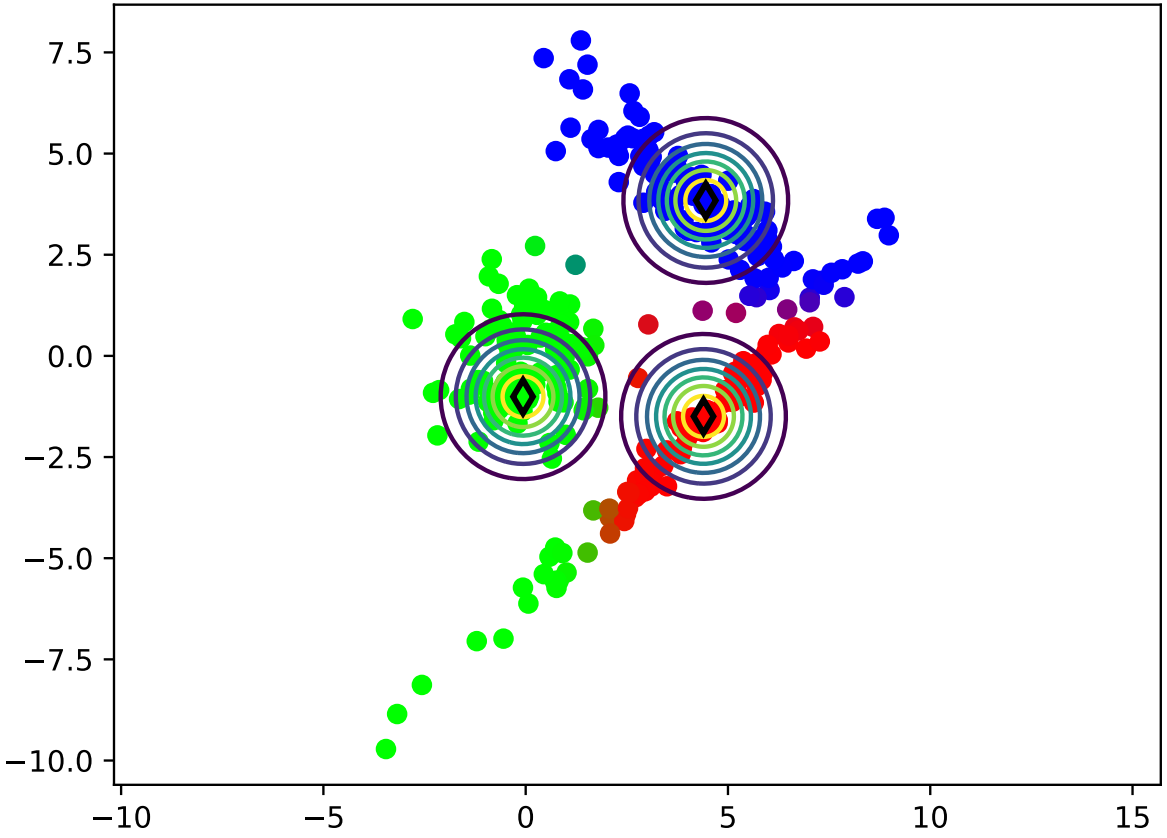
Demo: elliptical clusters

E... M... E... M... E... M... E... M...



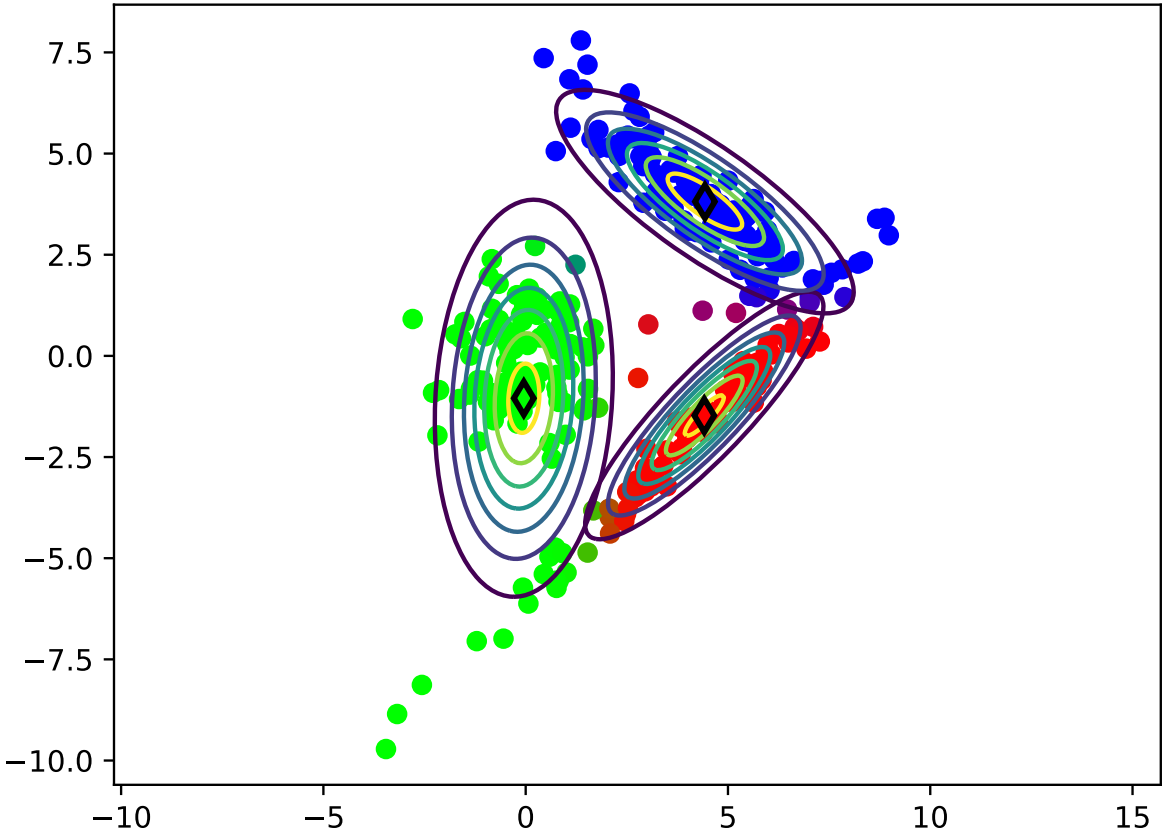
Demo: elliptical clusters

E... M... E... M... E... M... E... M...



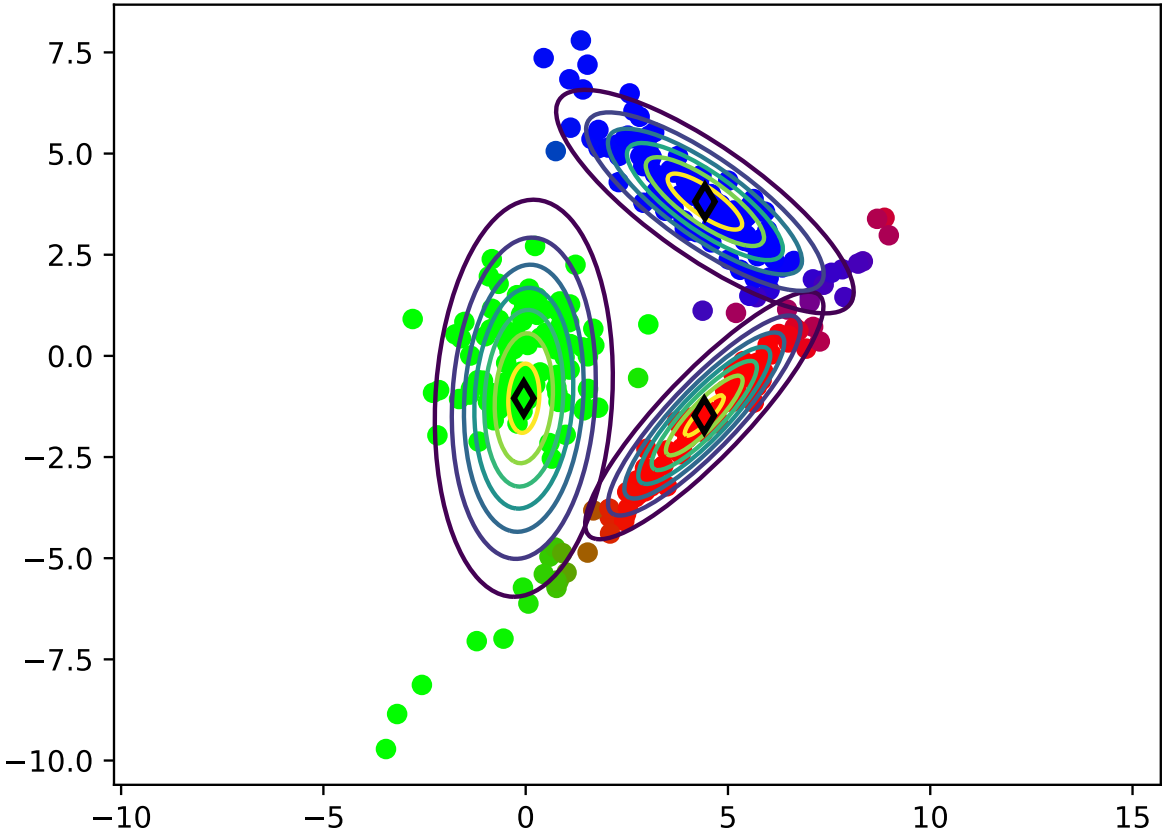
Demo: elliptical clusters

E... M... E... M... E... M... E... M...



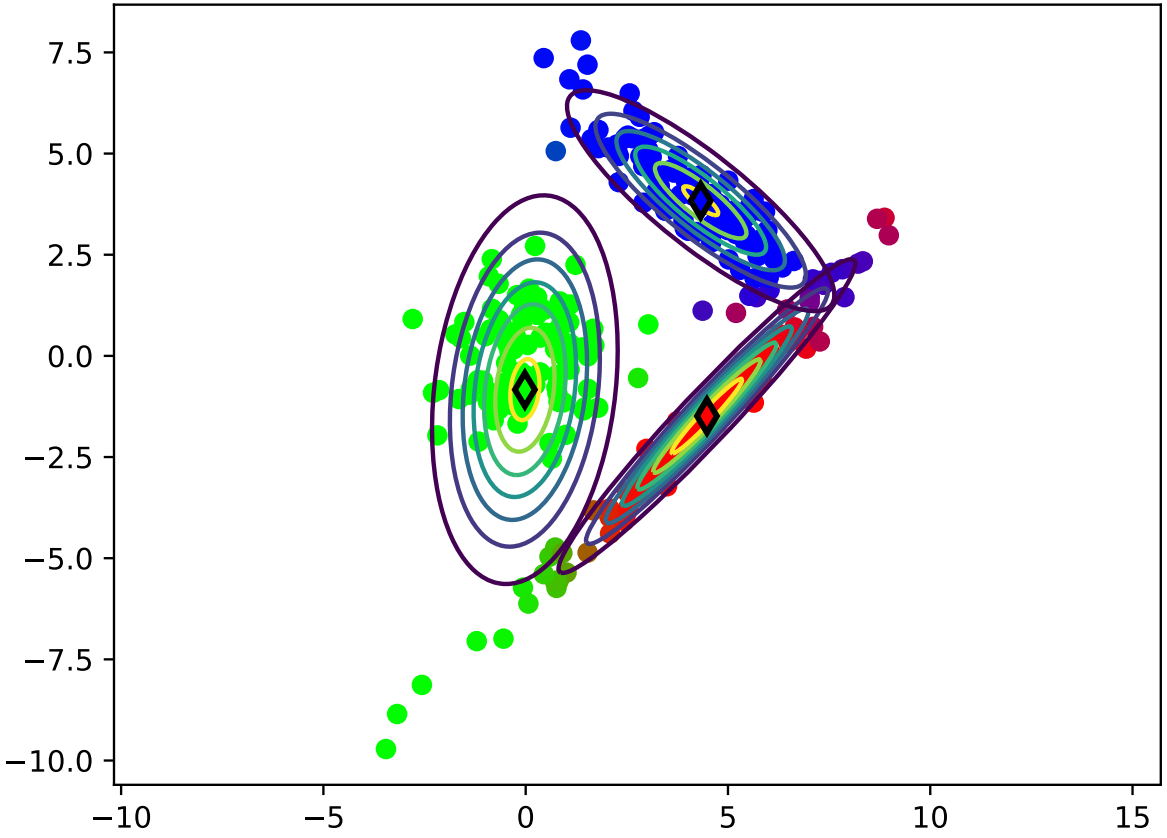
Demo: elliptical clusters

E... M... E... M... E... M... E... M...



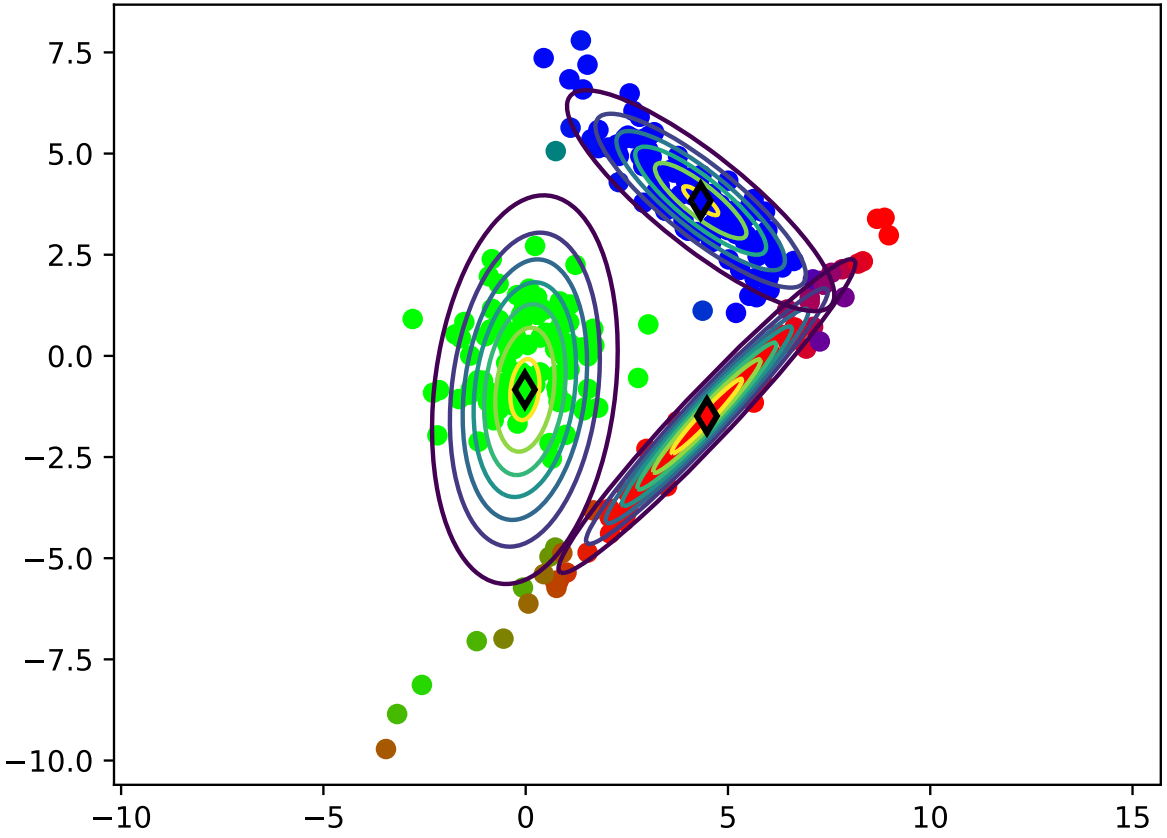
Demo: elliptical clusters

E... M... E... M... E... M... E... M...



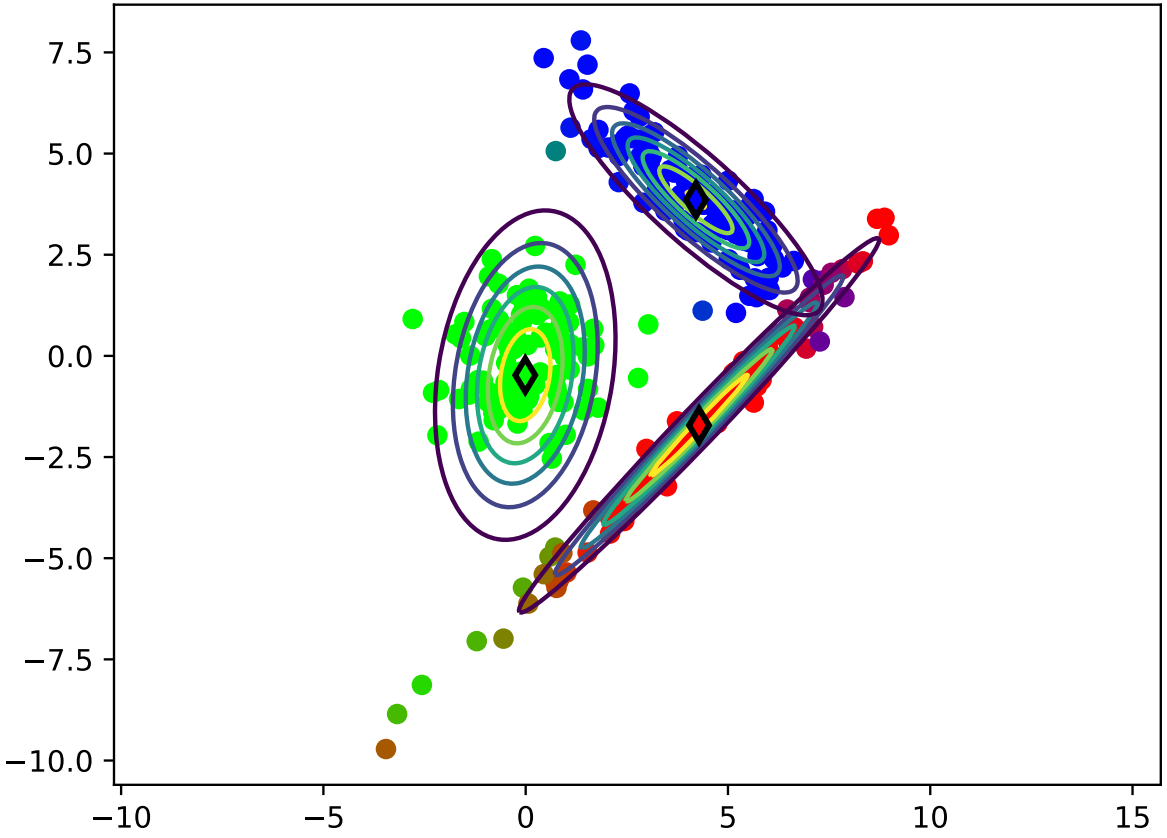
Demo: elliptical clusters

E... M... E... M... E... M... E... M...



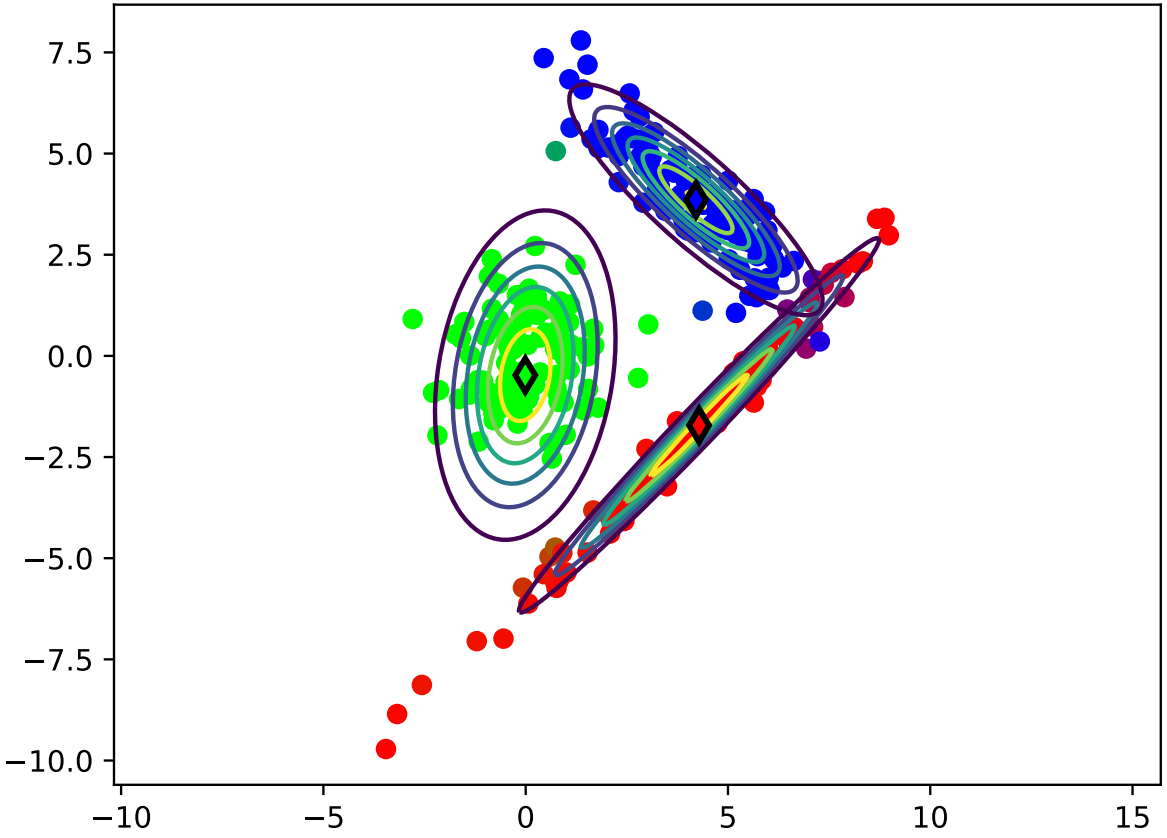
Demo: elliptical clusters

E... M... E... M... E... M... E... M...



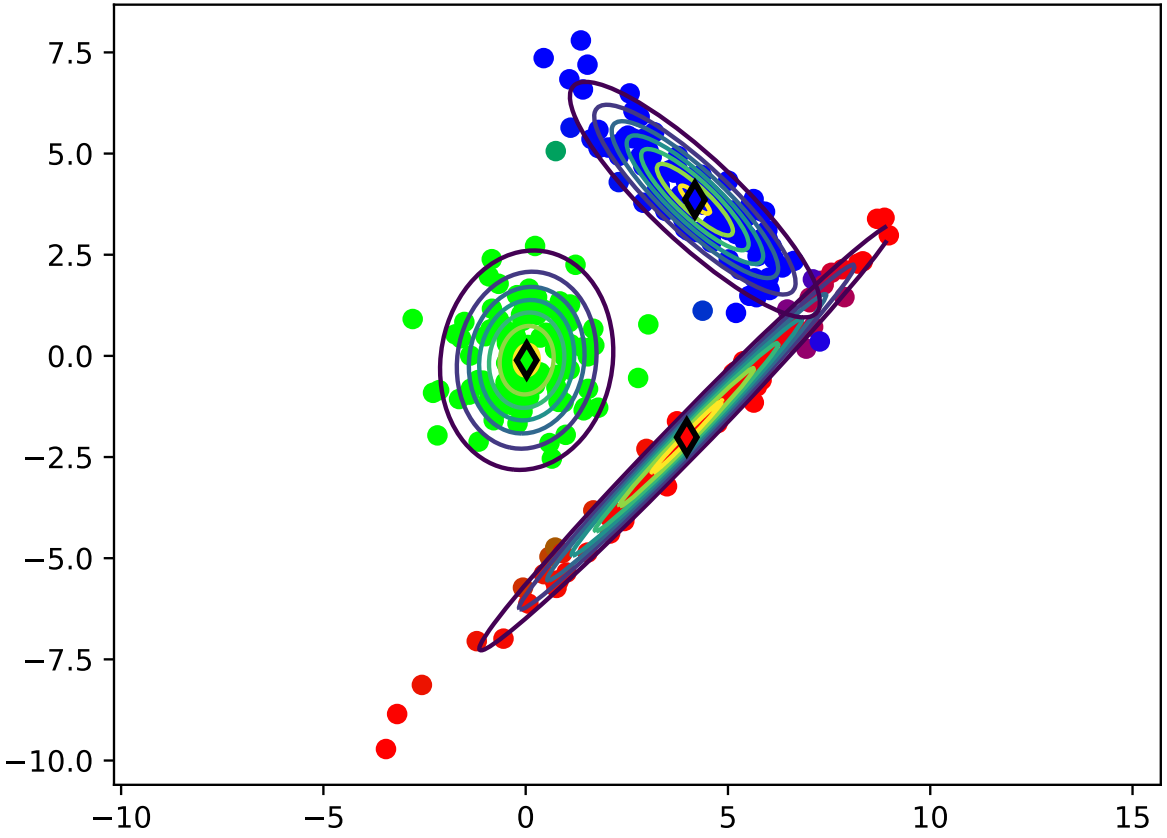
Demo: elliptical clusters

E... M... E... M... E... M... E... M...



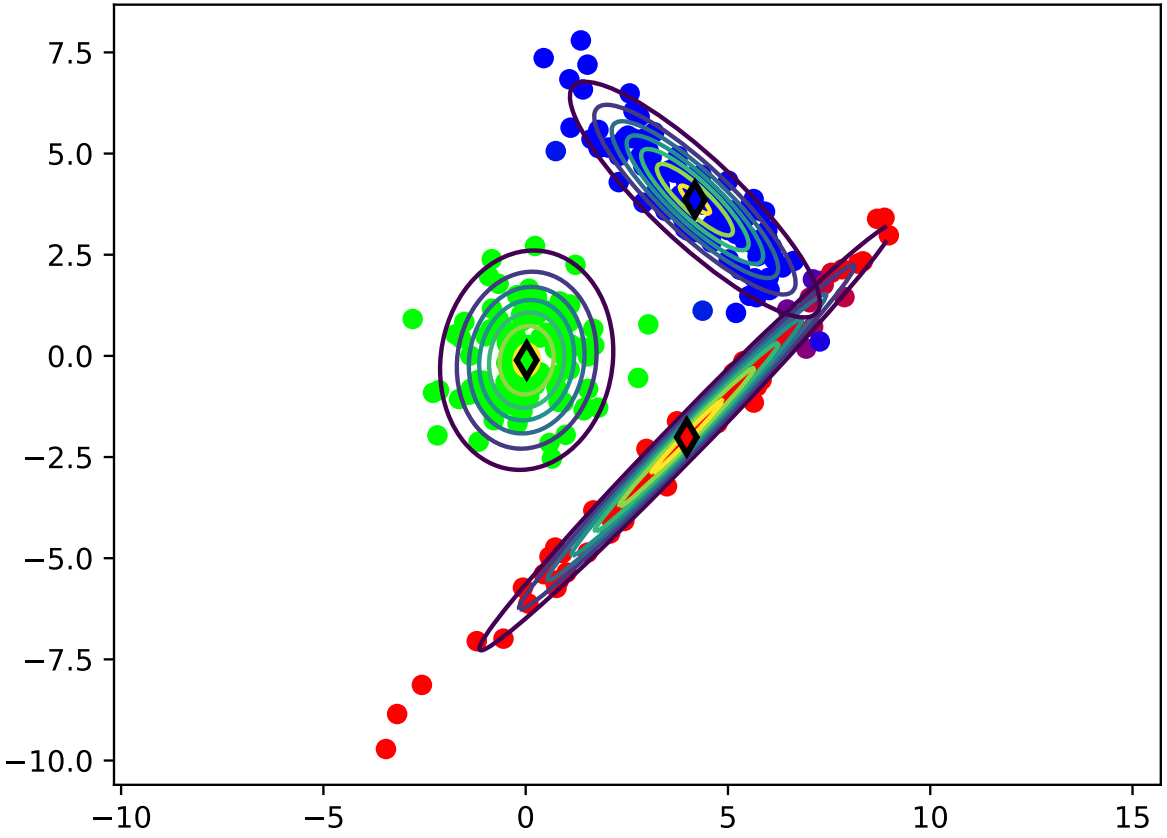
Demo: elliptical clusters

E... M... E... M... E... M... E... M...



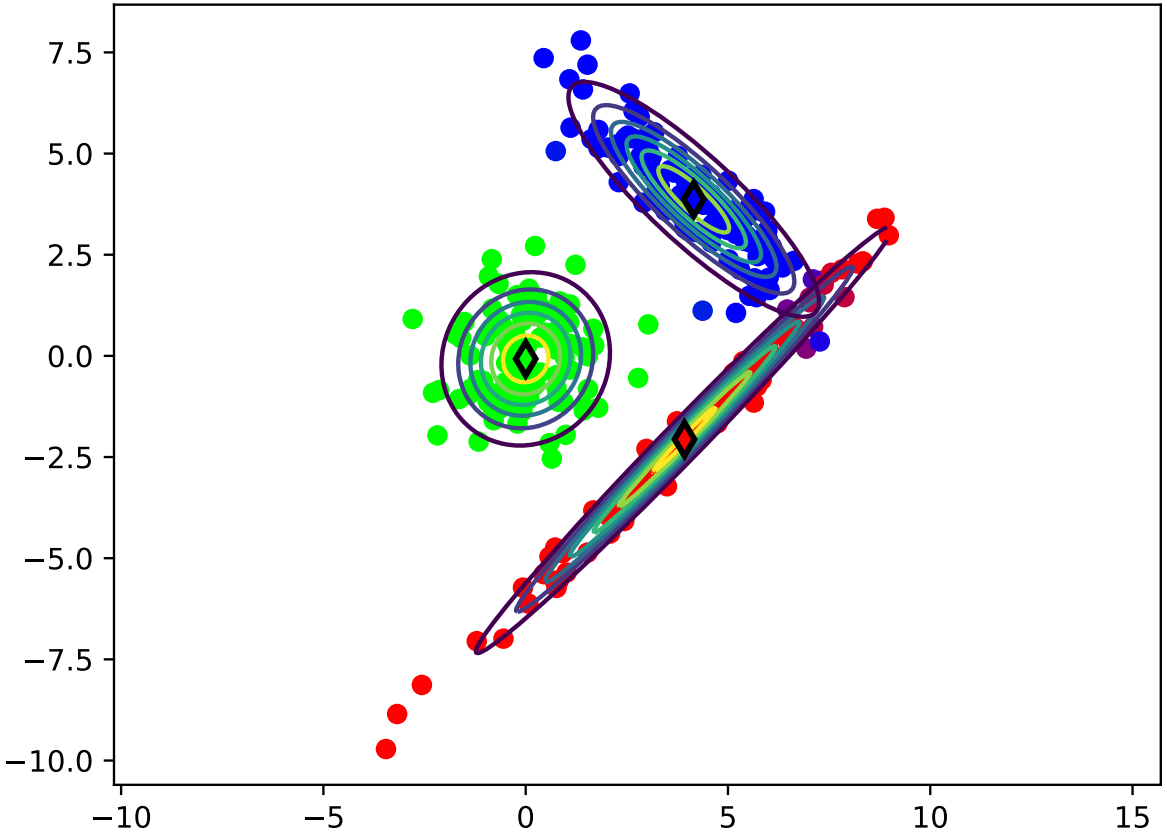
Demo: elliptical clusters

E... M... E... M... E... M... E... M...



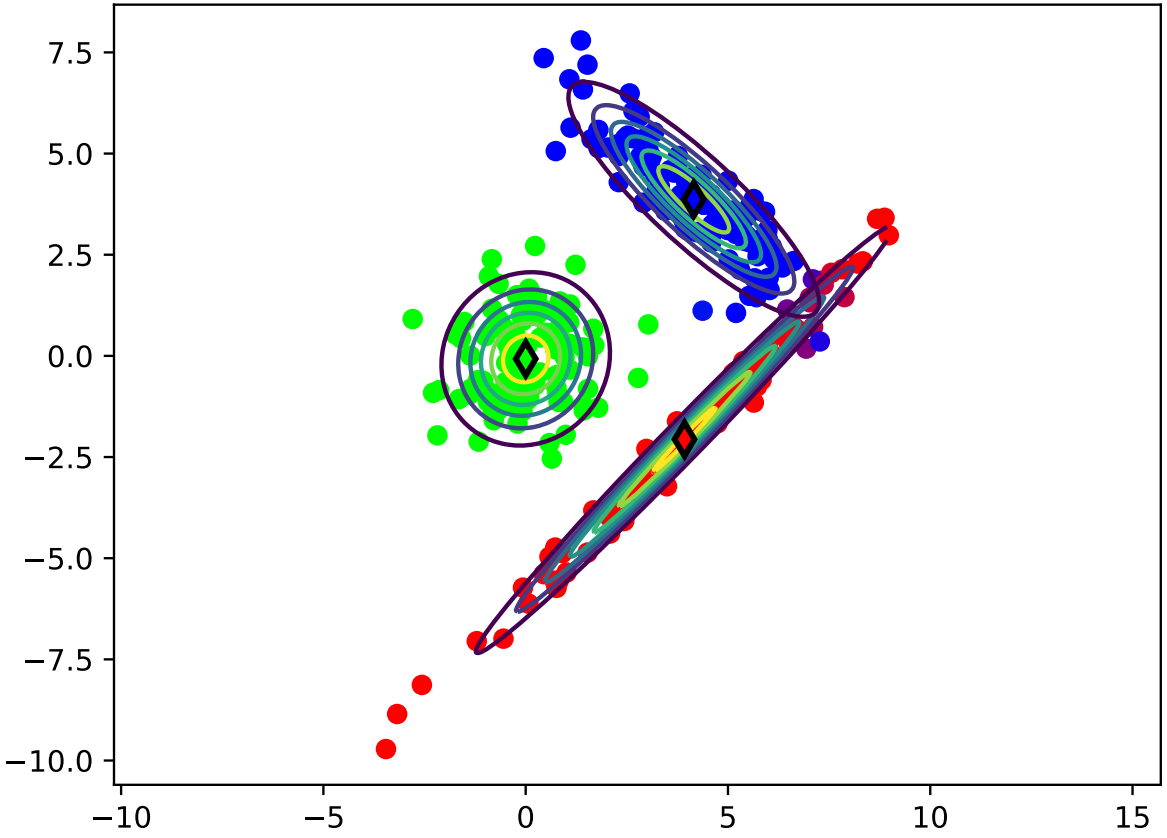
Demo: elliptical clusters

E... M... E... M... E... M... E... M...



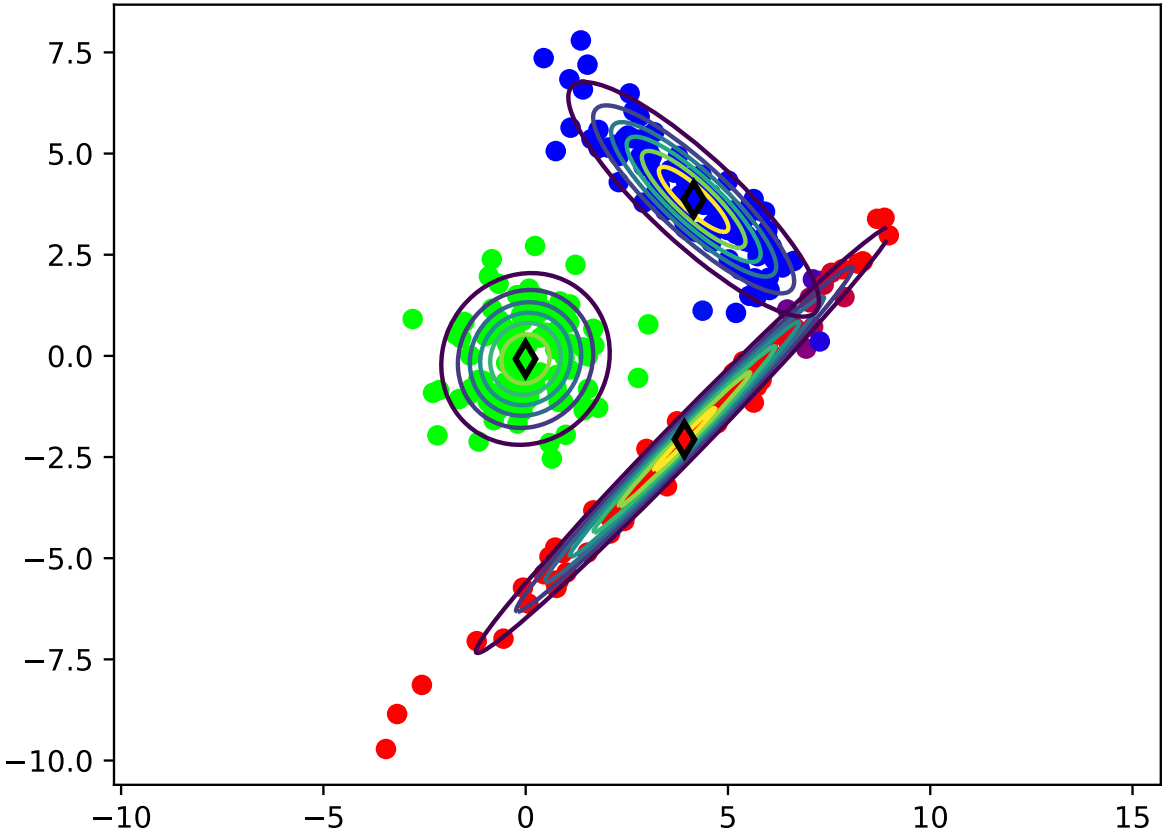
Demo: elliptical clusters

E... M... E... M... E... M... E... M...



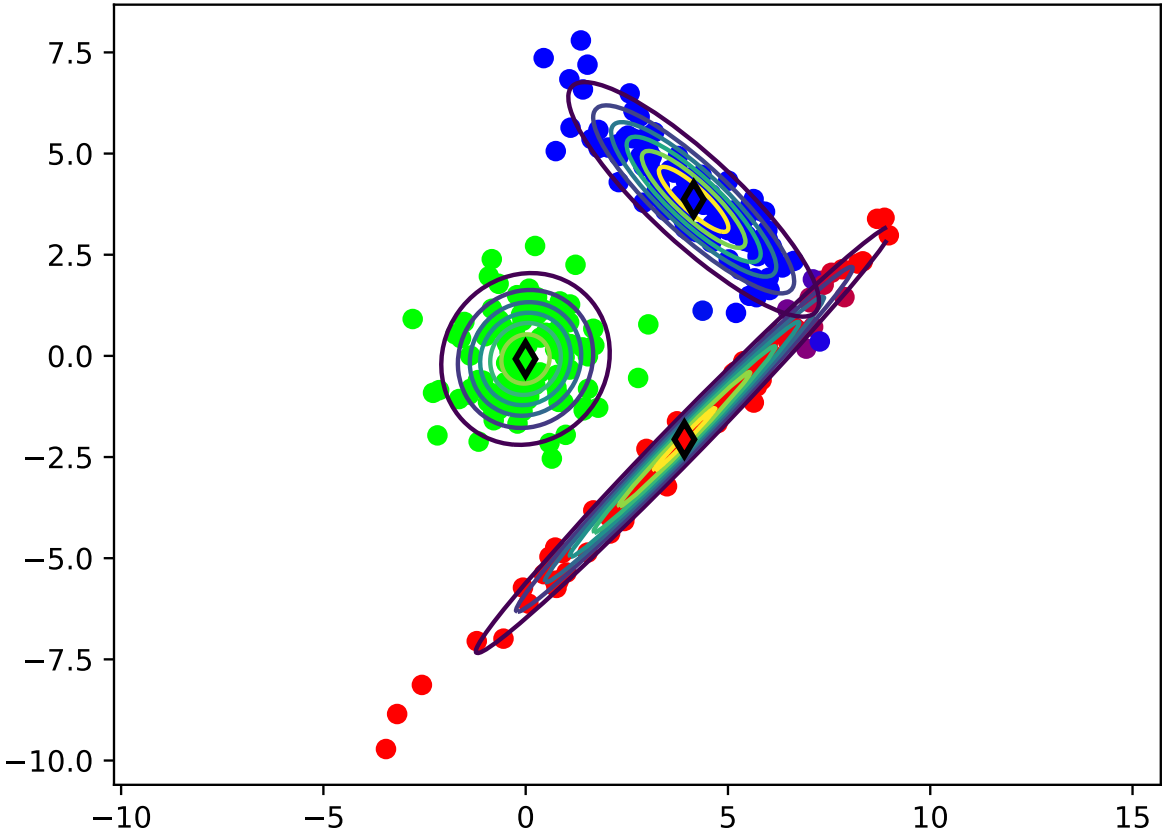
Demo: elliptical clusters

E... M... E... M... E... M... E... M...



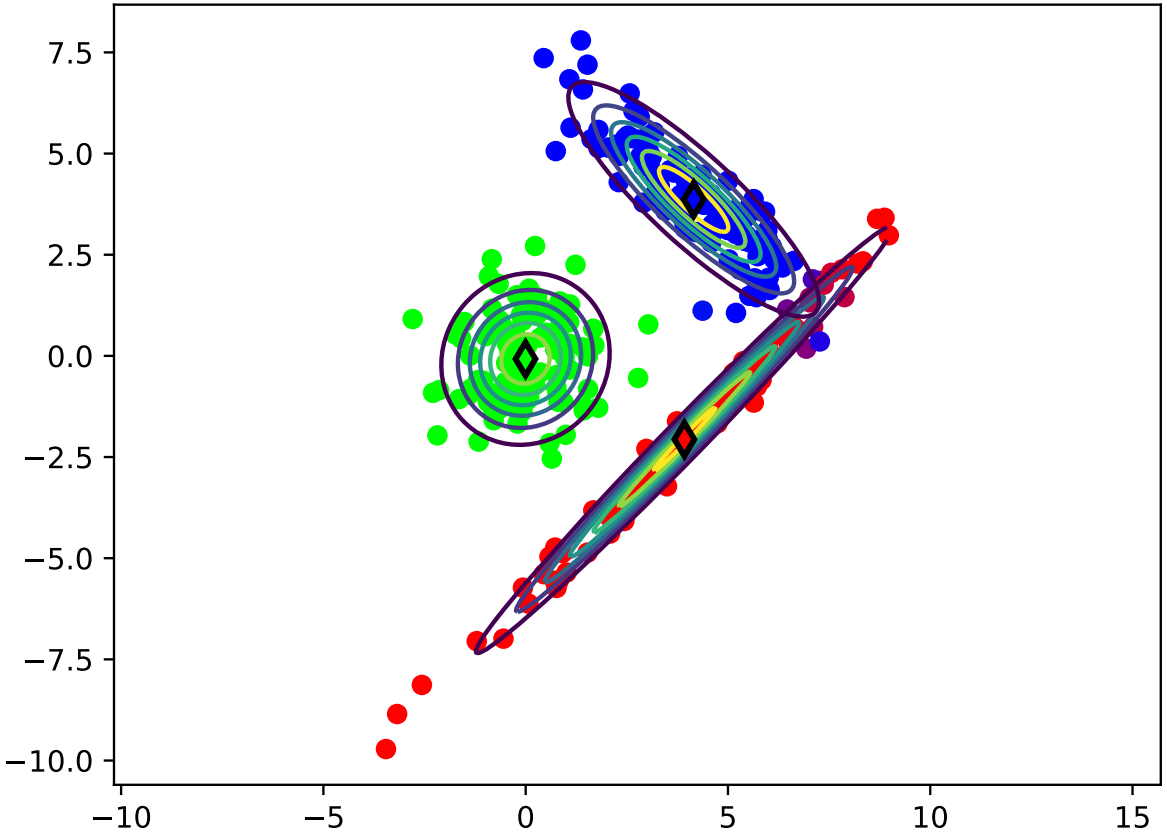
Demo: elliptical clusters

E... M... E... M... E... M... E... M...



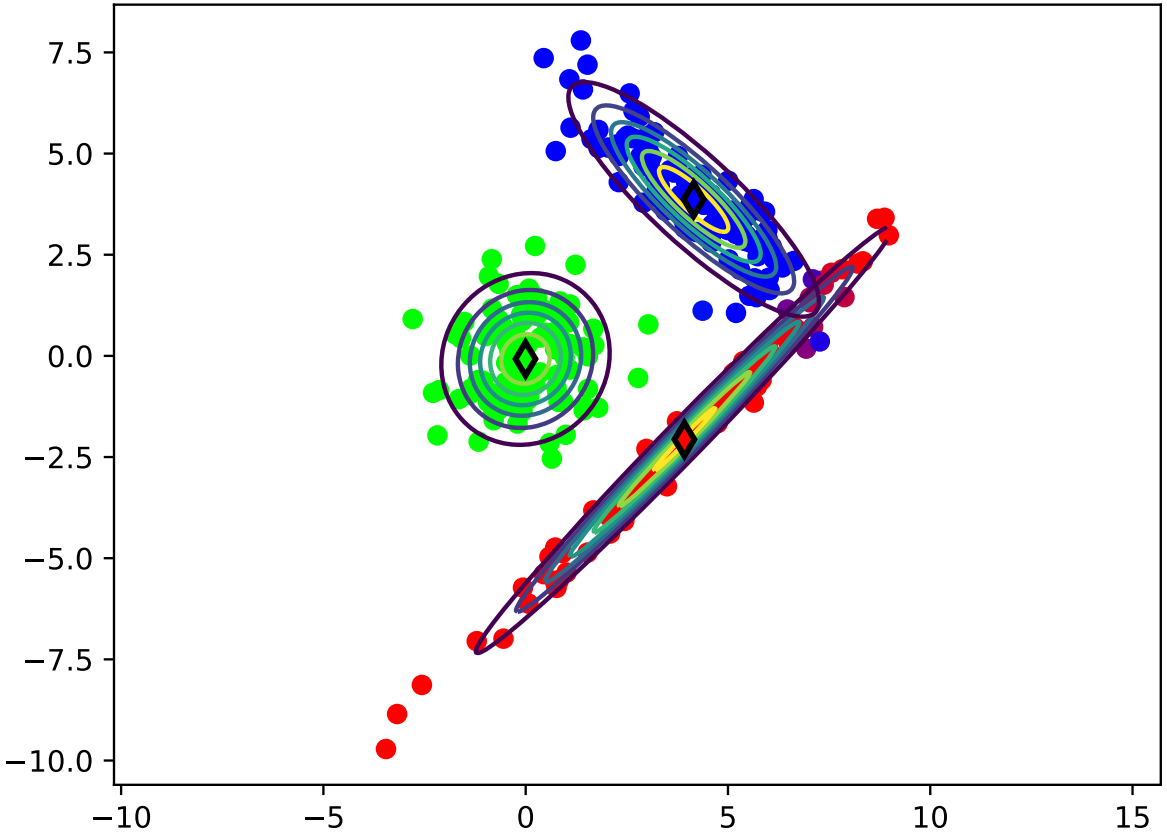
Demo: elliptical clusters

E... M... E... M... E... M... E... M...



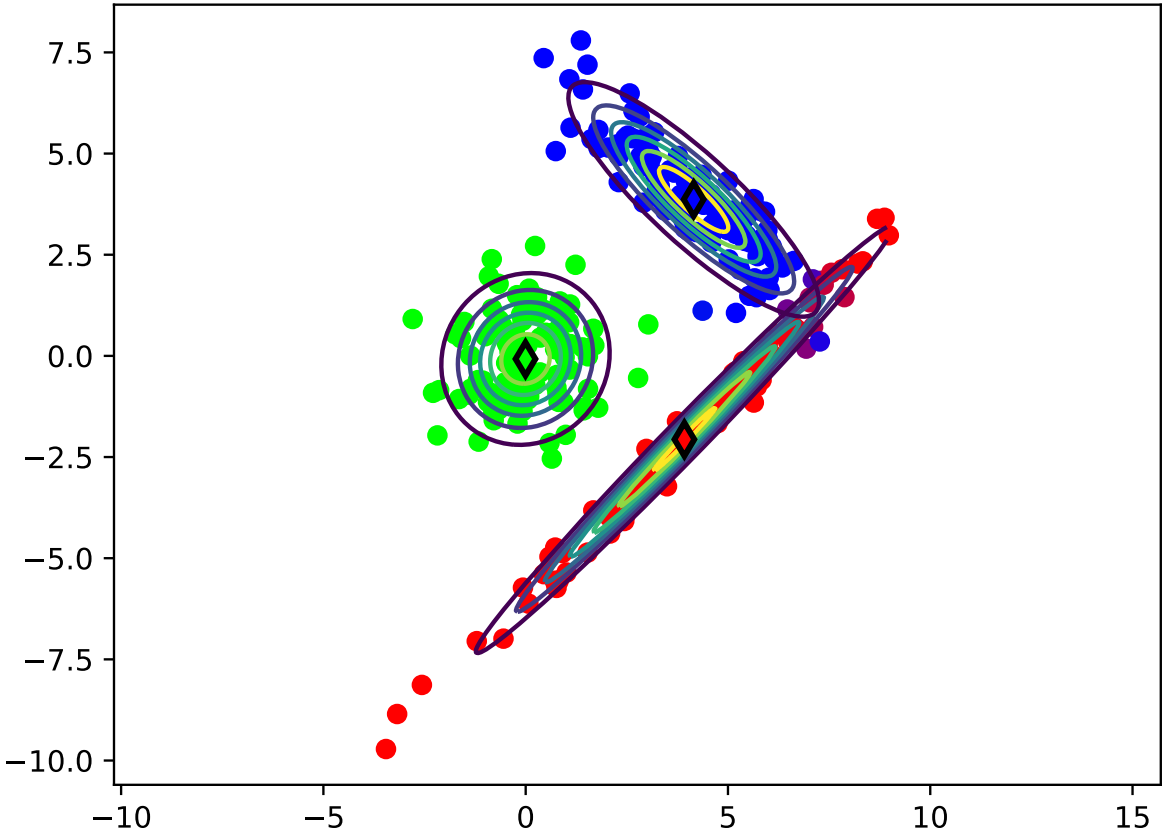
Demo: elliptical clusters

E... M... E... M... E... M... E... M...



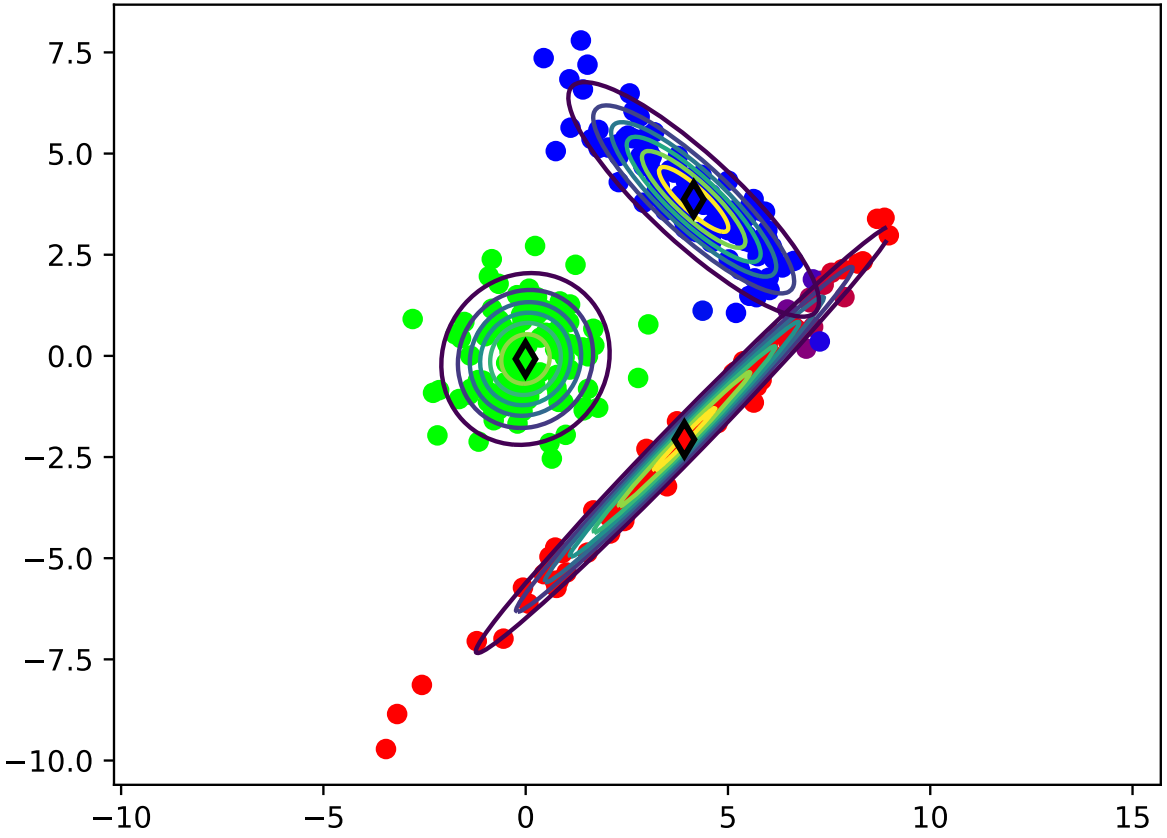
Demo: elliptical clusters

E... M... E... M... E... M... E... M...



Demo: elliptical clusters

E... M... E... M... E... M... E... M...



Theorem.

Suppose $(\boldsymbol{R}_0, \boldsymbol{\theta}_0) \in \mathcal{R}_{n,k} \times \boldsymbol{\Theta}$ arbitrary, thereafter $(\boldsymbol{R}_t, \boldsymbol{\theta}_t)$ given by E-M:

$$(\boldsymbol{R}_t)_{ij} := p_{\boldsymbol{\theta}_{t-1}}(y = j | \boldsymbol{x}_i). \qquad \text{and} \qquad \boldsymbol{\theta}_t := \arg\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathcal{L}(\boldsymbol{\theta}; \boldsymbol{R}_t)$$

Then

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}_t; \boldsymbol{R}_t) &\leq \max_{\boldsymbol{R} \in \mathcal{R}_{n \times k}} \mathcal{L}(\boldsymbol{\theta}_t; \boldsymbol{R}) = \mathcal{L}(\boldsymbol{\theta}_t; \boldsymbol{R}_{t+1}) = \mathcal{L}(\boldsymbol{\theta}_t) \\ &\leq \mathcal{L}(\boldsymbol{\theta}_{t+1}; \boldsymbol{R}_{t+1}). \end{aligned}$$

In particular, $\mathcal{L}(\boldsymbol{\theta}_t) \leq \mathcal{L}(\boldsymbol{\theta}_{t+1})$.

Theorem.
Suppose $(\boldsymbol{R}_0, \boldsymbol{\theta}_0) \in \mathcal{R}_{n,k} \times \boldsymbol{\Theta}$ arbitrary, thereafter $(\boldsymbol{R}_t, \boldsymbol{\theta}_t)$ given by E-M:

$$(\boldsymbol{R}_t)_{ij} := p_{\boldsymbol{\theta}_{t-1}}(y = j | \boldsymbol{x}_i). \qquad \text{and} \qquad \boldsymbol{\theta}_t := \arg\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathcal{L}(\boldsymbol{\theta}; \boldsymbol{R}_t)$$

Then

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}_t; \boldsymbol{R}_t) &\leq \max_{\boldsymbol{R} \in \mathcal{R}_{n \times k}} \mathcal{L}(\boldsymbol{\theta}_t; \boldsymbol{R}) = \mathcal{L}(\boldsymbol{\theta}_t; \boldsymbol{R}_{t+1}) = \mathcal{L}(\boldsymbol{\theta}_t) \\ &\leq \mathcal{L}(\boldsymbol{\theta}_{t+1}; \boldsymbol{R}_{t+1}). \end{aligned}$$

In particular, $\mathcal{L}(\boldsymbol{\theta}_t) \leq \mathcal{L}(\boldsymbol{\theta}_{t+1})$.

Remarks.

- We proved a similar guarantee for k-means, which is also an *alternating minimization* scheme.
- Similarly, MLE for Gaussian mixtures is NP-hard; it is also known to need exponentially many samples in k to information-theoretically recover the parameters.

Proof. We've already shown:

- $\mathcal{L}(\boldsymbol{\theta}_t; \boldsymbol{R}_{t+1}) = \mathcal{L}(\boldsymbol{\theta}_t)$;
- $\mathcal{L}(\boldsymbol{\theta}_t; \boldsymbol{R}_{t+1}) = \max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathcal{L}(\boldsymbol{\theta}; \boldsymbol{R}_{t+1}) \leq \mathcal{L}(\boldsymbol{\theta}_{t+1}; \boldsymbol{R}_{t+1})$ by definition of $\boldsymbol{\theta}_{t+1}$.

We still need to show: $\mathcal{L}(\boldsymbol{\theta}_t; \boldsymbol{R}_{t+1}) = \max_{\boldsymbol{R} \in \mathcal{R}_{n,k}} \mathcal{L}(\boldsymbol{\theta}_{t+1}; \boldsymbol{R})$.
We'll give two proofs.

Proof. We've already shown:

- $\mathcal{L}(\boldsymbol{\theta}_t; \boldsymbol{R}_{t+1}) = \mathcal{L}(\boldsymbol{\theta}_t)$;
- $\mathcal{L}(\boldsymbol{\theta}_t; \boldsymbol{R}_{t+1}) = \max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathcal{L}(\boldsymbol{\theta}; \boldsymbol{R}_{t+1}) \leq \mathcal{L}(\boldsymbol{\theta}_{t+1}; \boldsymbol{R}_{t+1})$ by definition of $\boldsymbol{\theta}_{t+1}$.

We still need to show: $\mathcal{L}(\boldsymbol{\theta}_t; \boldsymbol{R}_{t+1}) = \max_{\boldsymbol{R} \in \mathcal{R}_{n,k}} \mathcal{L}(\boldsymbol{\theta}_{t+1}; \boldsymbol{R})$.
We'll give two proofs.

By concavity of $\ln$ ("Jensen's inequality" in convexity lectures), for any $\boldsymbol{R} \in \mathcal{R}_{n,k}$,

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}_t; \boldsymbol{R}) &= \sum_{i=1}^{n} \sum_{j=1}^{k} \boldsymbol{R}_{ij} \ln \frac{p_{\boldsymbol{\theta}_t}(\boldsymbol{x}_i, y_i = j)}{\boldsymbol{R}_{ij}} \\
&\leq \sum_{i=1}^{n} \ln \left(\sum_{j=1}^{k} \boldsymbol{R}_{ij} \frac{p_{\boldsymbol{\theta}_t}(\boldsymbol{x}_i, y_i = j)}{\boldsymbol{R}_{ij}} \right) \\
&= \sum_{i=1}^{n} \ln p_{\boldsymbol{\theta}_t}(\boldsymbol{x}_i) = \mathcal{L}(\boldsymbol{\theta}_t) = \mathcal{L}(\boldsymbol{\theta}_t; \boldsymbol{R}_{t+1}).
\end{aligned}$$

Since $\boldsymbol{R}$ was arbitrary, $\max_{\boldsymbol{R} \in \mathcal{R}} \mathcal{L}(\boldsymbol{\theta}_t; \boldsymbol{R}) = \mathcal{L}(\boldsymbol{\theta}_t; \boldsymbol{R}_{t+1})$.

Proof (continued). Here's a second proof of that missing fact.
To evaluate $\arg\max_{\boldsymbol{R}\in\mathcal{R}_{n,k}} \mathcal{L}(\boldsymbol{\theta};\boldsymbol{R})$, consider Lagrangian

$$\sum_{i=1}^{n}\left(\sum_{j=1}^{k} R_{ij}\ln p_{\boldsymbol{\theta}}(\boldsymbol{x}_i, y=j) - \sum_{j=1}^{k} R_{ij}\ln R_{ij} + \lambda_i\left(\sum_{j=1}^{k} R_{ij} - 1\right)\right).$$

Fixing i and taking the gradient with respect to R_{ij} for any j,

$$0 = \ln p_{\boldsymbol{\theta}}(\boldsymbol{x}_i, y_i = j) - \ln R_{ij} - 1 + \lambda_i,$$

giving $R_{ij} = p_{\boldsymbol{\theta}}(\boldsymbol{x}_i, y=j)\exp(\lambda_i - 1)$. Since moreover

$$1 = \sum_j R_{ij} = \exp(\lambda_i - 1)\sum_j p_{\boldsymbol{\theta}}(\boldsymbol{x}_i, y=j) = \exp(\lambda_i - 1)p_{\boldsymbol{\theta}}(\boldsymbol{x}_i),$$

it follows that $\exp(\lambda_i - 1) = {}^{1}/_{p_{\boldsymbol{\theta}}(\boldsymbol{x}_i)}$,
and the optimal $\boldsymbol{R}$ satisfies $R_{ij} = {}^{p_{\boldsymbol{\theta}}(\boldsymbol{x}_i, y=j)}/_{p_{\boldsymbol{\theta}}(\boldsymbol{x}_i)} = p_{\boldsymbol{\theta}}(y=j|\boldsymbol{x}_i)$.

□

Related issues.

Parameter constraints.

E-M for GMMs **still works** if we freeze or constrain some parameters.

Parameter constraints.

E-M for GMMs **still works** if we freeze or constrain some parameters.

Examples:

- **No weights:** initialize $\boldsymbol{\pi} = (1/k, \ldots, 1/k)$ and never update it.
- **Diagonal covariance matrices:** update everything as before, except $\boldsymbol{\Sigma}_j := \mathrm{diag}((\boldsymbol{\sigma}_j)_1^2, \ldots, (\boldsymbol{\sigma}_j)_d^2)$ where

$$(\boldsymbol{\sigma}_j)_l^2 := \frac{\sum_{i=1}^n R_{ij}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)_l^2}{n\pi_j};$$

that is: we use coordinate-wise sample variances weighted by $\boldsymbol{R}$.
Why is this a good idea?

Parameter constraints.

E-M for GMMs **still works** if we freeze or constrain some parameters.

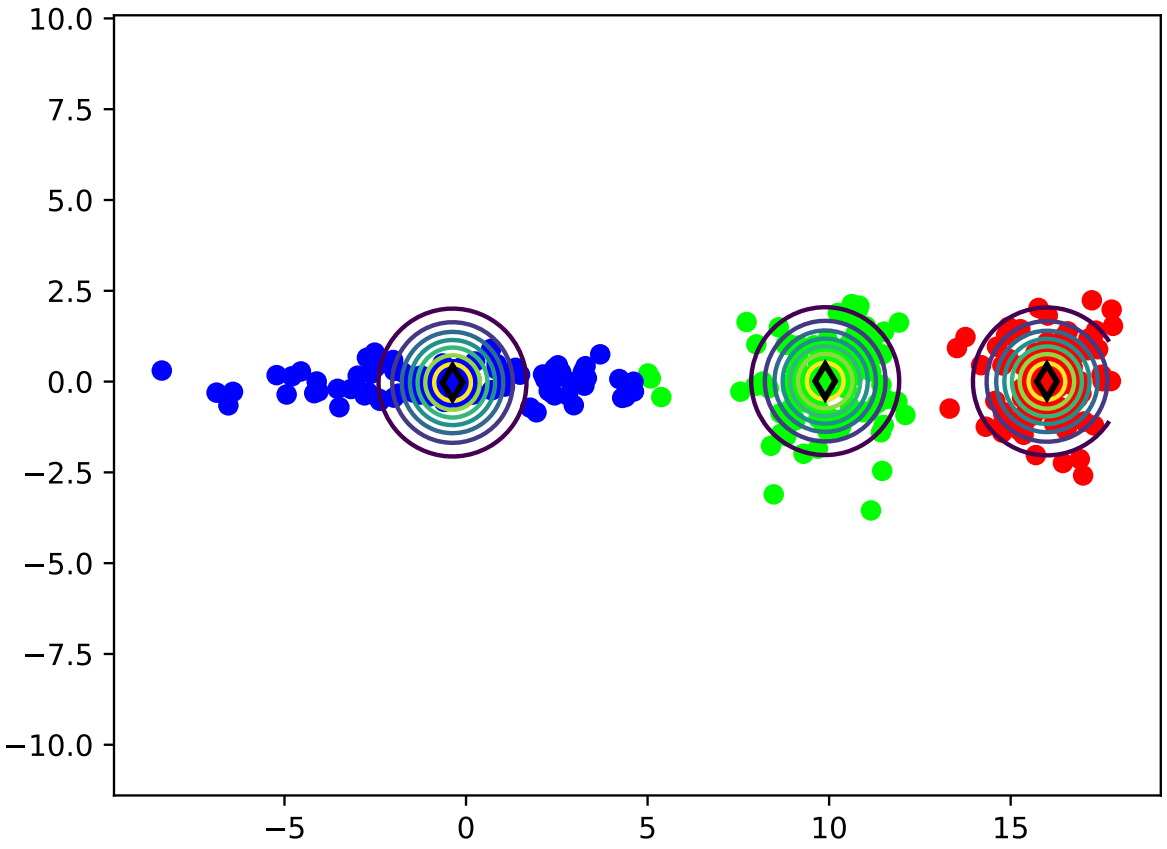
Examples:

- **No weights:** initialize $\boldsymbol{\pi} = (1/k, \ldots, 1/k)$ and never update it.
- **Diagonal covariance matrices:** update everything as before, except $\boldsymbol{\Sigma}_j := \mathrm{diag}((\boldsymbol{\sigma}_j)_1^2, \ldots, (\boldsymbol{\sigma}_j)_d^2)$ where

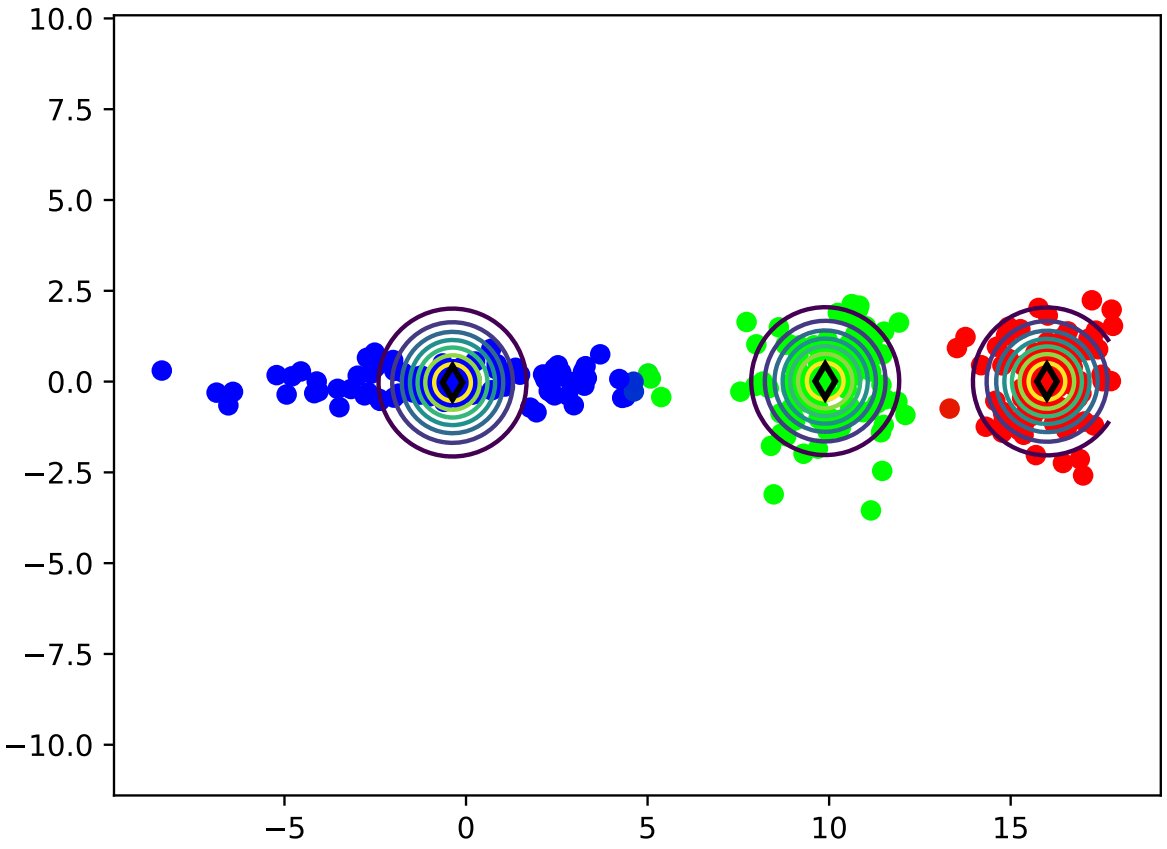
$$(\boldsymbol{\sigma}_j)_l^2 := \frac{\sum_{i=1}^n R_{ij}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)_l^2}{n\pi_j};$$

that is: we use coordinate-wise sample variances weighted by $\boldsymbol{R}$.
Why is this a good idea?
Computation (of inverse), sample complexity, ...

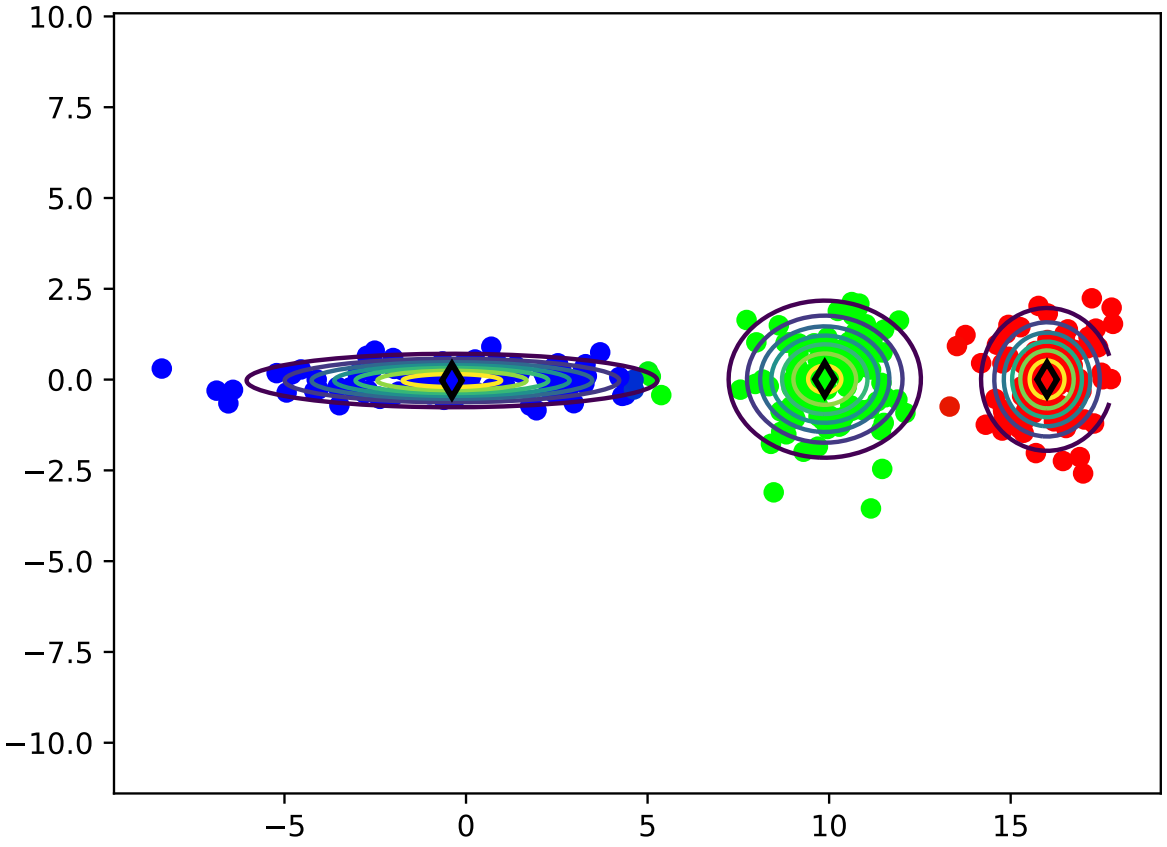
Gaussian Mixture Model with diagonal covariances.



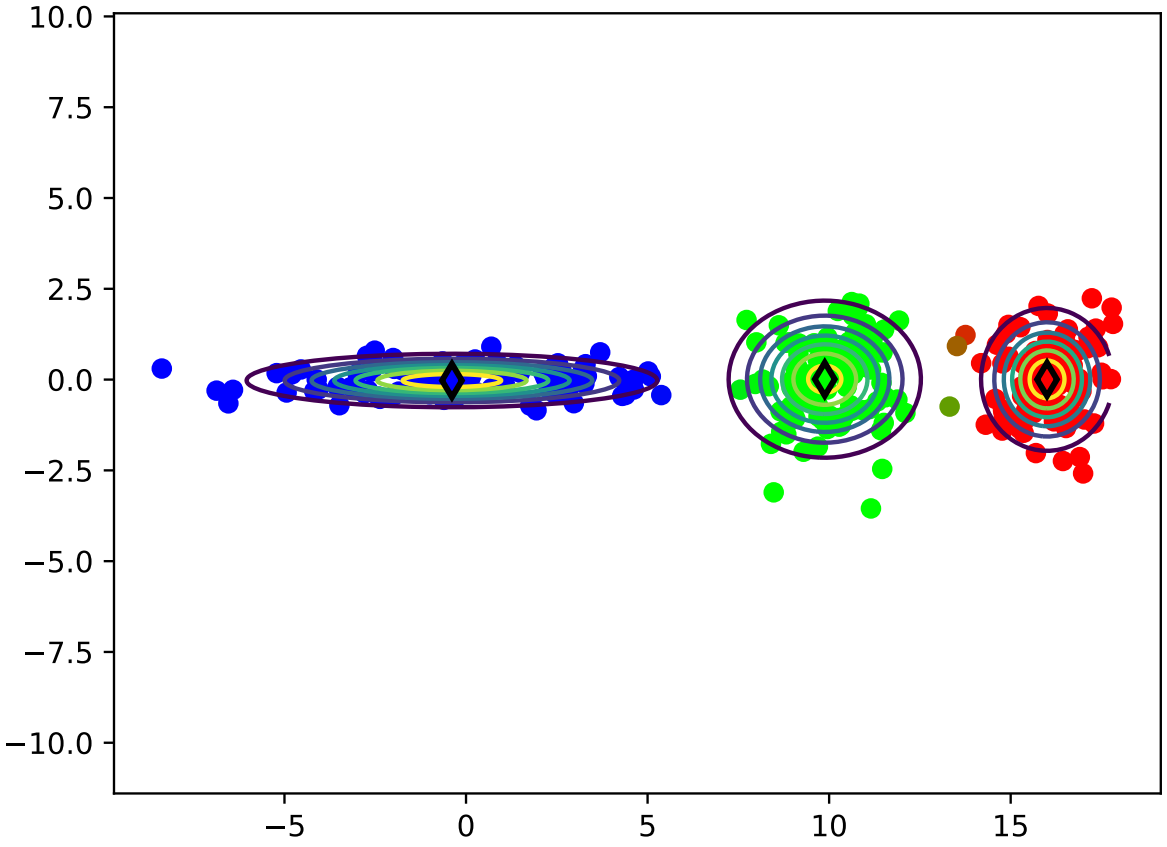
Gaussian Mixture Model with diagonal covariances.



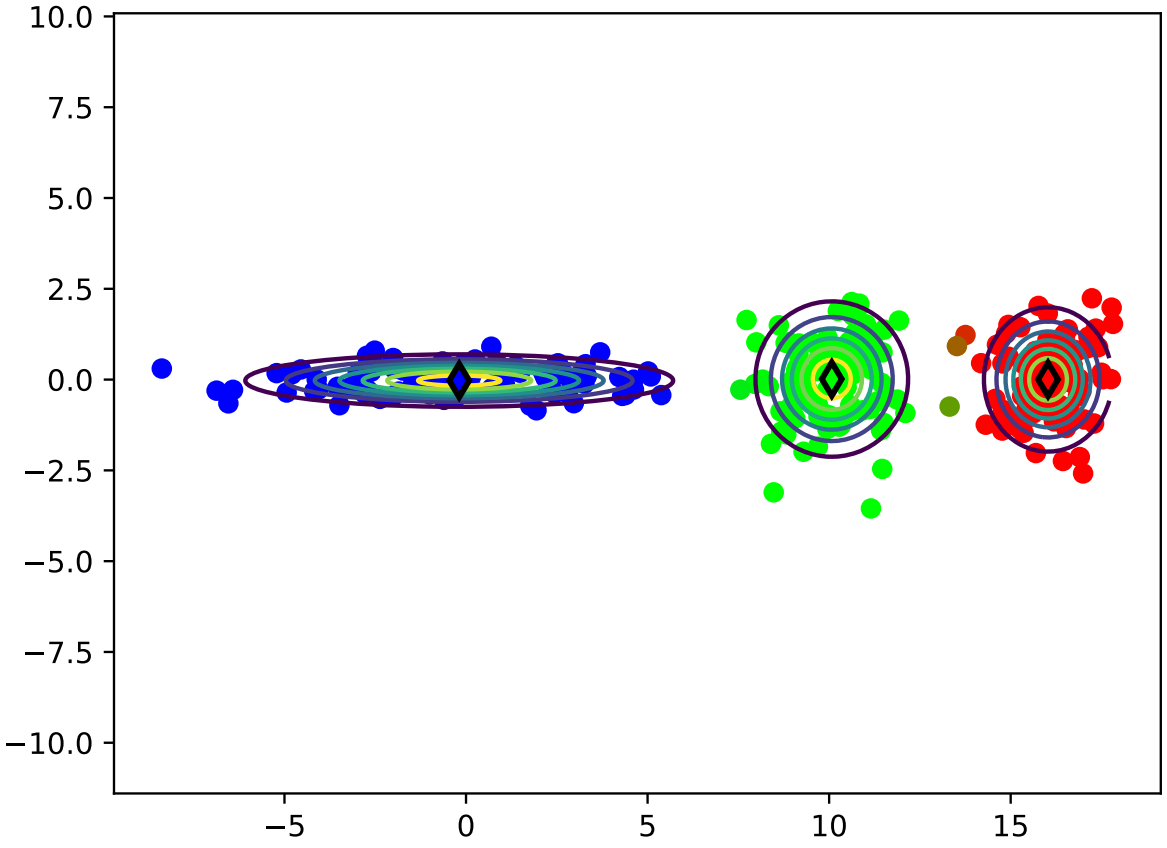
Gaussian Mixture Model with diagonal covariances.



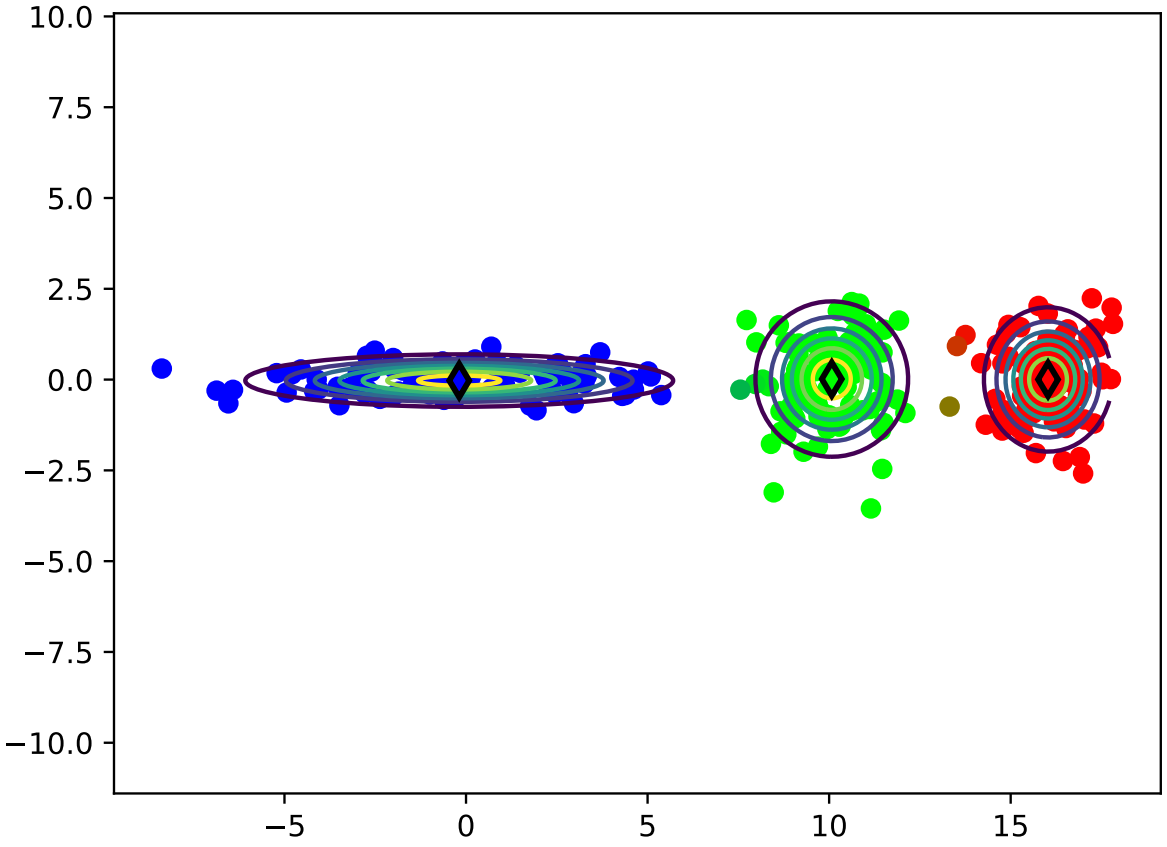
Gaussian Mixture Model with diagonal covariances.



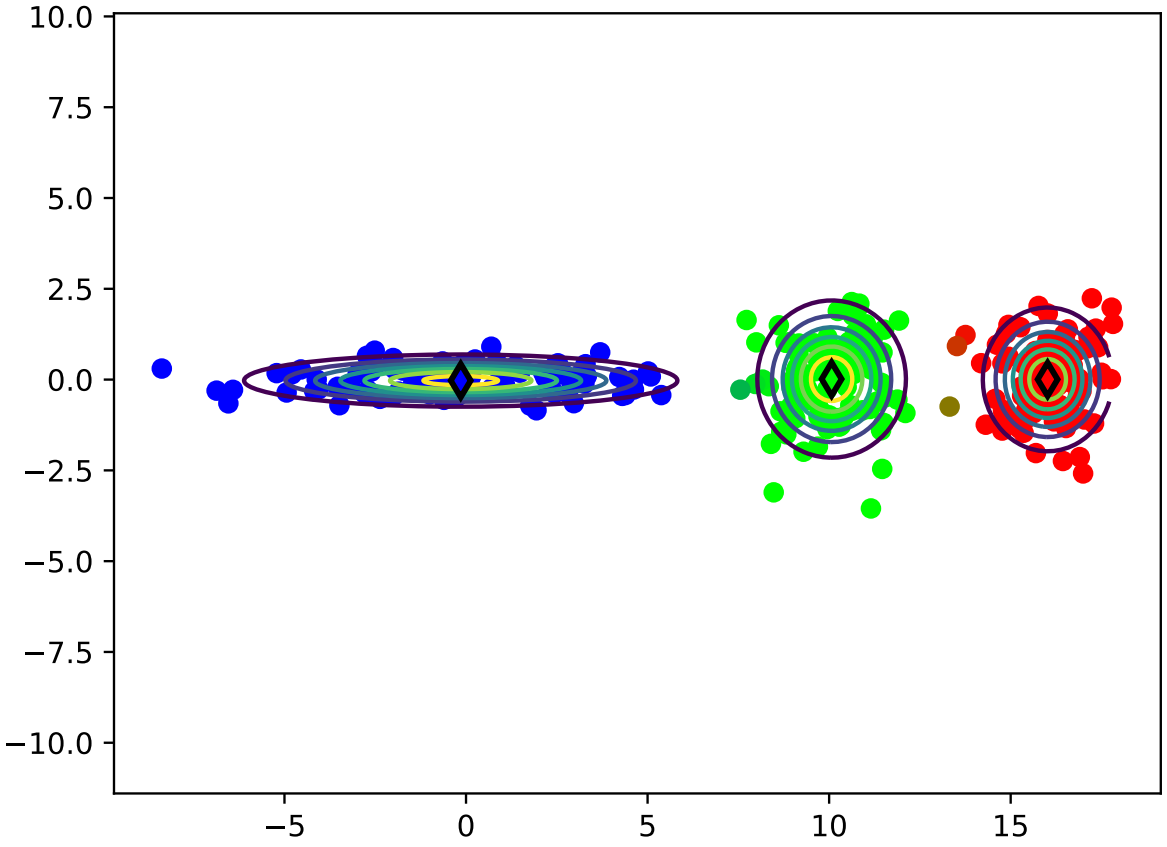
Gaussian Mixture Model with diagonal covariances.



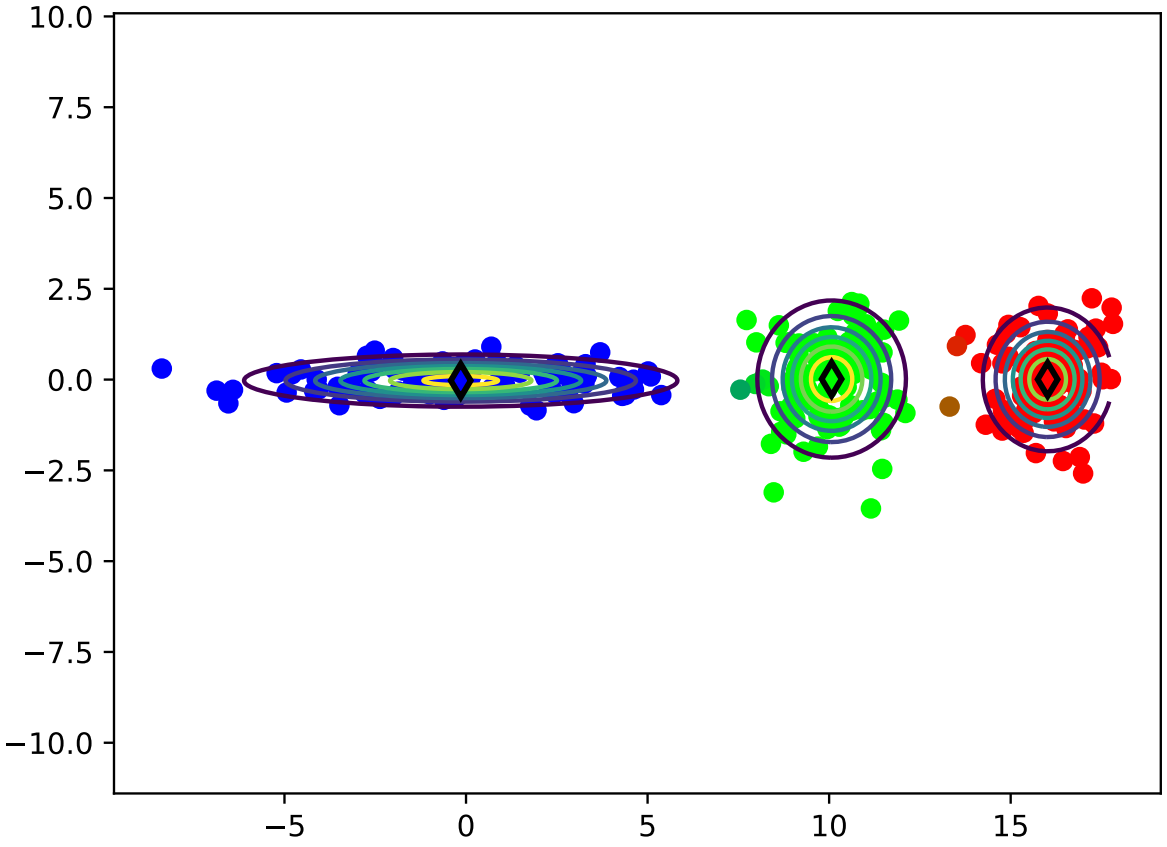
Gaussian Mixture Model with diagonal covariances.



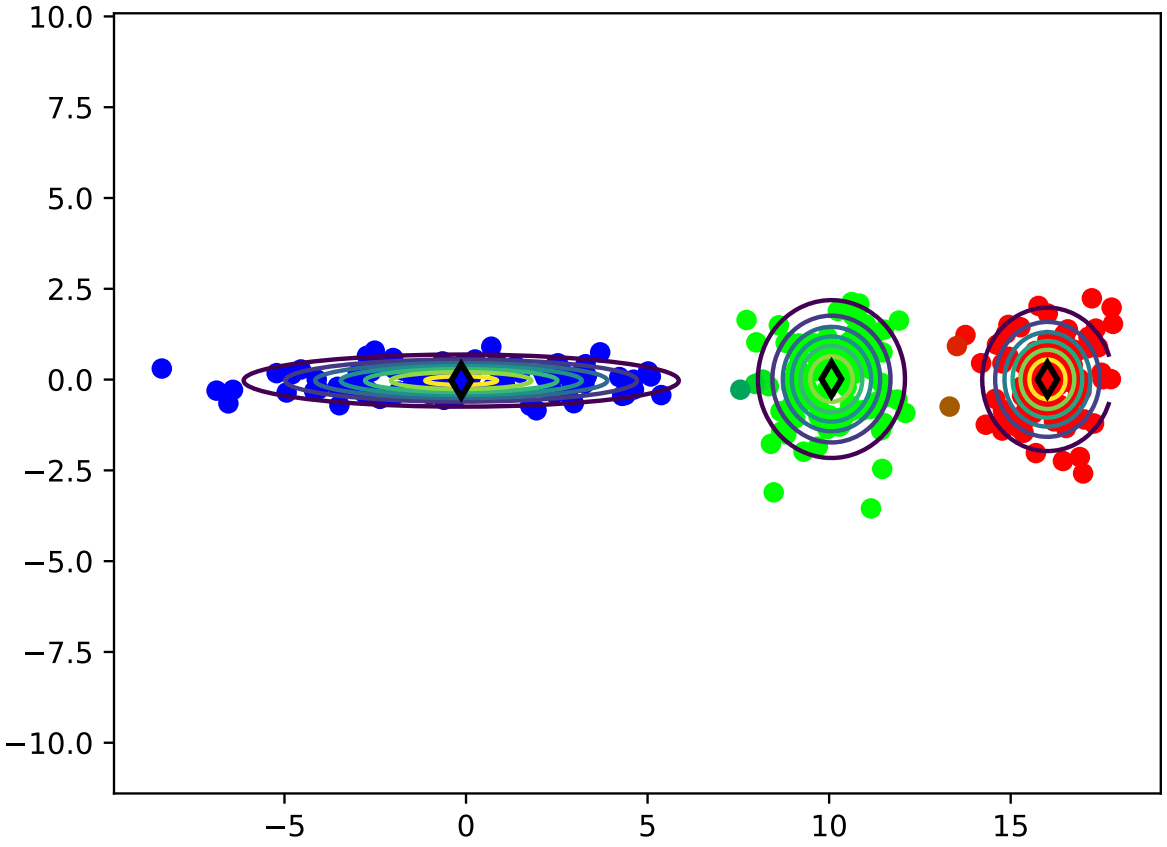
Gaussian Mixture Model with diagonal covariances.



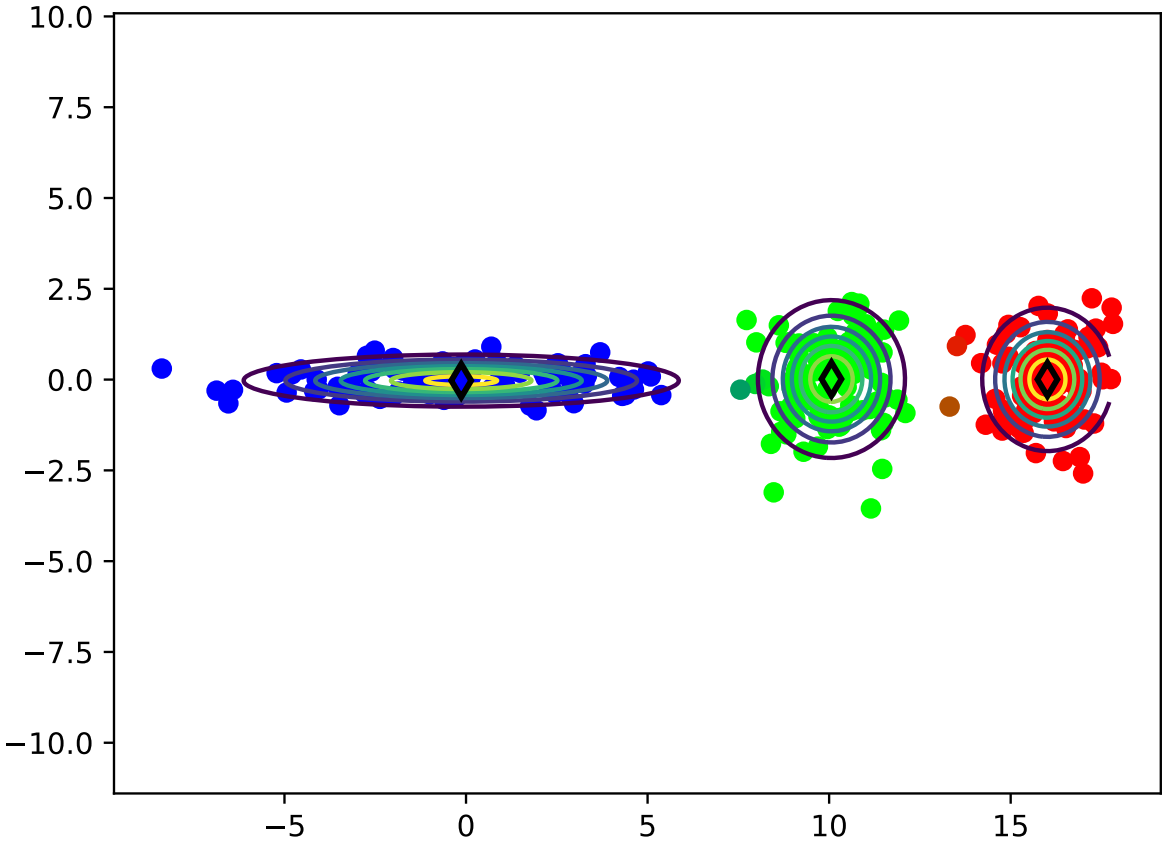
Gaussian Mixture Model with diagonal covariances.



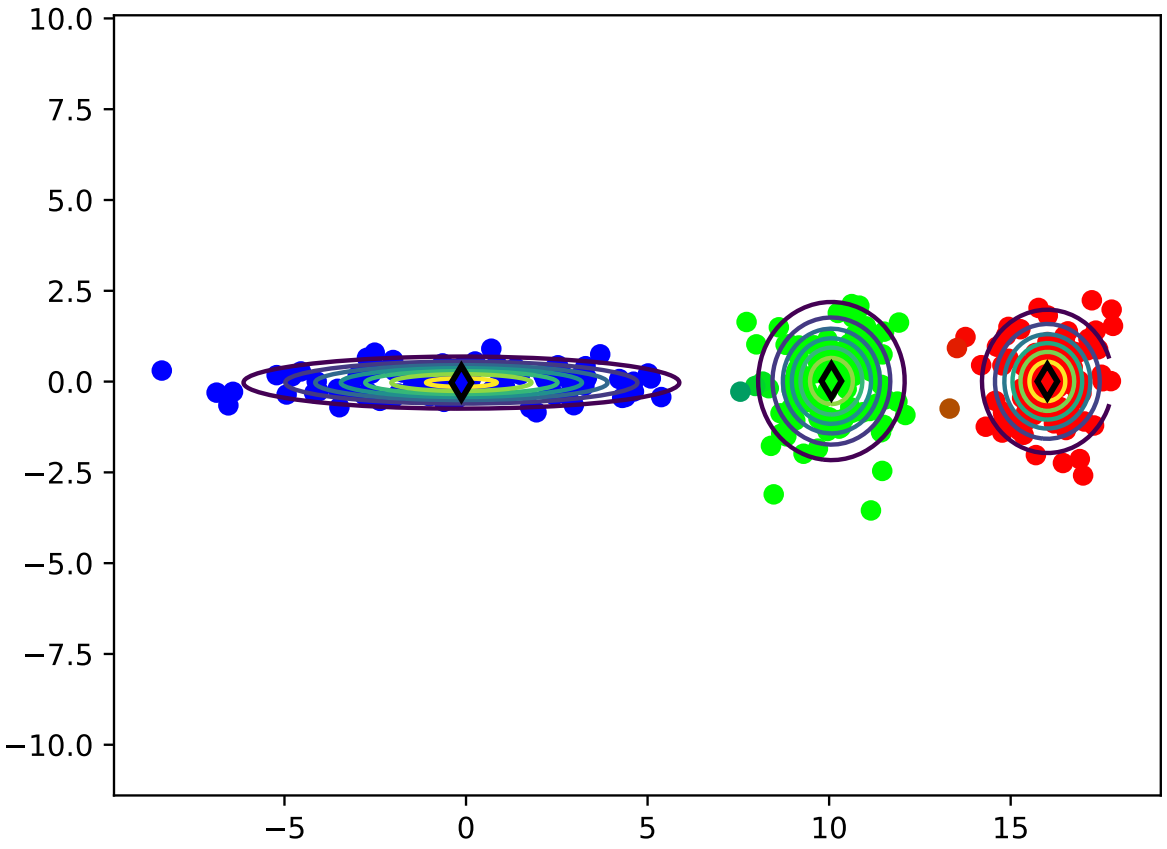
Gaussian Mixture Model with diagonal covariances.



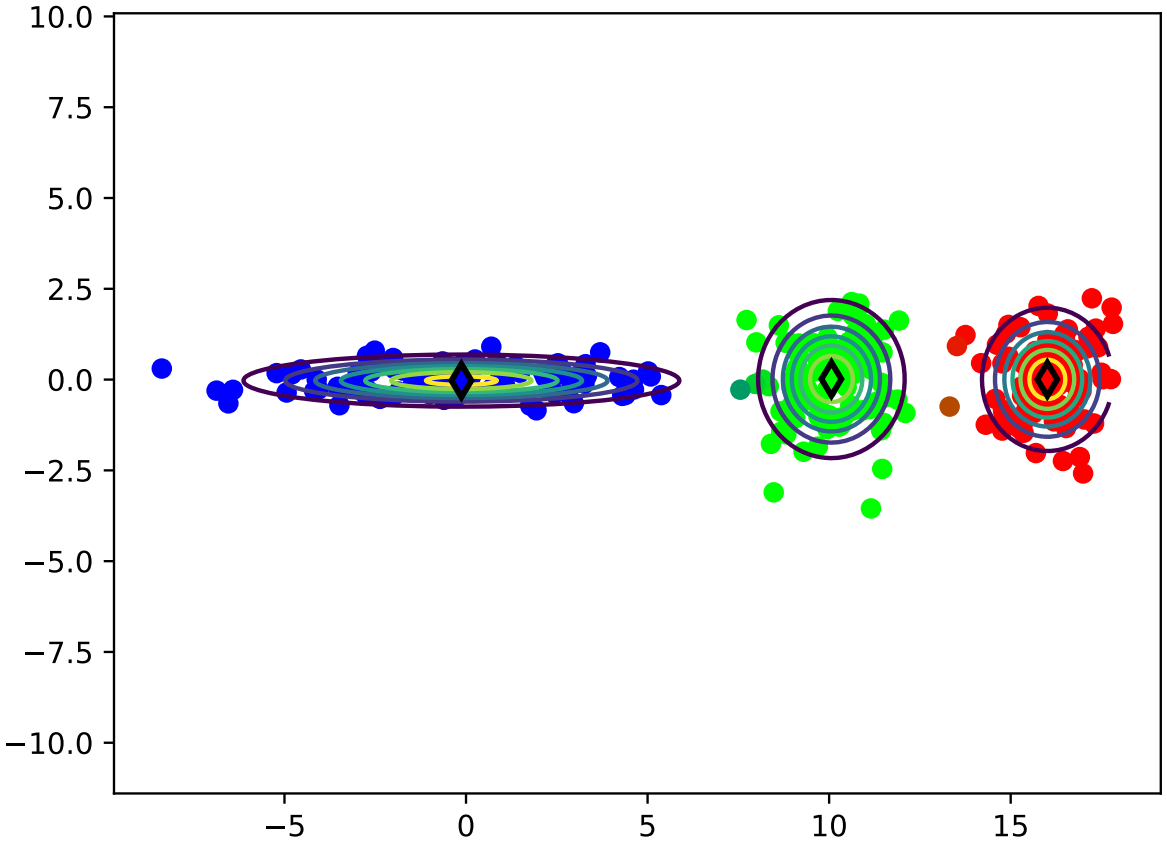
Gaussian Mixture Model with diagonal covariances.



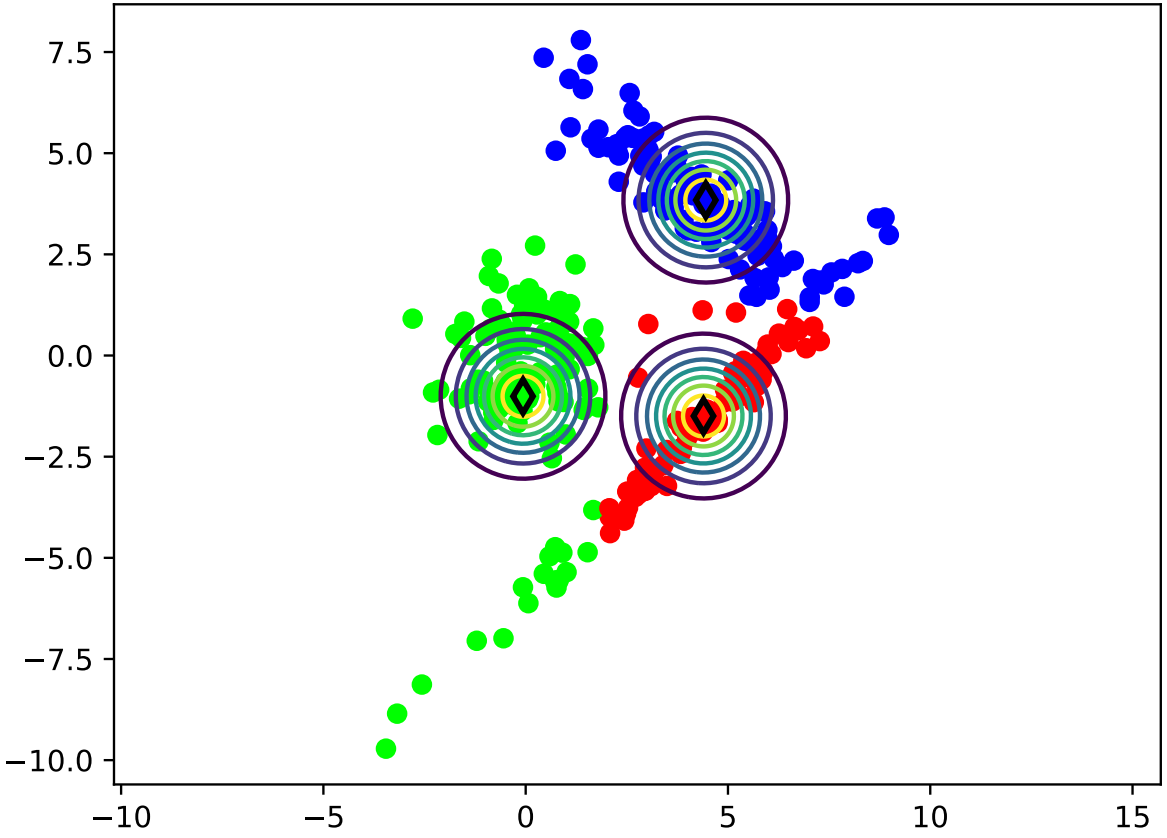
Gaussian Mixture Model with diagonal covariances.



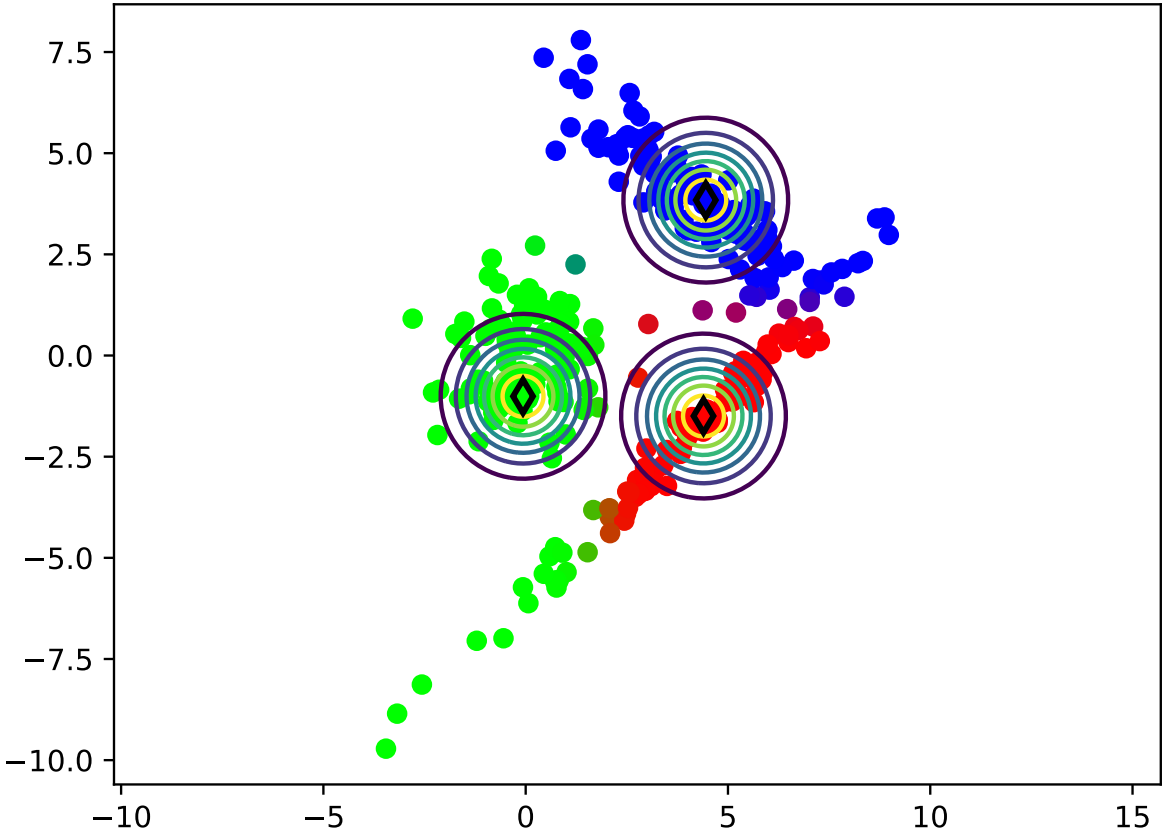
Gaussian Mixture Model with diagonal covariances.



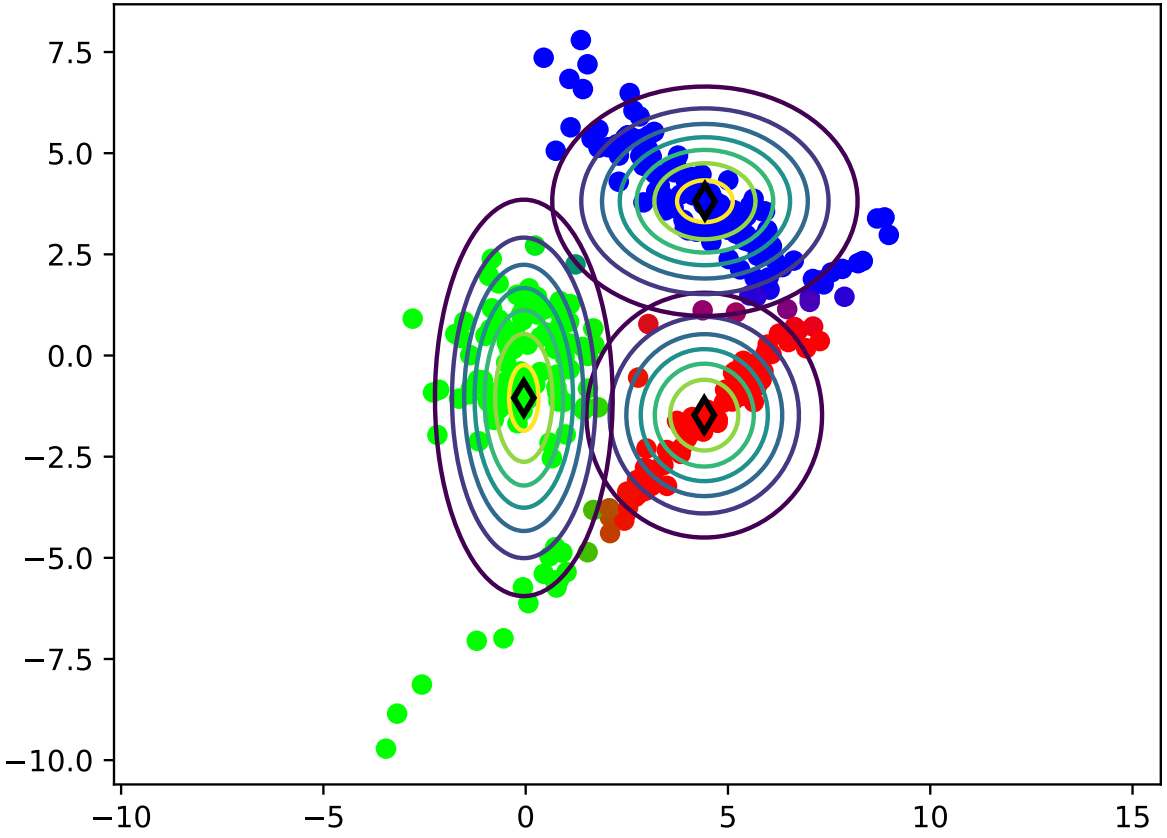
Gaussian Mixture Model with diagonal covariances.



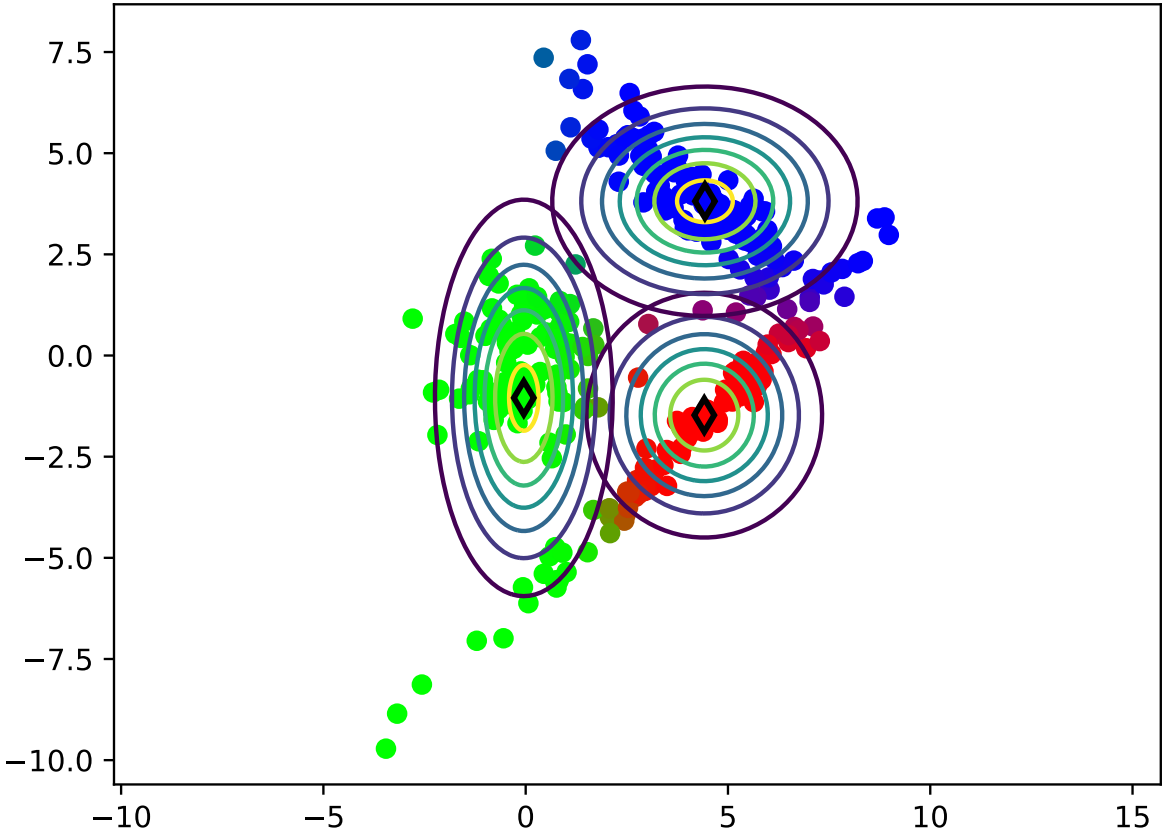
Gaussian Mixture Model with diagonal covariances.



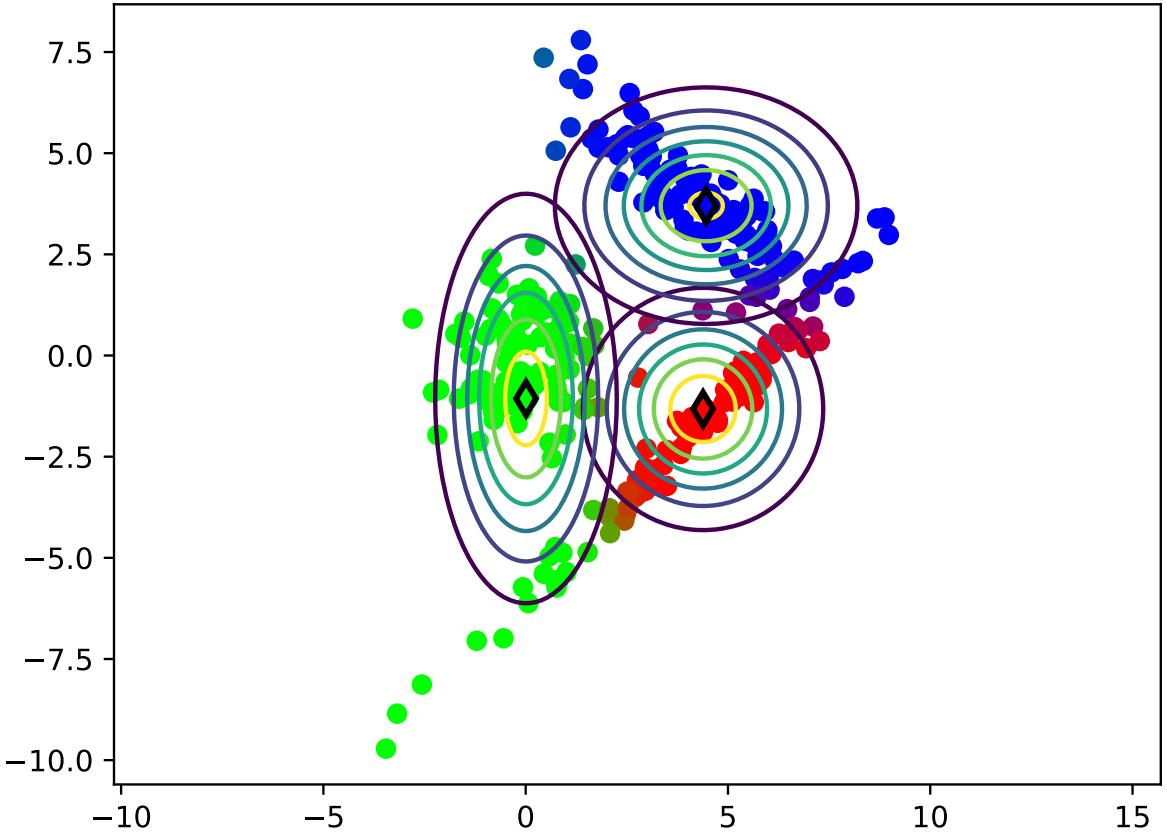
Gaussian Mixture Model with diagonal covariances.



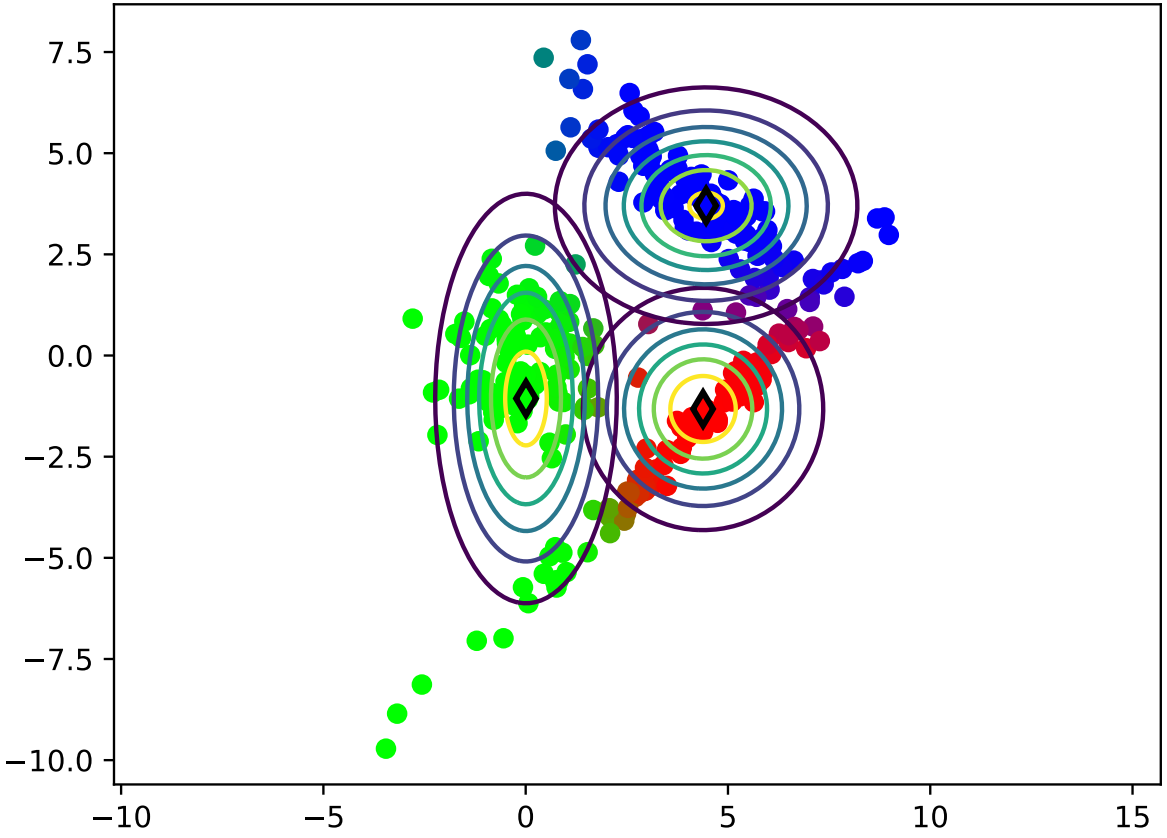
Gaussian Mixture Model with diagonal covariances.



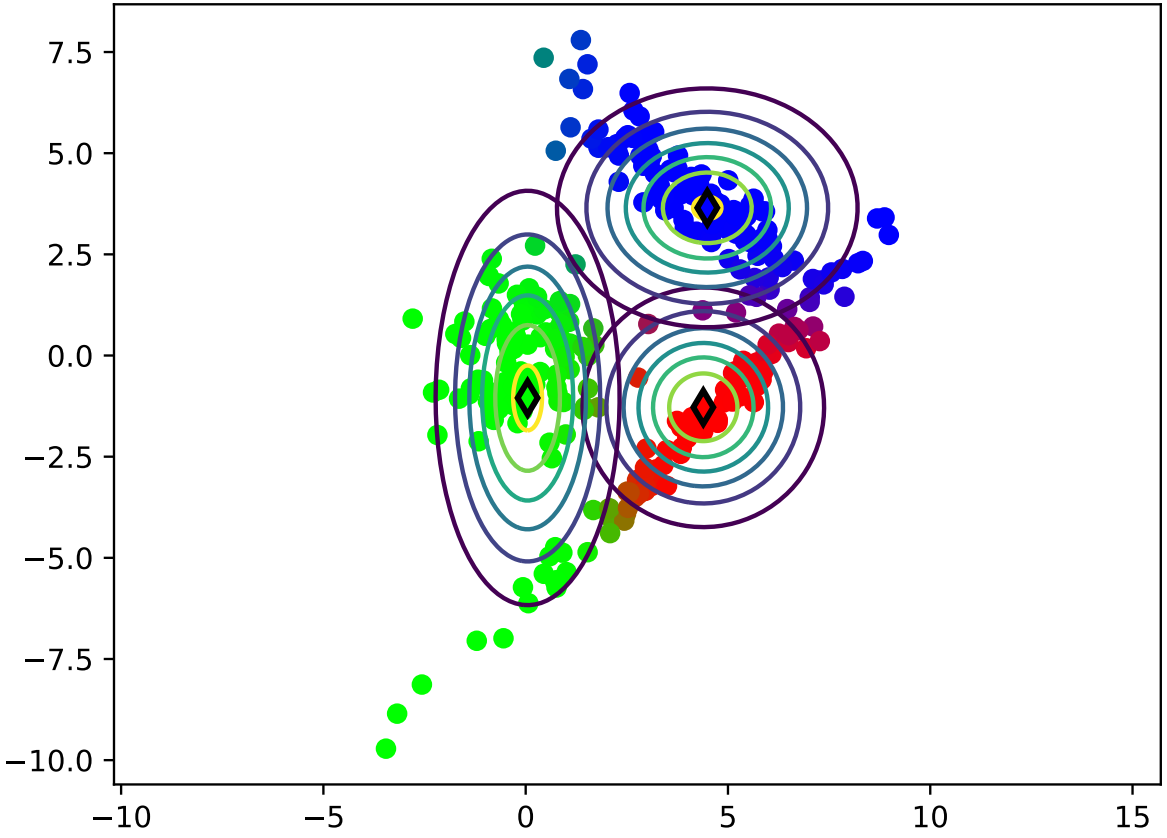
Gaussian Mixture Model with diagonal covariances.



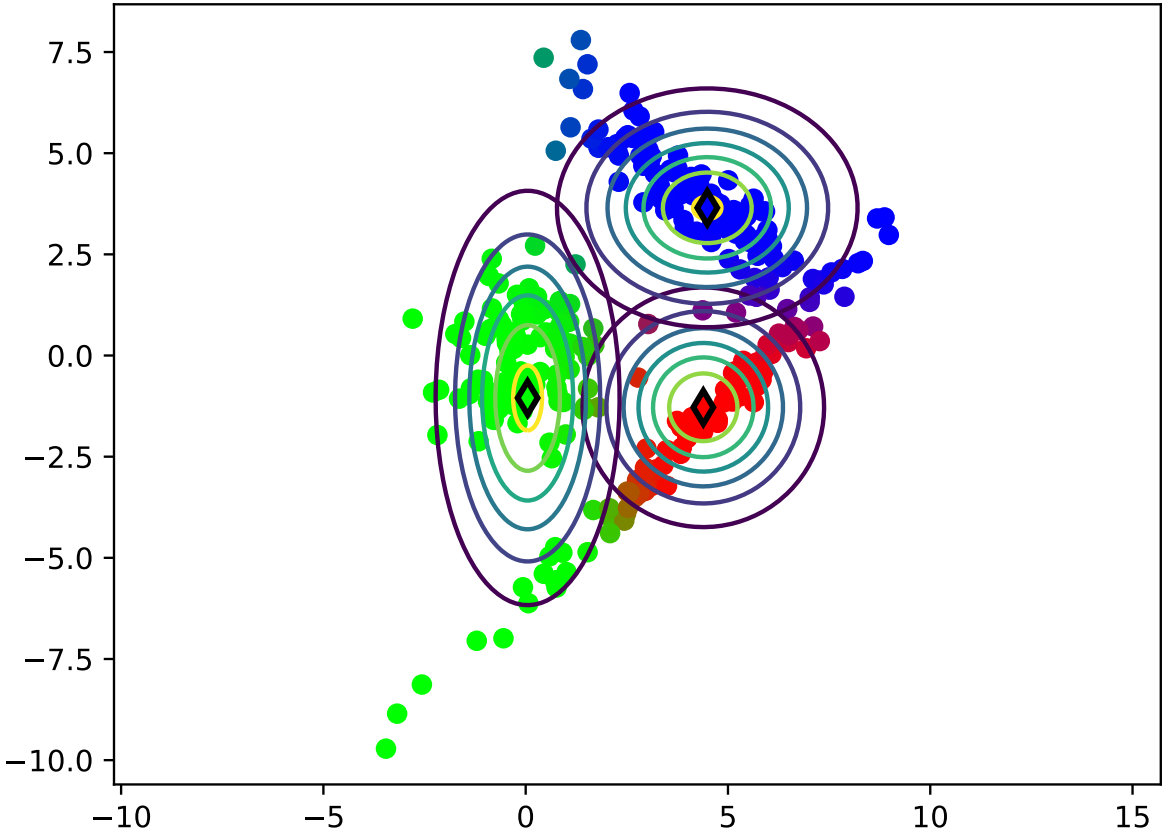
Gaussian Mixture Model with diagonal covariances.



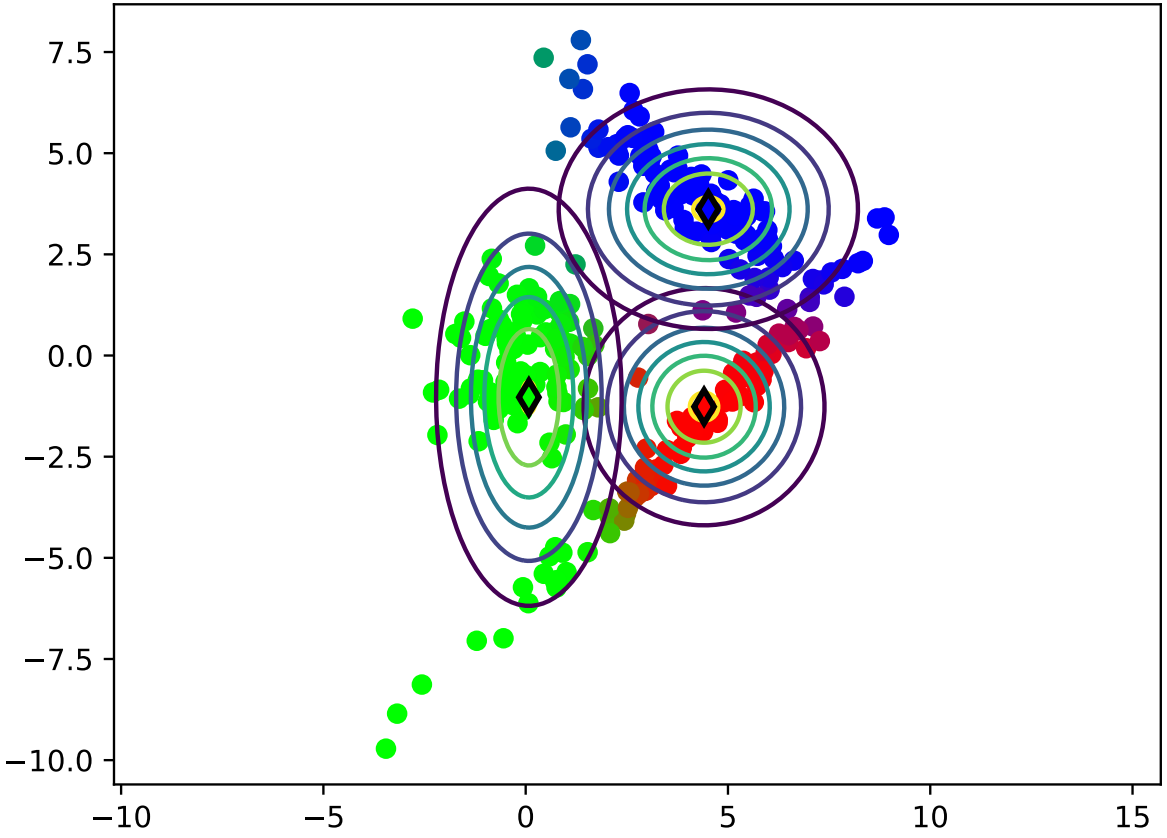
Gaussian Mixture Model with diagonal covariances.



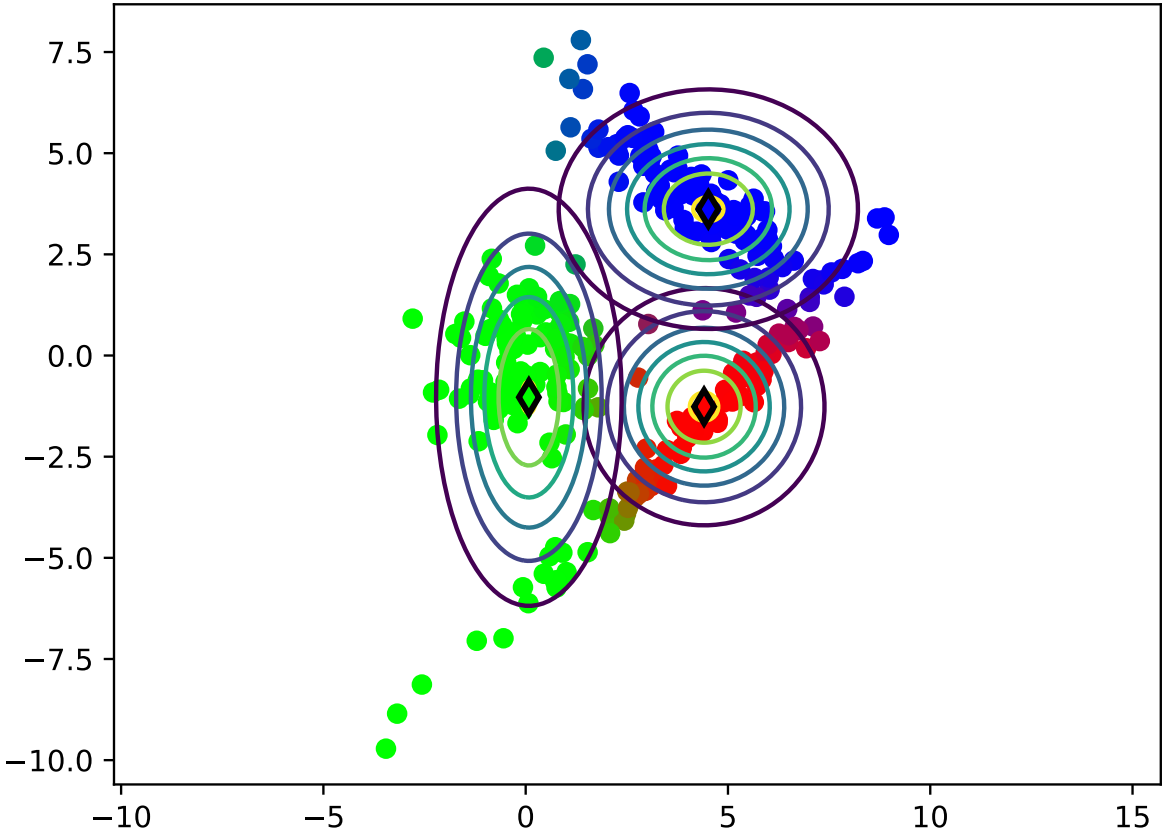
Gaussian Mixture Model with diagonal covariances.



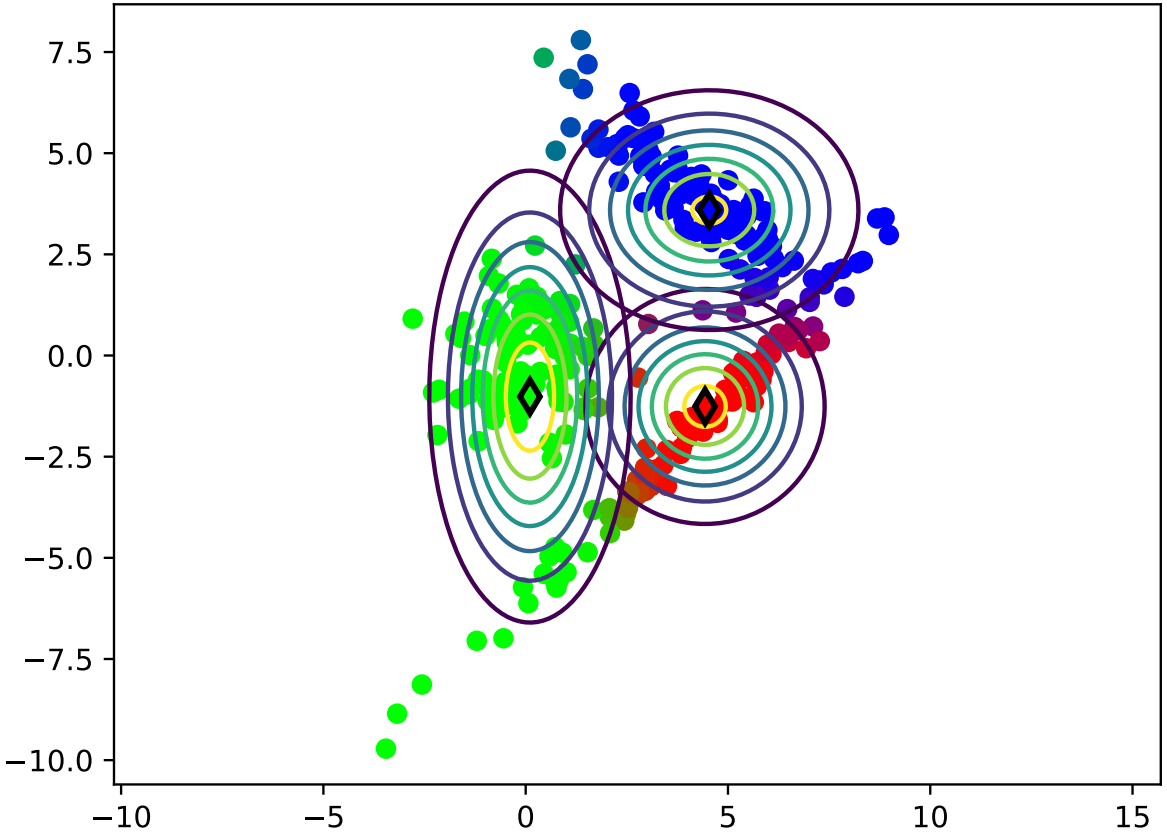
Gaussian Mixture Model with diagonal covariances.



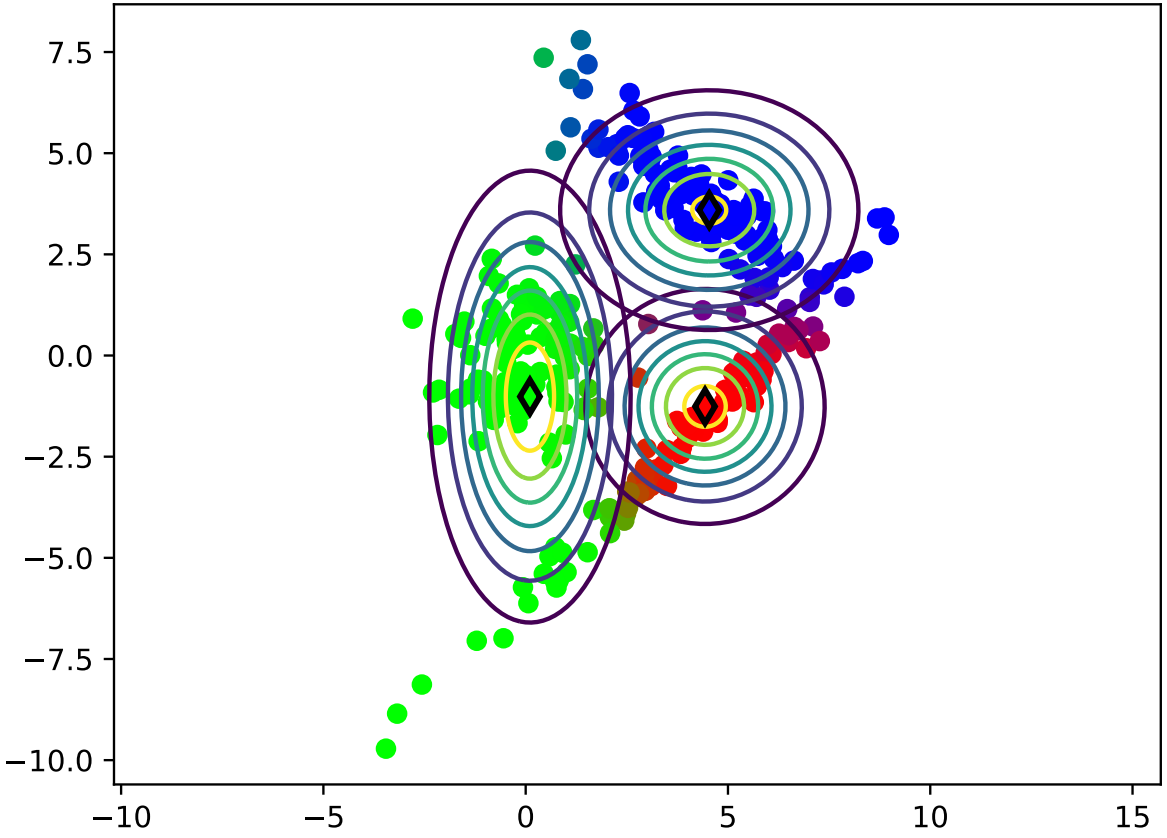
Gaussian Mixture Model with diagonal covariances.



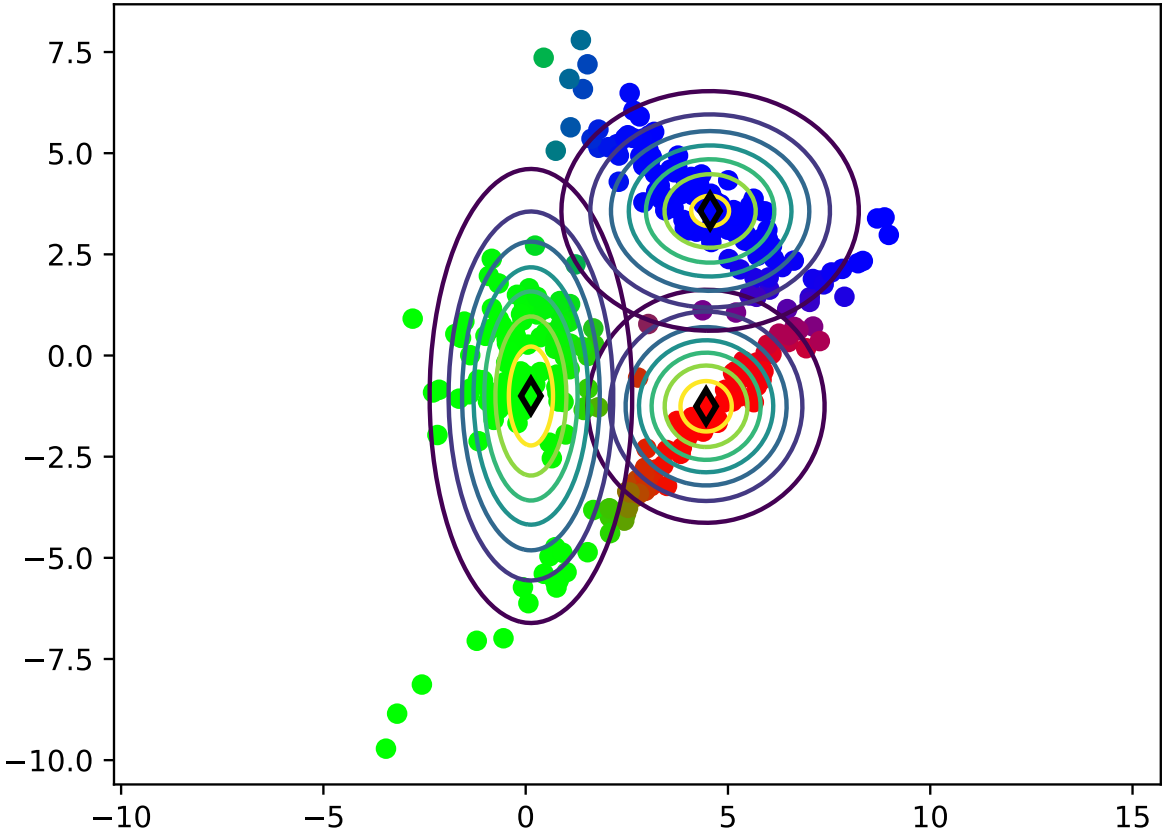
Gaussian Mixture Model with diagonal covariances.



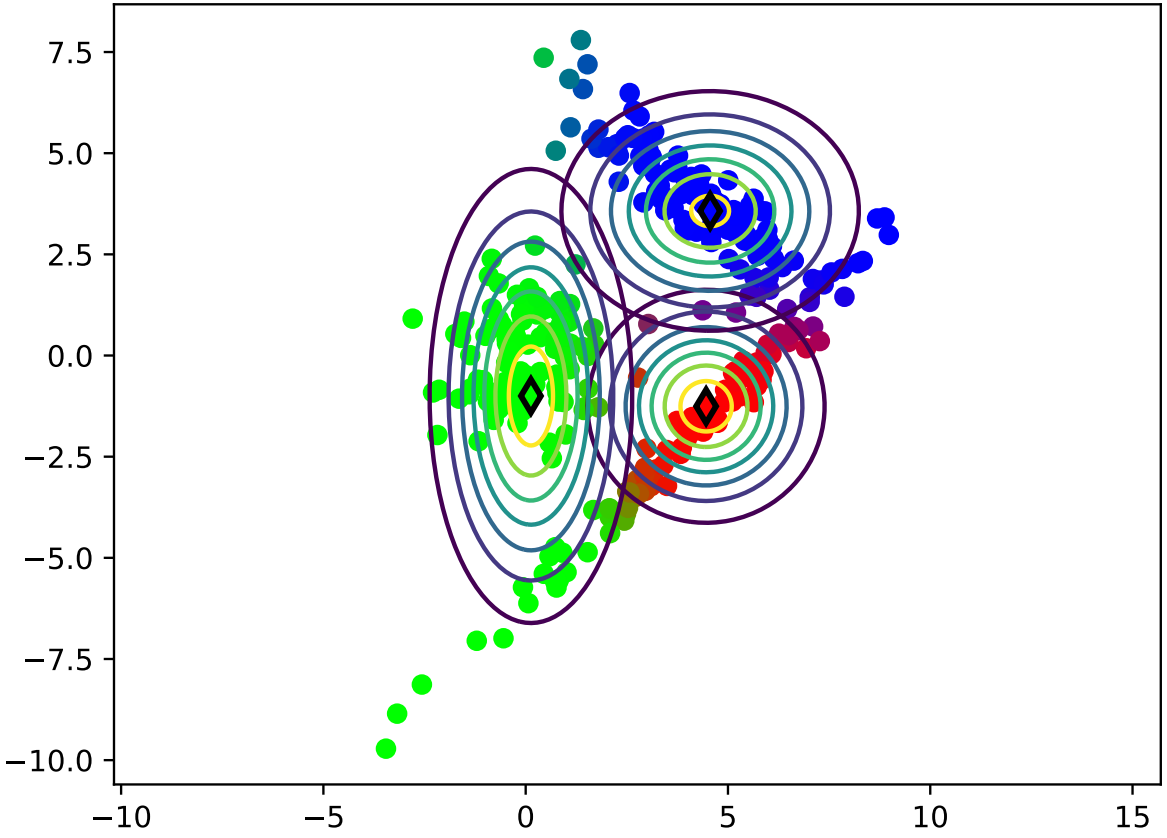
Gaussian Mixture Model with diagonal covariances.



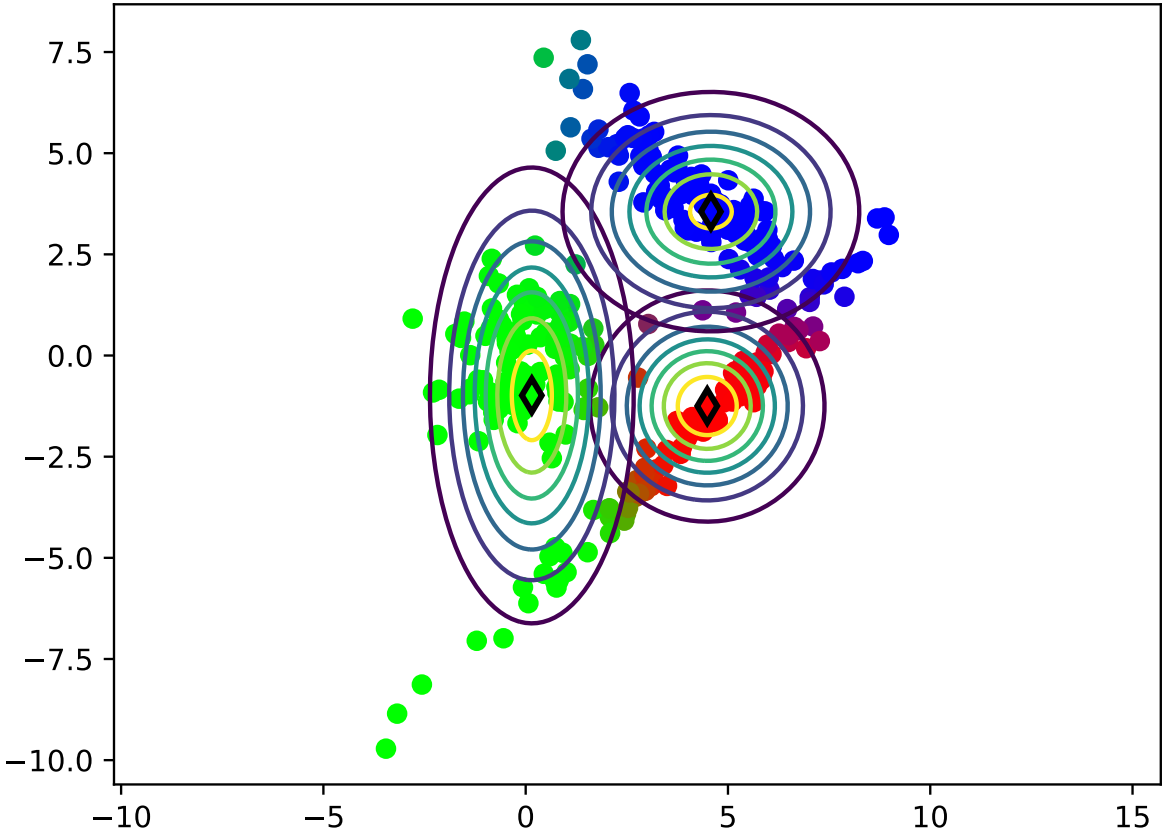
Gaussian Mixture Model with diagonal covariances.



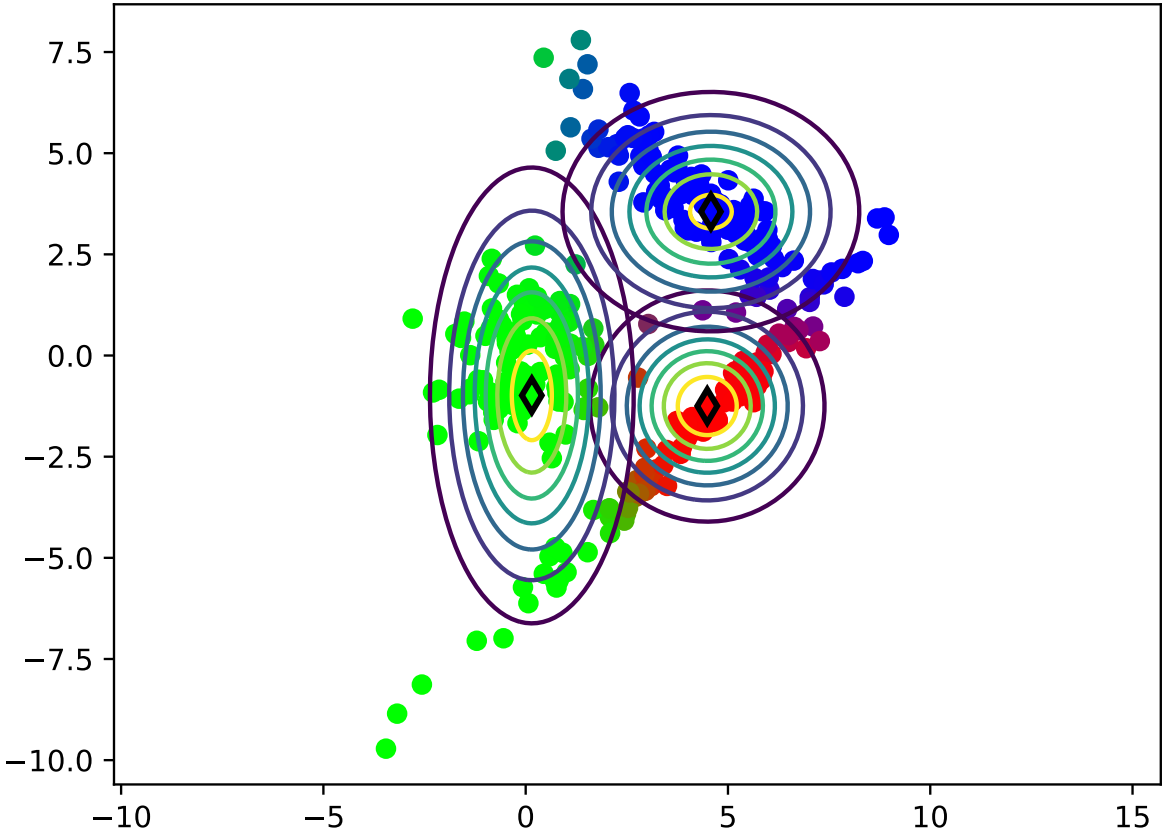
Gaussian Mixture Model with diagonal covariances.



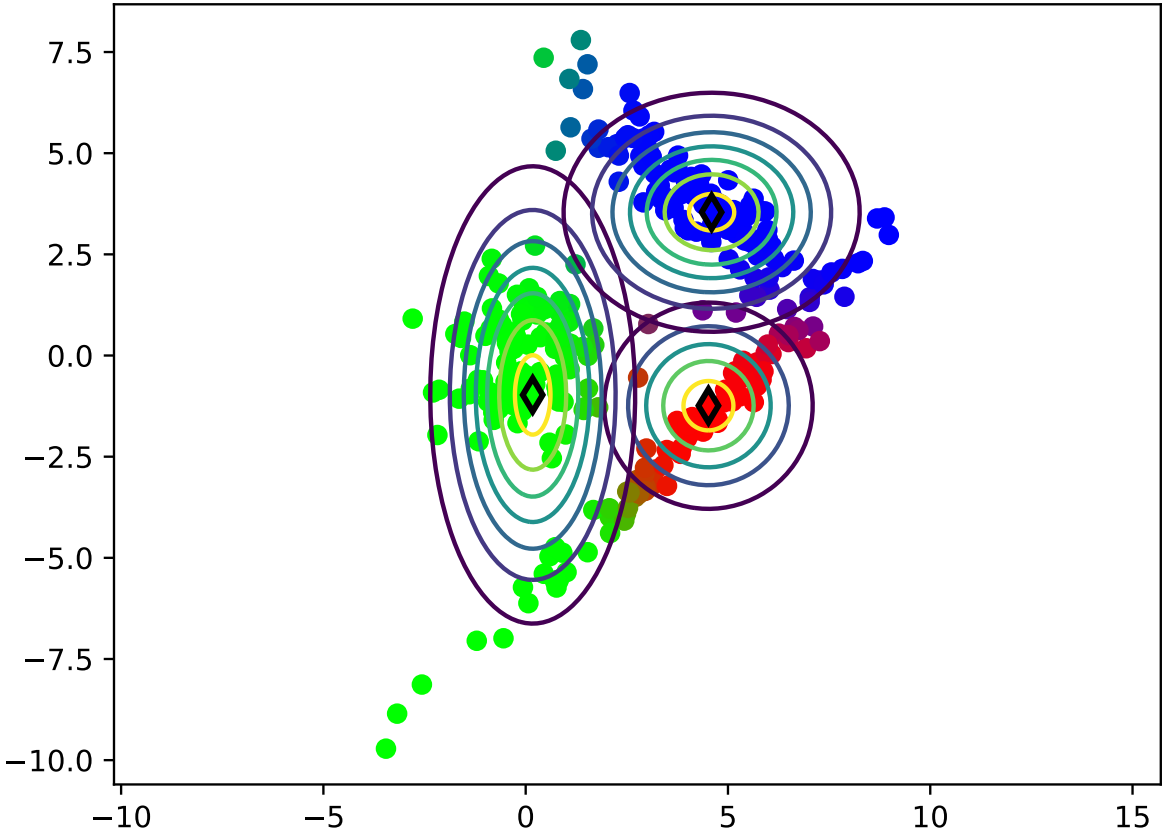
Gaussian Mixture Model with diagonal covariances.



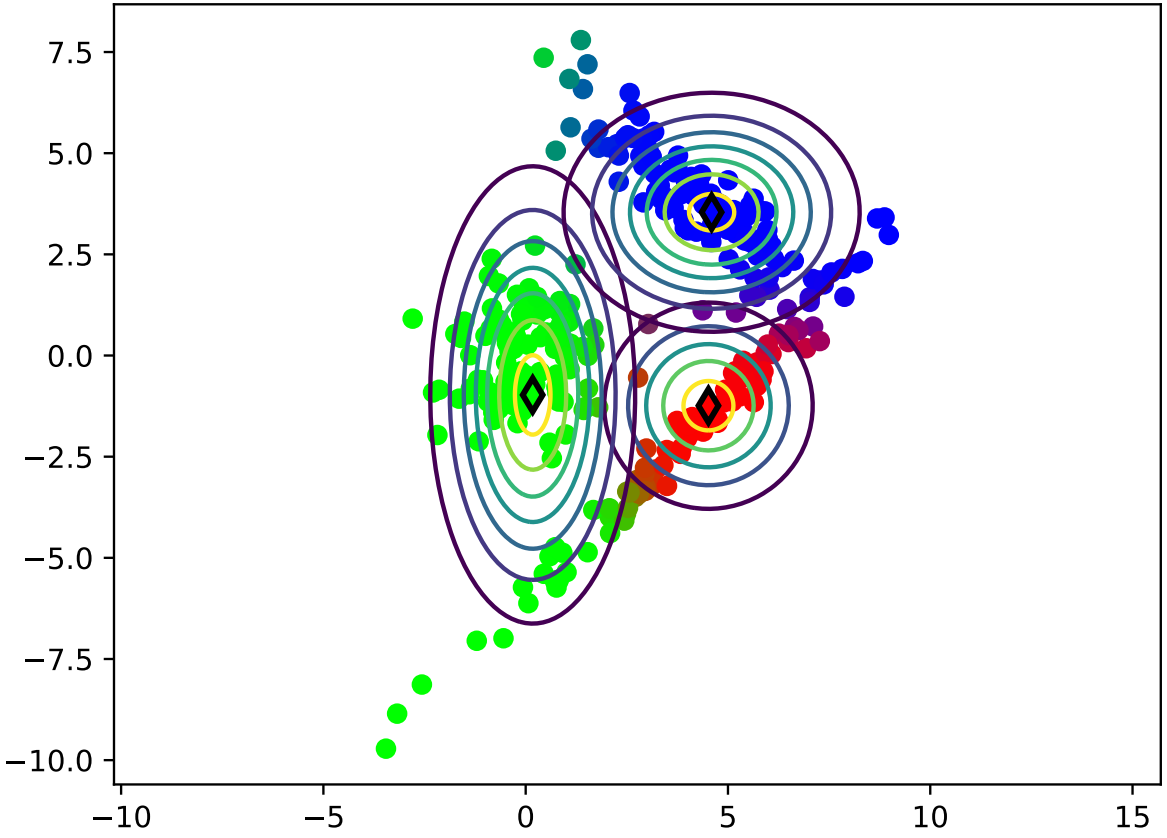
Gaussian Mixture Model with diagonal covariances.



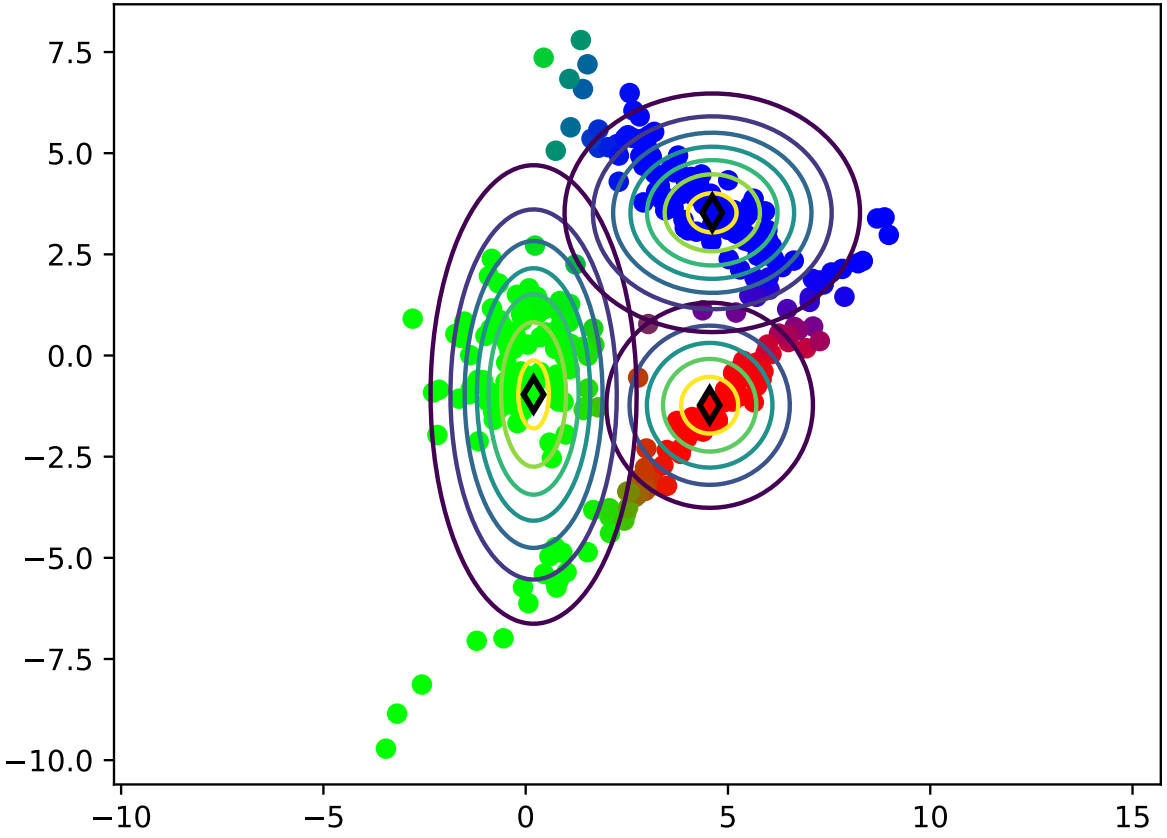
Gaussian Mixture Model with diagonal covariances.



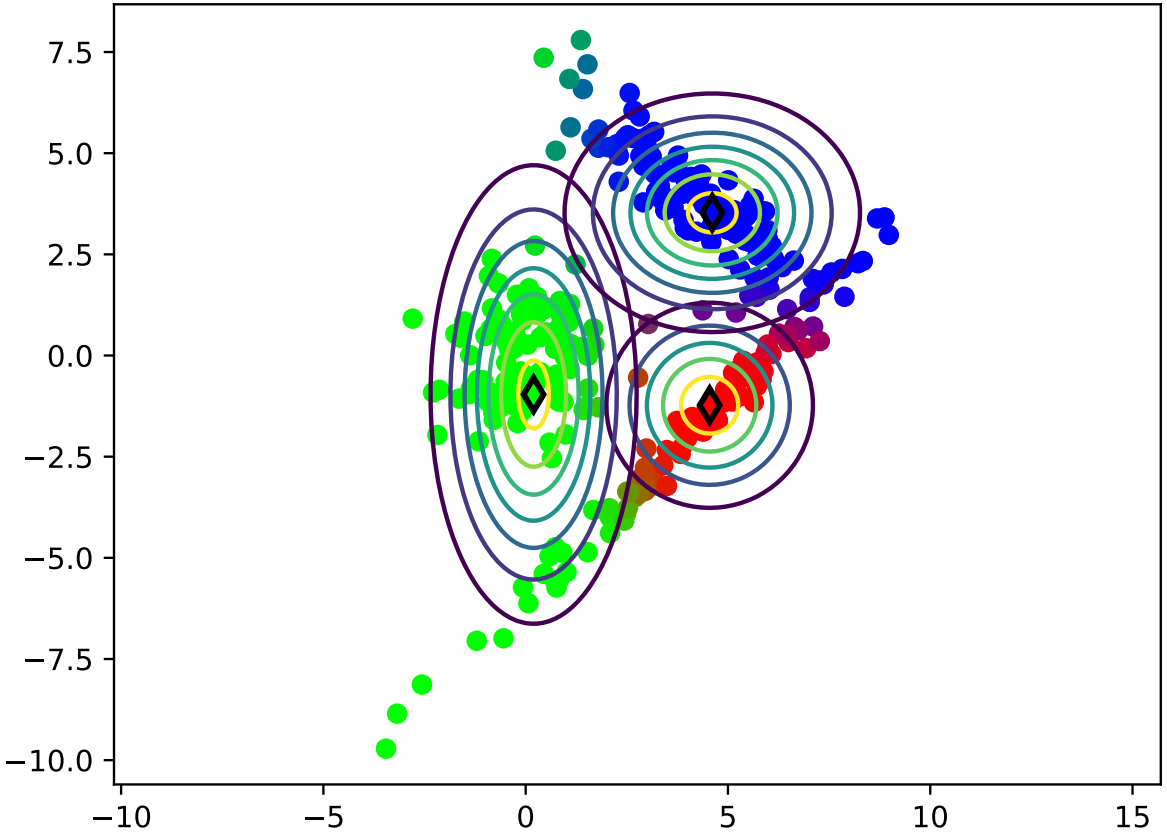
Gaussian Mixture Model with diagonal covariances.



Gaussian Mixture Model with diagonal covariances.



Gaussian Mixture Model with diagonal covariances.



Singularities

E-M with GMMs suffers from **singularities**:
trivial situations where the likelihood goes to ∞ but the solution is bad.

- Suppose:
  $d = 1$, $k = 2$, $\pi_j = 1/2$,
  $n = 3$ with $\boldsymbol{x}_1 = -1$ and $\boldsymbol{x}_2 = +1$ and $\boldsymbol{x}_3 = +3$.
  Initialize with $\boldsymbol{\mu}_1 = 0$ and $\boldsymbol{\sigma}_1 = 1$,
  but $\boldsymbol{\mu}_2 = +3 = \boldsymbol{x}_3$ and $\boldsymbol{\sigma}_2 = 1/100$.
  Then $\boldsymbol{\sigma}_2 \to 0$ and $\mathcal{L} \uparrow \infty$.

Interpolating between k-means and GMM E-M

Same M-step: fix $\boldsymbol{\pi} = (1/k, \ldots, 1/k)$ and $\boldsymbol{\Sigma}_j = c\boldsymbol{I}$ for a fixed $c > 0$.

Interpolating between k-means and GMM E-M

Same M-step: fix $\boldsymbol{\pi} = (1/k, \ldots, 1/k)$ and $\boldsymbol{\Sigma}_j = c\boldsymbol{I}$ for a fixed $c > 0$.

Same E-step: define $q_{ij} := \frac{1}{2}\|\boldsymbol{x}_i - \boldsymbol{\mu}_j\|^2$; the E-step chooses

$$R_{ij} := p_{\boldsymbol{\theta}}(y_i = j|\boldsymbol{x}_i) = \frac{p_{\boldsymbol{\theta}}(y_i = j, \boldsymbol{x}_i)}{p_{\boldsymbol{\theta}}(\boldsymbol{x}_i)} = \frac{p_{\boldsymbol{\theta}}(y_i = j, \boldsymbol{x}_i)}{\sum_{l=1}^k p_{\boldsymbol{\theta}}(y_i = l, \boldsymbol{x}_i)}$$

$$= \frac{\pi_j p_{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j}(\boldsymbol{x}_i)}{\sum_{l=1}^k \pi_l p_{\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l}(\boldsymbol{x}_i)} = \frac{\exp(-q_{ij}/c)}{\sum_{l=1}^k \exp(-q_{il}/c)}$$

Fix $i \in \{1, \ldots, n\}$ and suppose minimum $q_i := \min_j q_{ij}$ is unique:

Interpolating between k-means and GMM E-M

Same M-step: fix $\boldsymbol{\pi} = (1/k, \ldots, 1/k)$ and $\boldsymbol{\Sigma}_j = c\boldsymbol{I}$ for a fixed $c > 0$.

Same E-step: define $q_{ij} := \frac{1}{2}\|\boldsymbol{x}_i - \boldsymbol{\mu}_j\|^2$; the E-step chooses

$$R_{ij} := p_{\boldsymbol{\theta}}(y_i = j|\boldsymbol{x}_i) = \frac{p_{\boldsymbol{\theta}}(y_i = j, \boldsymbol{x}_i)}{p_{\boldsymbol{\theta}}(\boldsymbol{x}_i)} = \frac{p_{\boldsymbol{\theta}}(y_i = j, \boldsymbol{x}_i)}{\sum_{l=1}^k p_{\boldsymbol{\theta}}(y_i = l, \boldsymbol{x}_i)}$$

$$= \frac{\pi_j p_{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j}(\boldsymbol{x}_i)}{\sum_{l=1}^k \pi_l p_{\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l}(\boldsymbol{x}_i)} = \frac{\exp(-q_{ij}/c)}{\sum_{l=1}^k \exp(-q_{il}/c)}$$

Fix $i \in \{1, \ldots, n\}$ and suppose minimum $q_i := \min_j q_{ij}$ is unique:

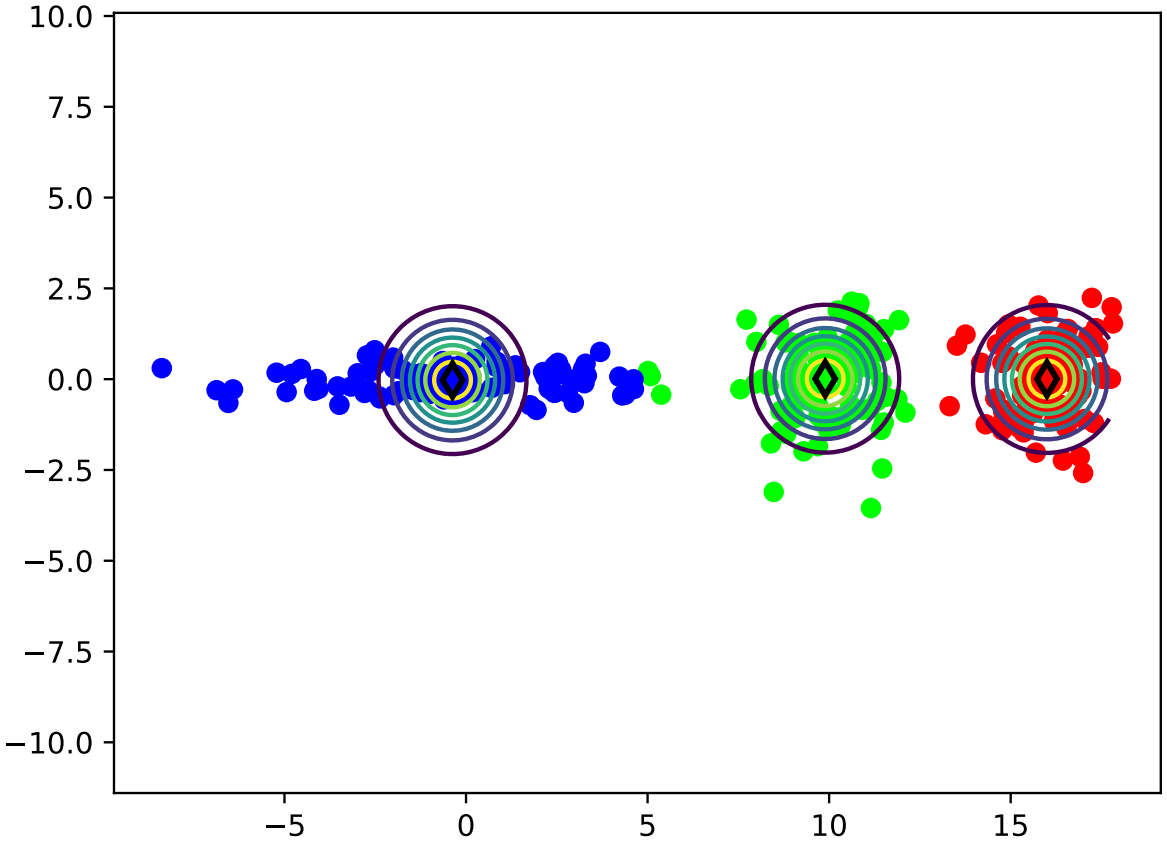
$$\lim_{c \downarrow 0} R_{ij} = \lim_{c \downarrow 0} \frac{\exp(-q_{ij}/c)}{\sum_{l=1}^k \exp(-q_{il}/c)} = \lim_{c \downarrow 0} \frac{\exp(q_i - q_{ij}/c)}{\sum_{l=1}^k \exp(q_i - q_{il}/c)}$$

$$= \begin{cases} 1 & q_{ij} = q_i, \\ 0 & q_{ij} \neq q_i. \end{cases}$$

That is, R becomes hard assignment A as $c \downarrow 0$.

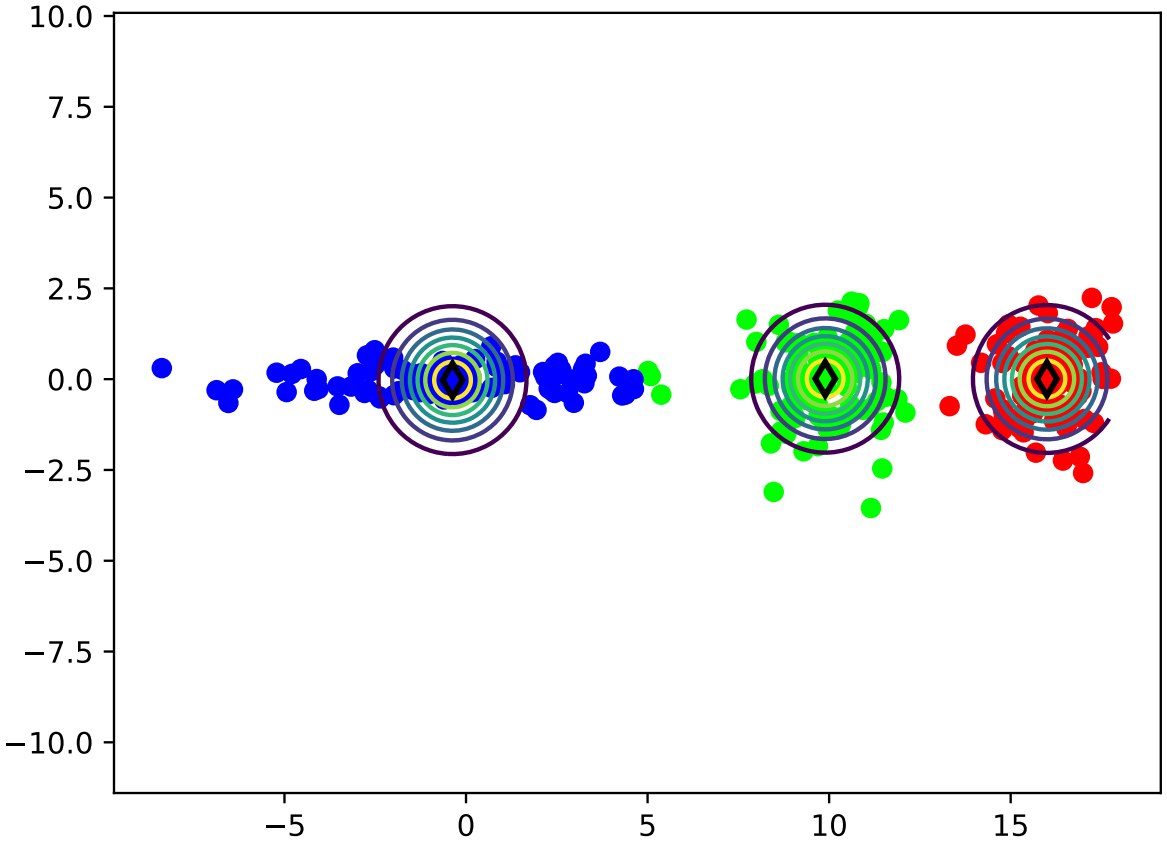
Interpolating between k-means and GMM E-M (part 2)

We can interpolate algorithmically,
meaning we can create algorithms that have elements of both.
Here's something like k-means but with weights and covariances.



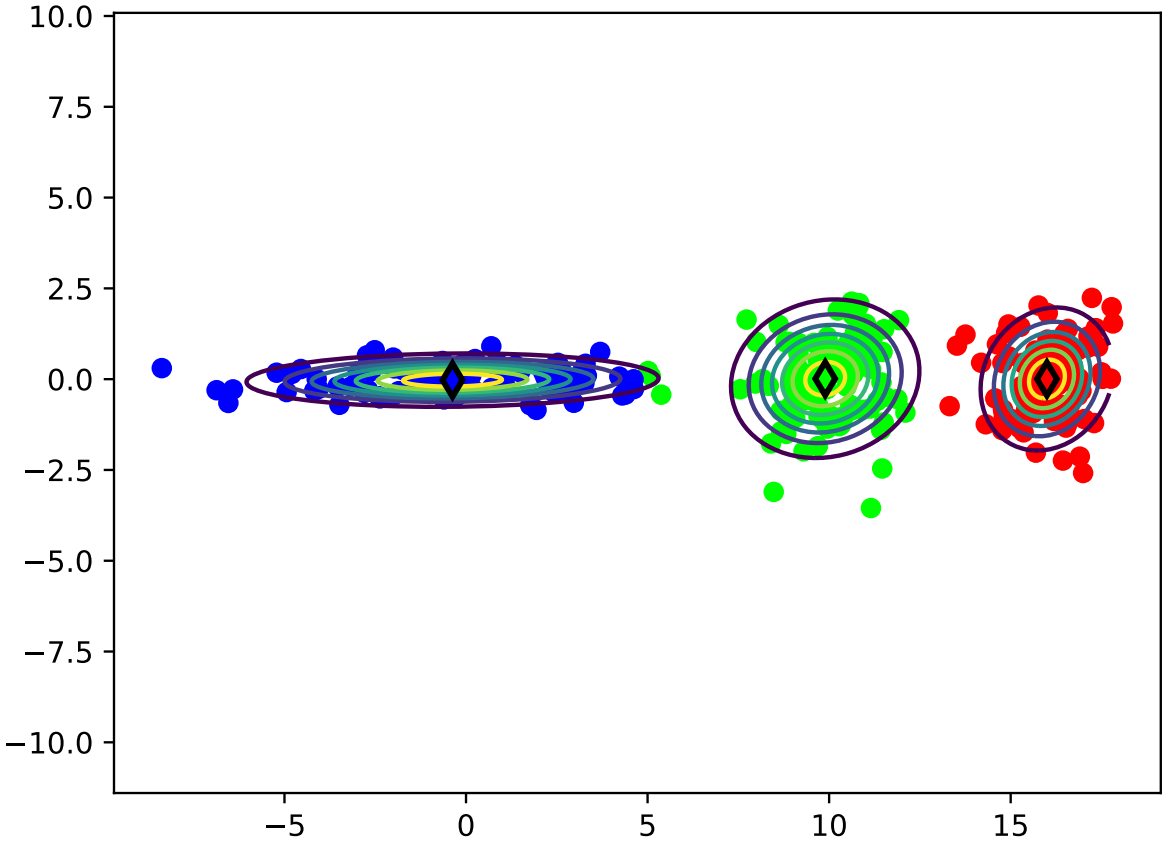
Interpolating between k-means and GMM E-M (part 2)

We can interpolate algorithmically,
meaning we can create algorithms that have elements of both.
Here's something like k-means but with weights and covariances.



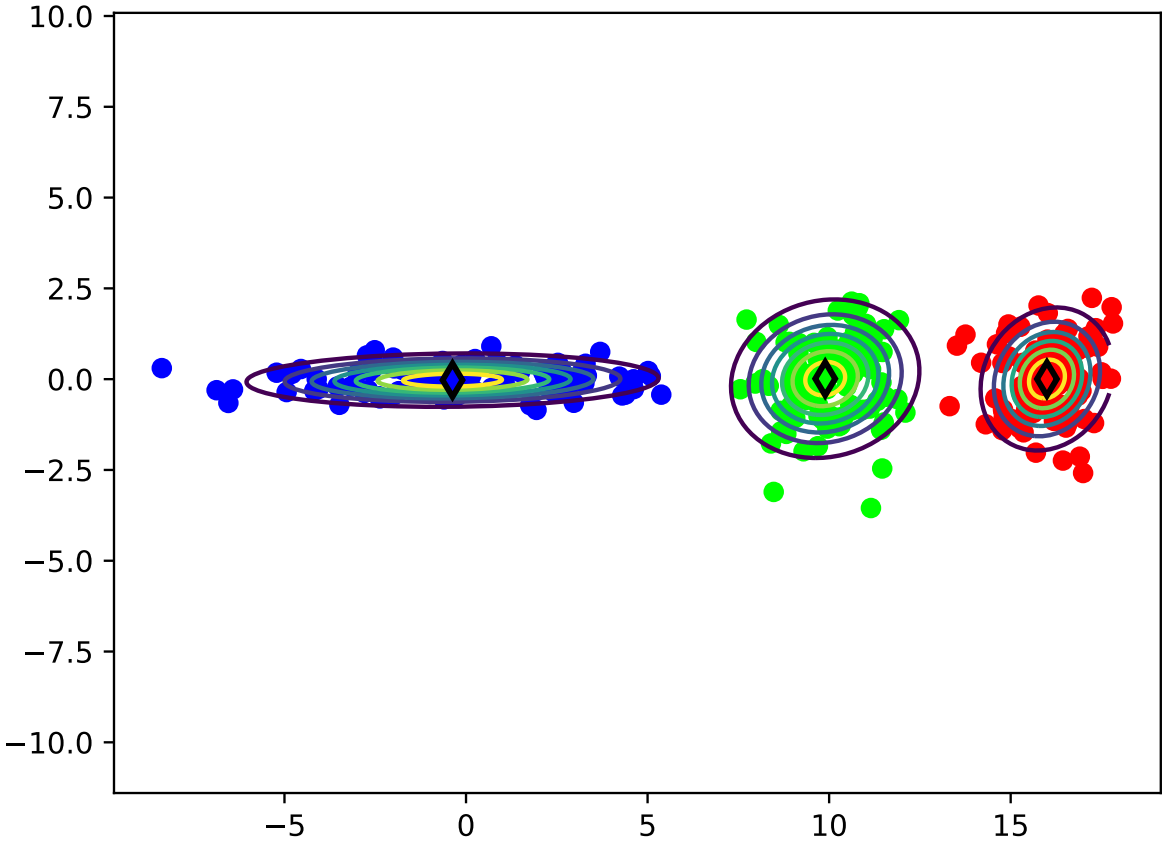
Interpolating between k-means and GMM E-M (part 2)

We can interpolate algorithmically,
meaning we can create algorithms that have elements of both.
Here's something like k-means but with weights and covariances.



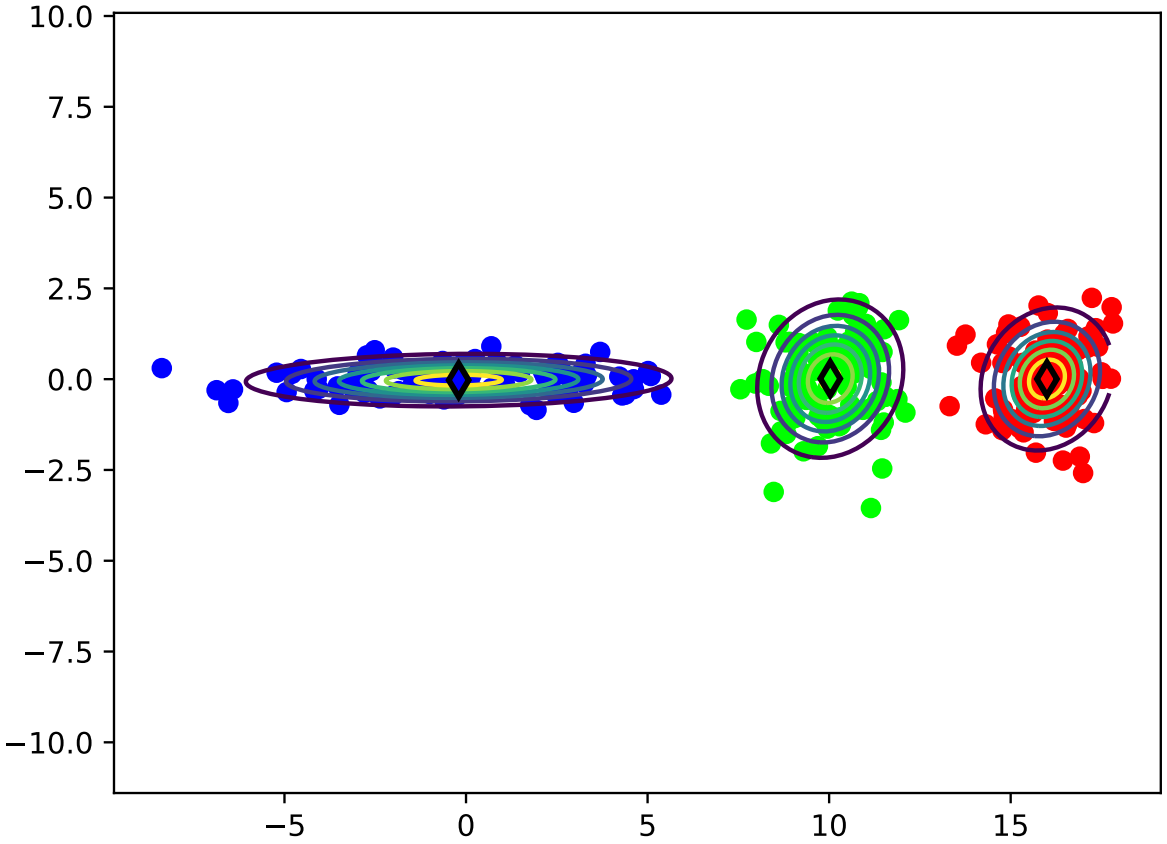
Interpolating between k-means and GMM E-M (part 2)

We can interpolate algorithmically,
meaning we can create algorithms that have elements of both.
Here's something like k-means but with weights and covariances.



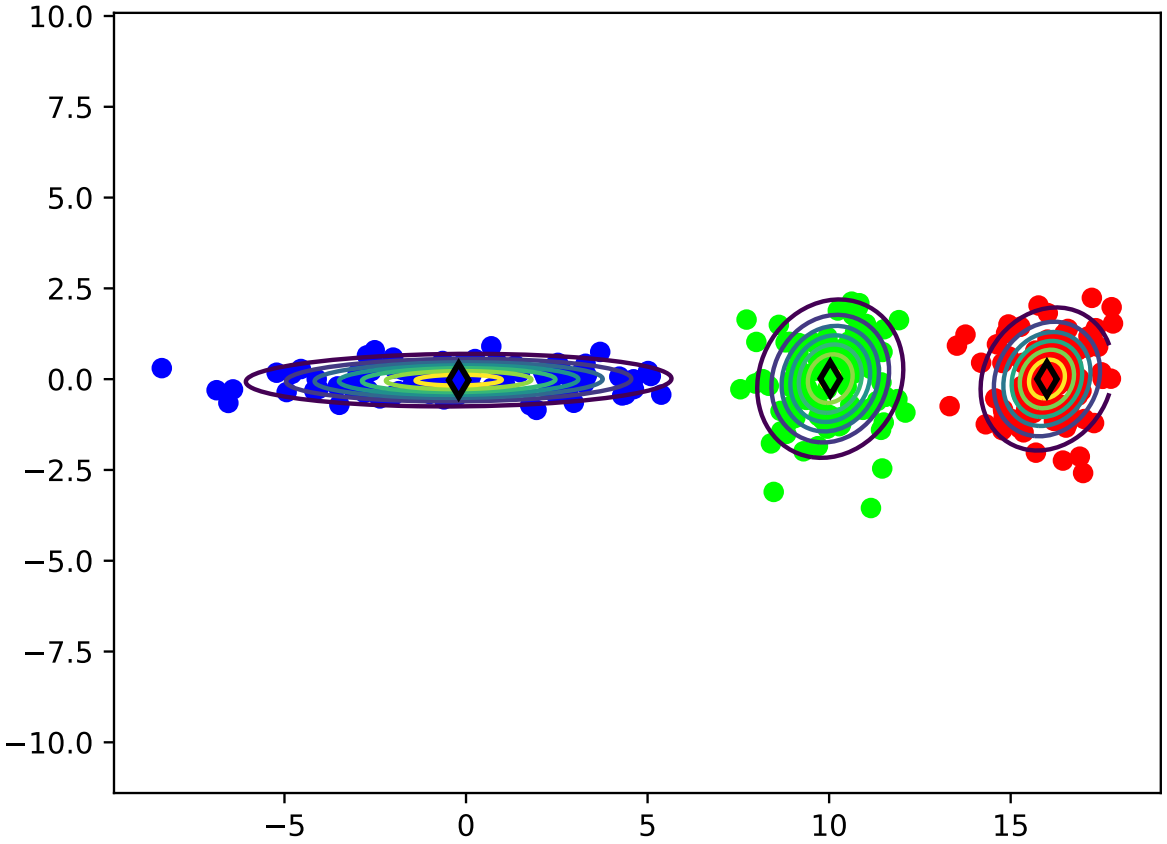
Interpolating between k-means and GMM E-M (part 2)

We can interpolate algorithmically,
meaning we can create algorithms that have elements of both.
Here's something like k-means but with weights and covariances.



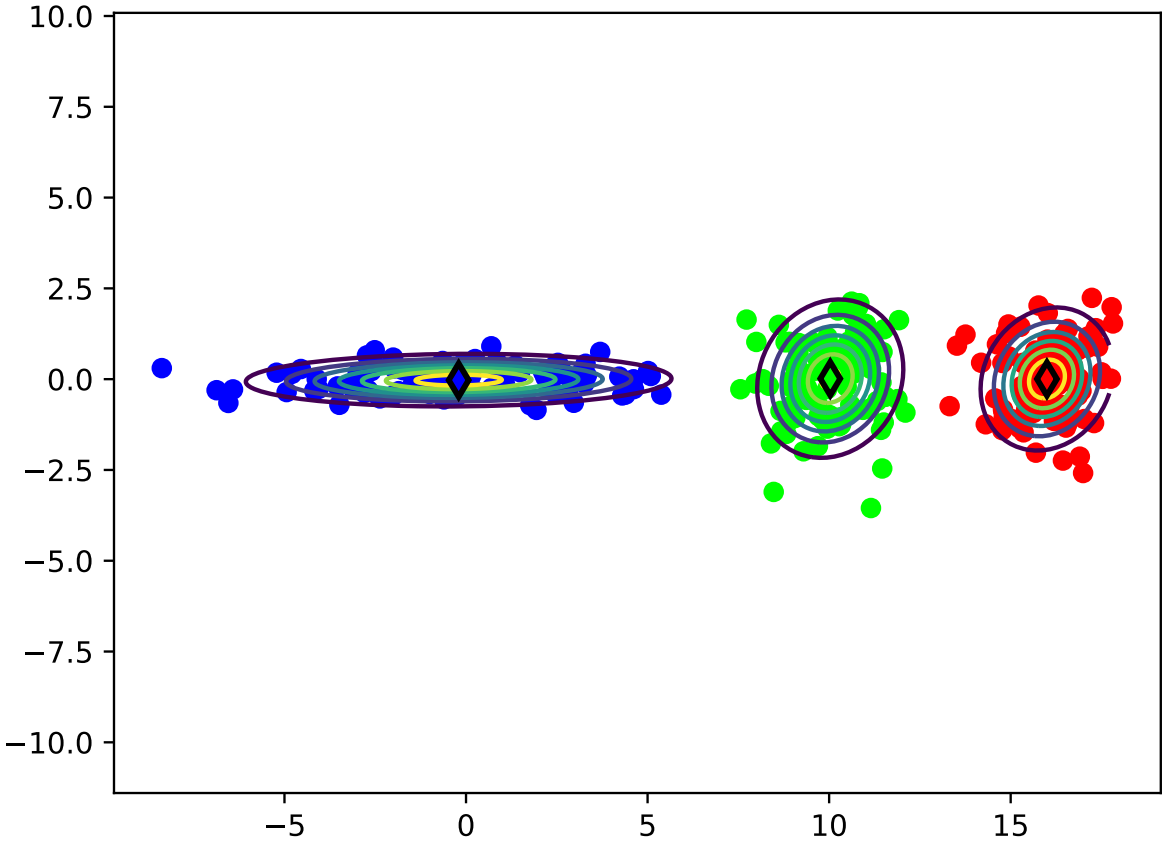
Interpolating between k-means and GMM E-M (part 2)

We can interpolate algorithmically,
meaning we can create algorithms that have elements of both.
Here's something like k-means but with weights and covariances.



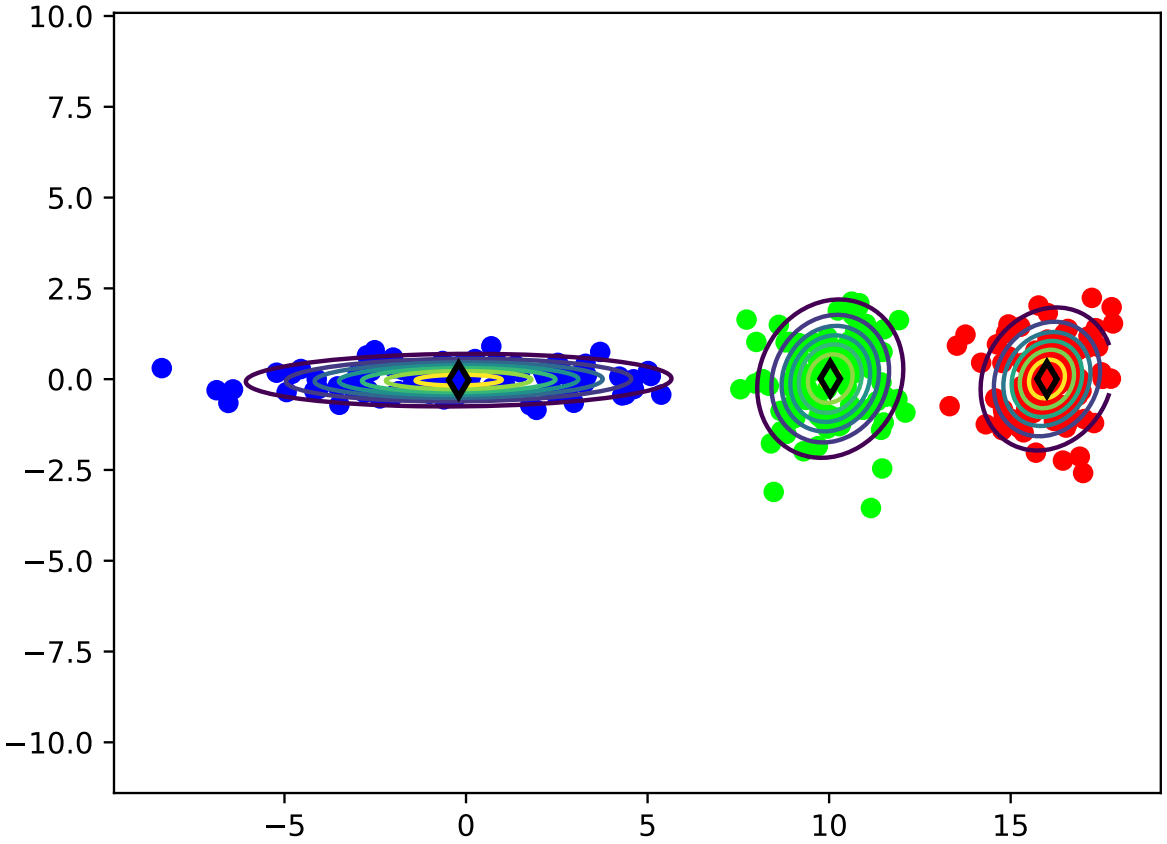
Interpolating between k-means and GMM E-M (part 2)

We can interpolate algorithmically,
meaning we can create algorithms that have elements of both.
Here's something like k-means but with weights and covariances.



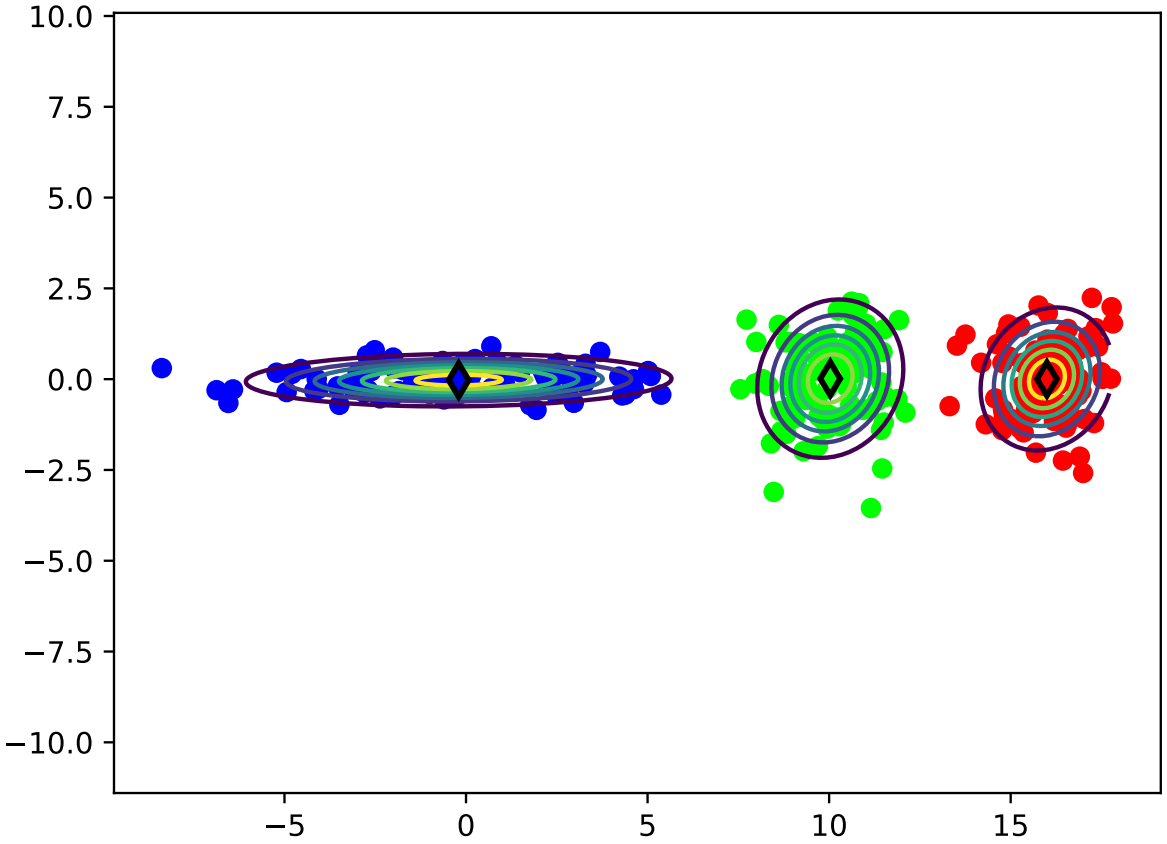
Interpolating between k-means and GMM E-M (part 2)

We can interpolate algorithmically,
meaning we can create algorithms that have elements of both.
Here's something like k-means but with weights and covariances.



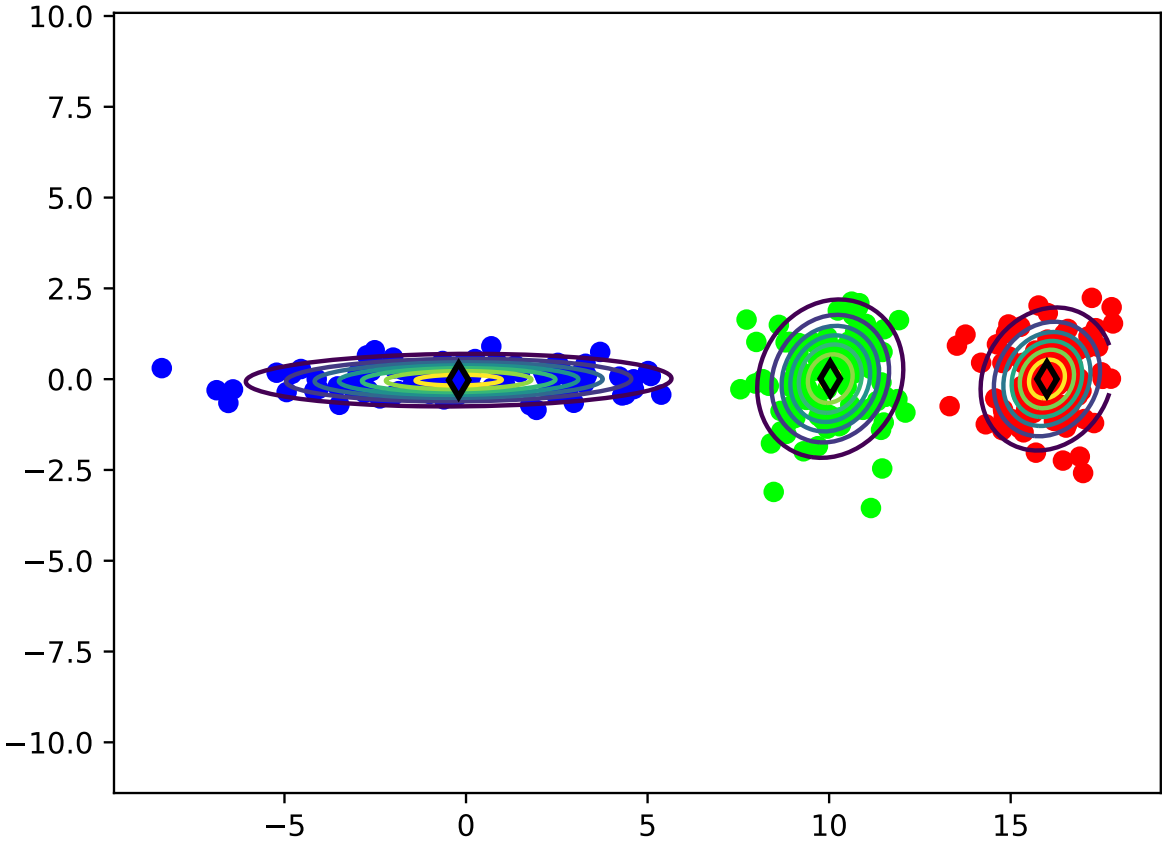
Interpolating between k-means and GMM E-M (part 2)

We can interpolate algorithmically,
meaning we can create algorithms that have elements of both.
Here's something like k-means but with weights and covariances.



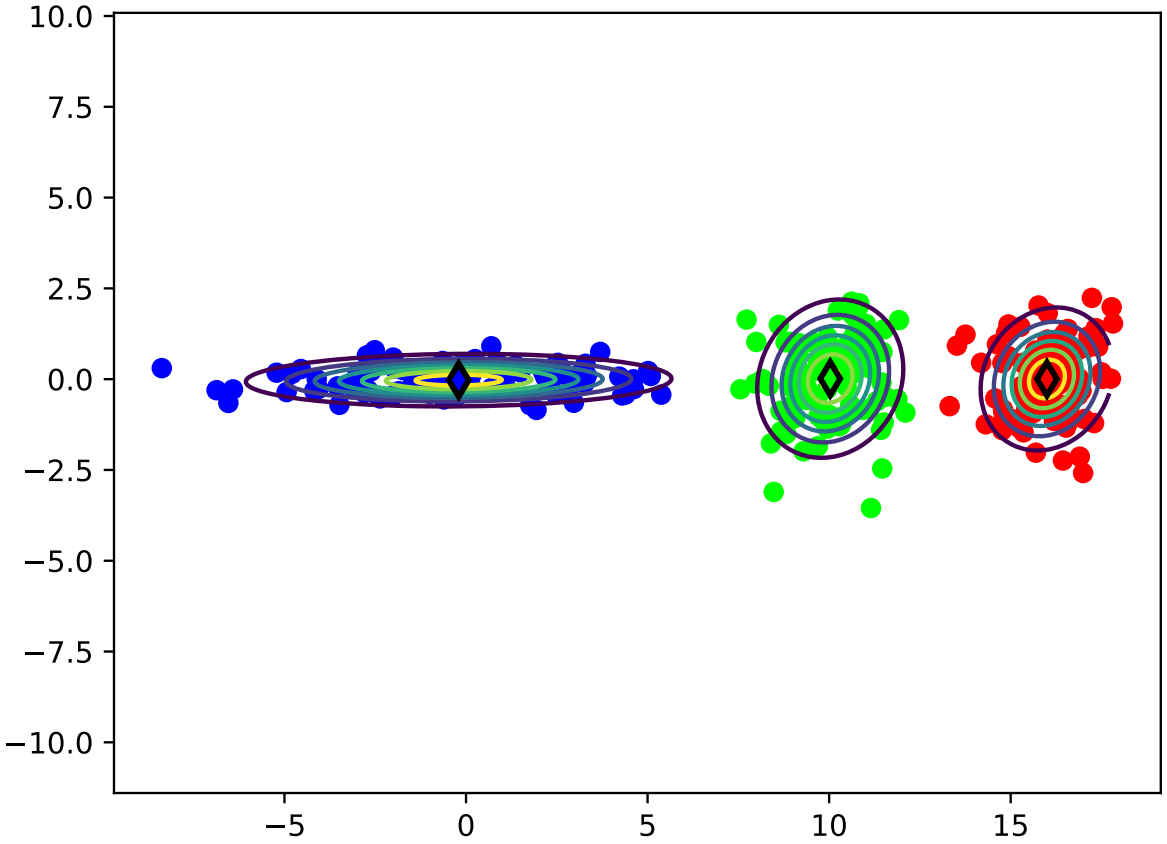
Interpolating between k-means and GMM E-M (part 2)

We can interpolate algorithmically,
meaning we can create algorithms that have elements of both.
Here's something like k-means but with weights and covariances.



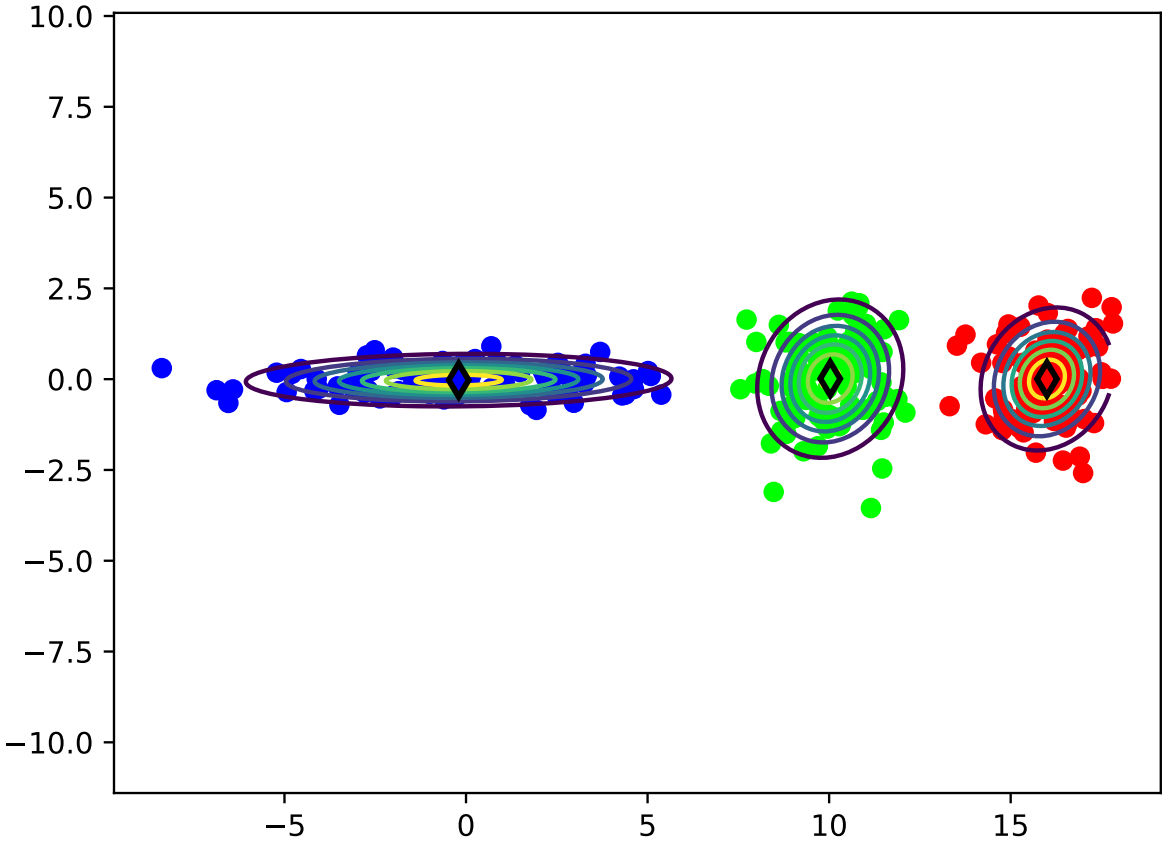
Interpolating between k-means and GMM E-M (part 2)

We can interpolate algorithmically,
meaning we can create algorithms that have elements of both.
Here's something like k-means but with weights and covariances.



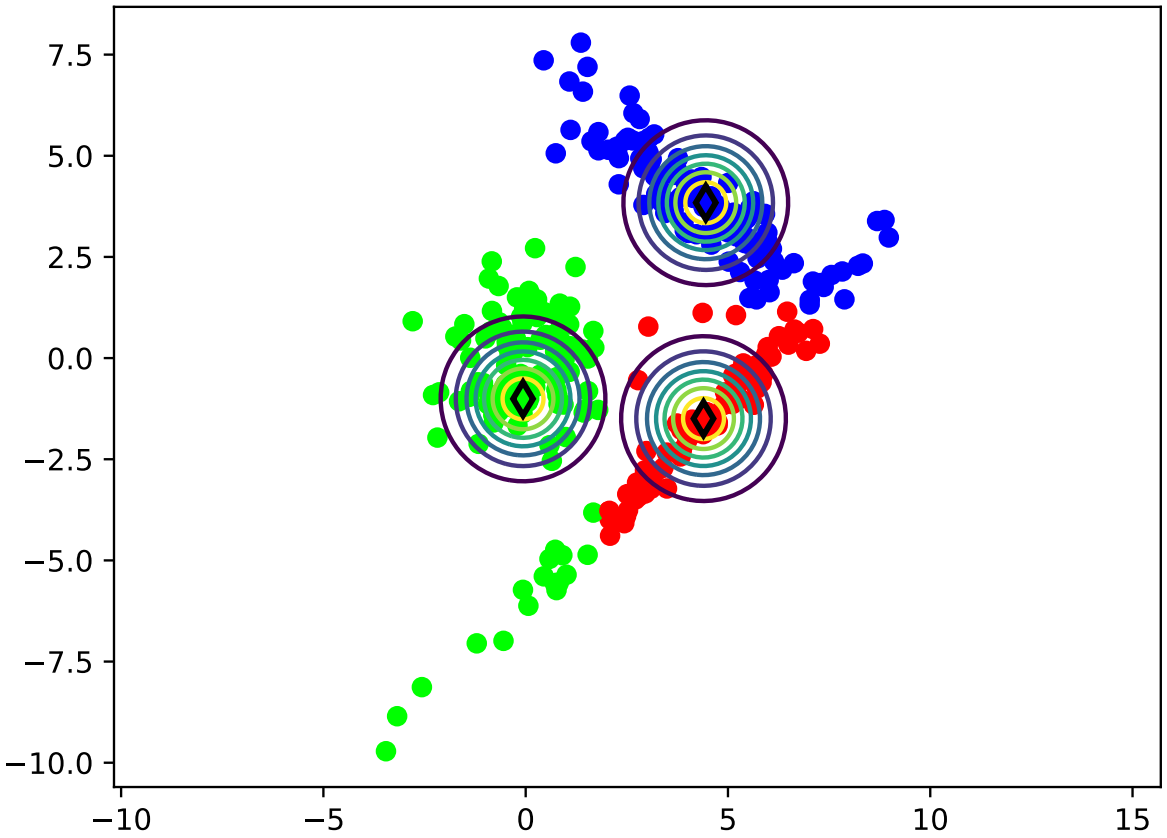
Interpolating between k-means and GMM E-M (part 2)

We can interpolate algorithmically,
meaning we can create algorithms that have elements of both.
Here's something like k-means but with weights and covariances.



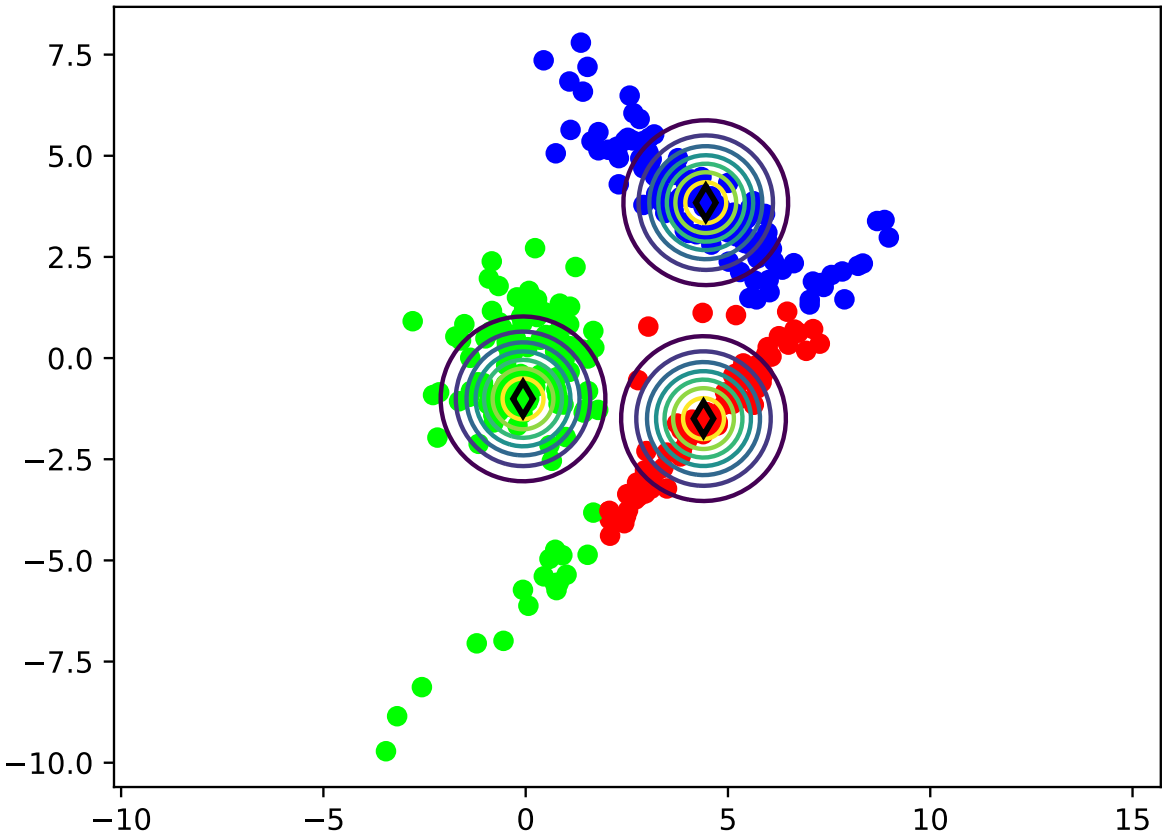
Interpolating between k-means and GMM E-M (part 2)

We can interpolate algorithmically,
meaning we can create algorithms that have elements of both.
Here's something like k-means but with weights and covariances.



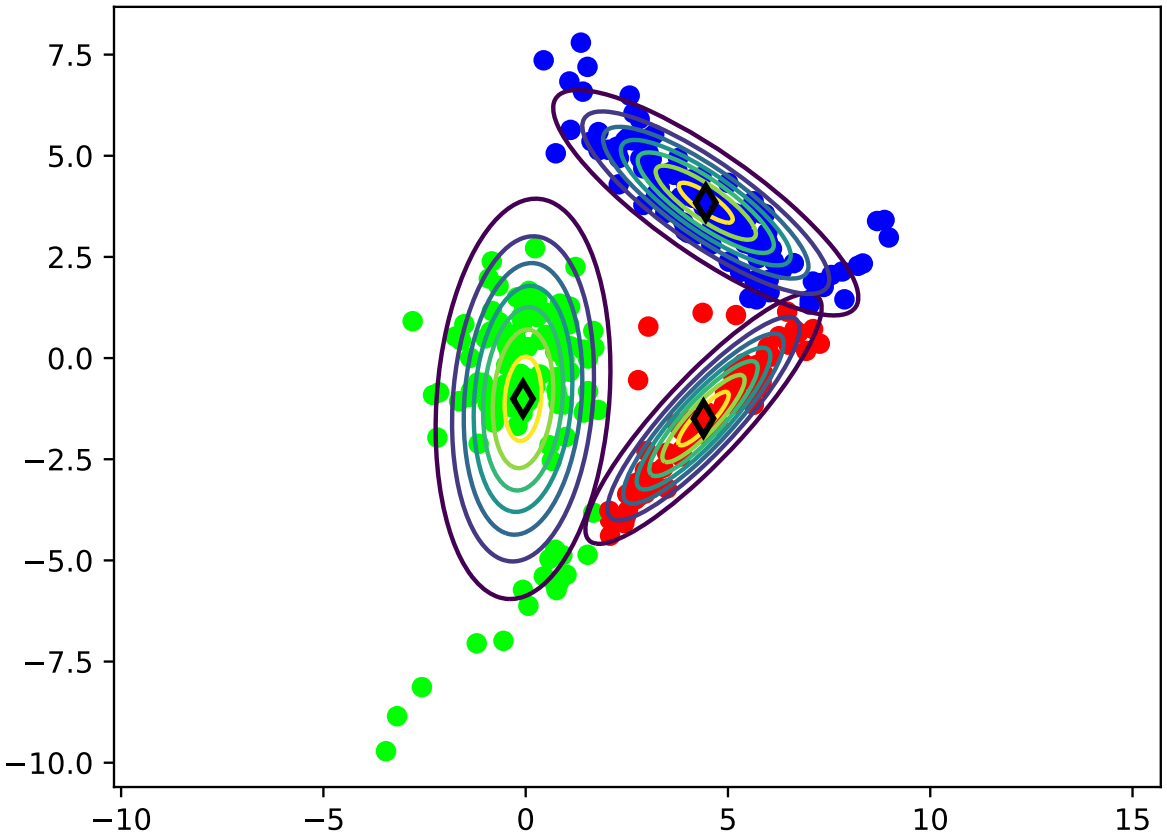
Interpolating between k-means and GMM E-M (part 2)

We can interpolate algorithmically,
meaning we can create algorithms that have elements of both.
Here's something like k-means but with weights and covariances.



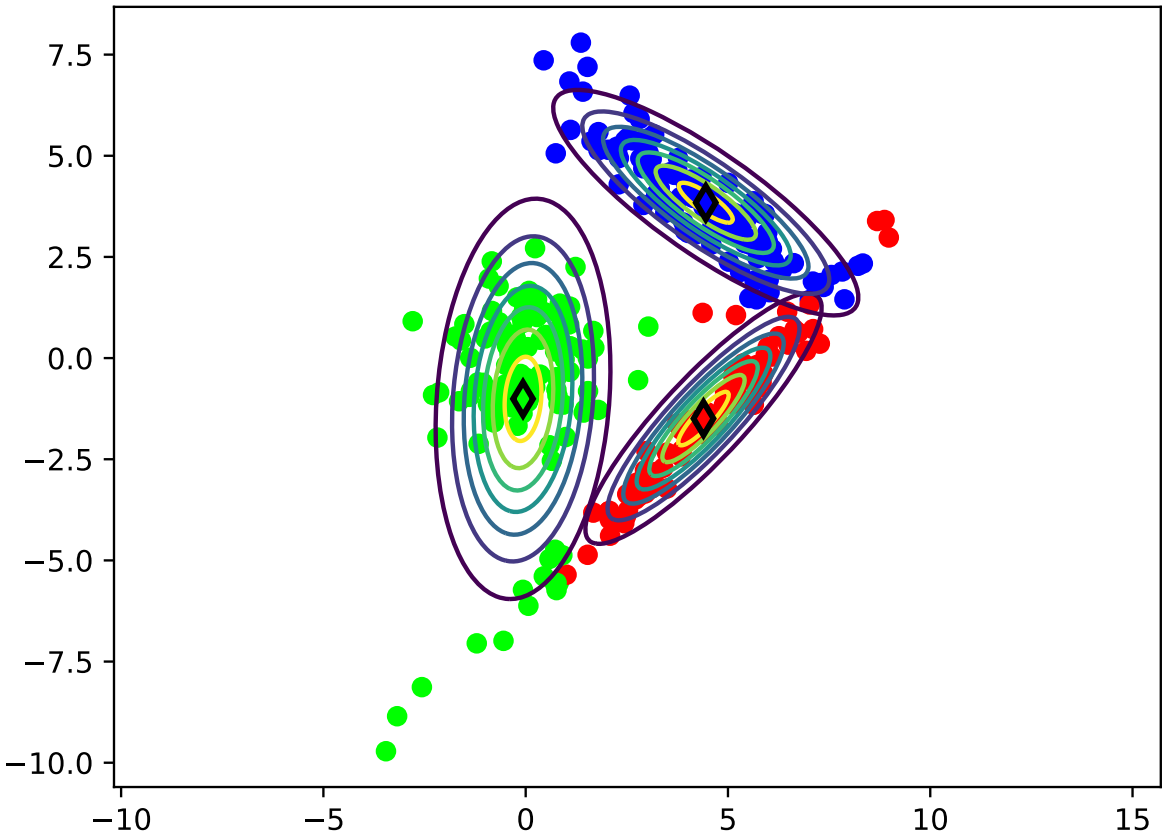
Interpolating between k-means and GMM E-M (part 2)

We can interpolate algorithmically,
meaning we can create algorithms that have elements of both.
Here's something like k-means but with weights and covariances.



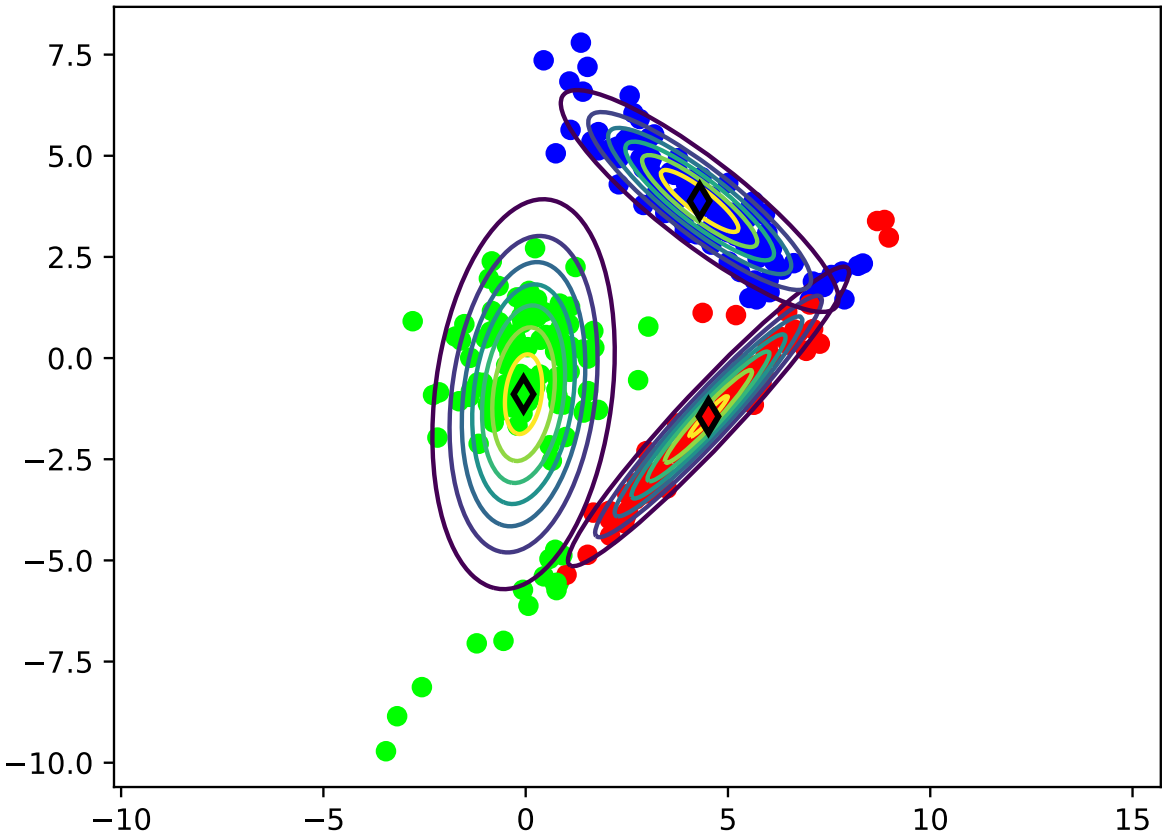
Interpolating between k-means and GMM E-M (part 2)

We can interpolate algorithmically,
meaning we can create algorithms that have elements of both.
Here's something like k-means but with weights and covariances.



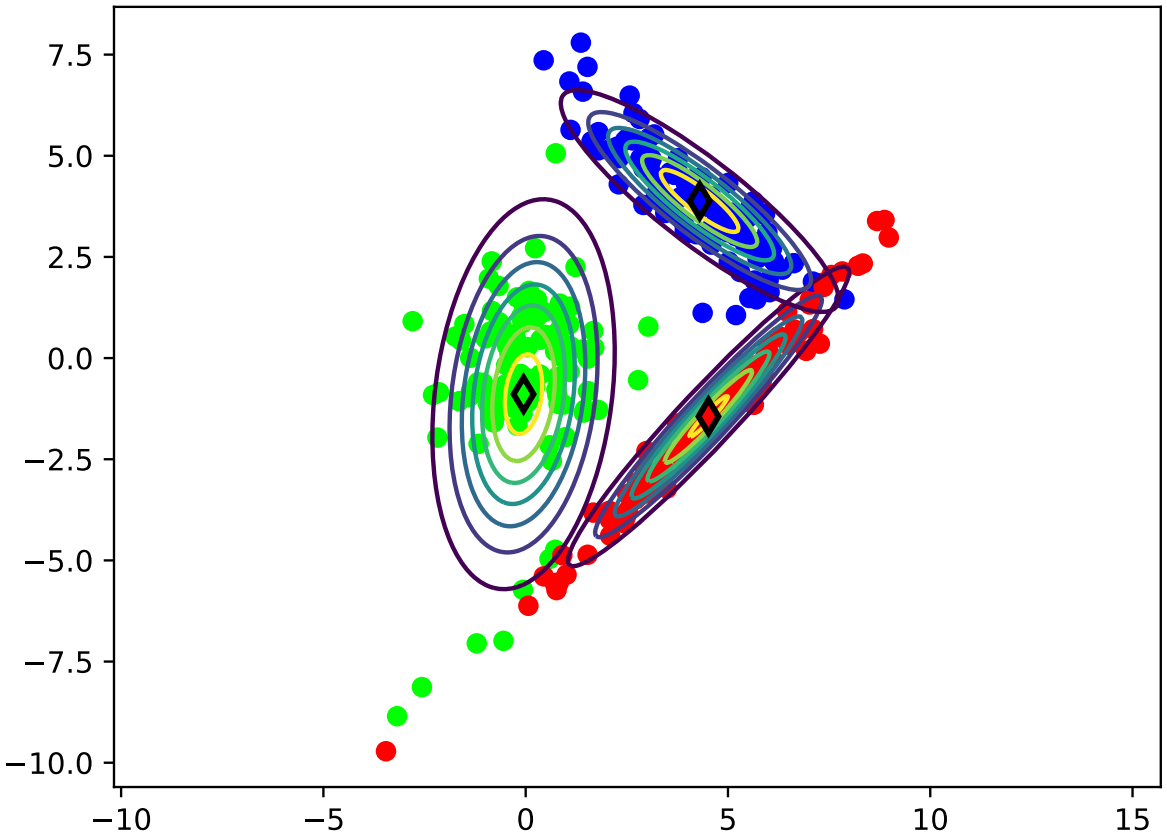
Interpolating between k-means and GMM E-M (part 2)

We can interpolate algorithmically,
meaning we can create algorithms that have elements of both.
Here's something like k-means but with weights and covariances.



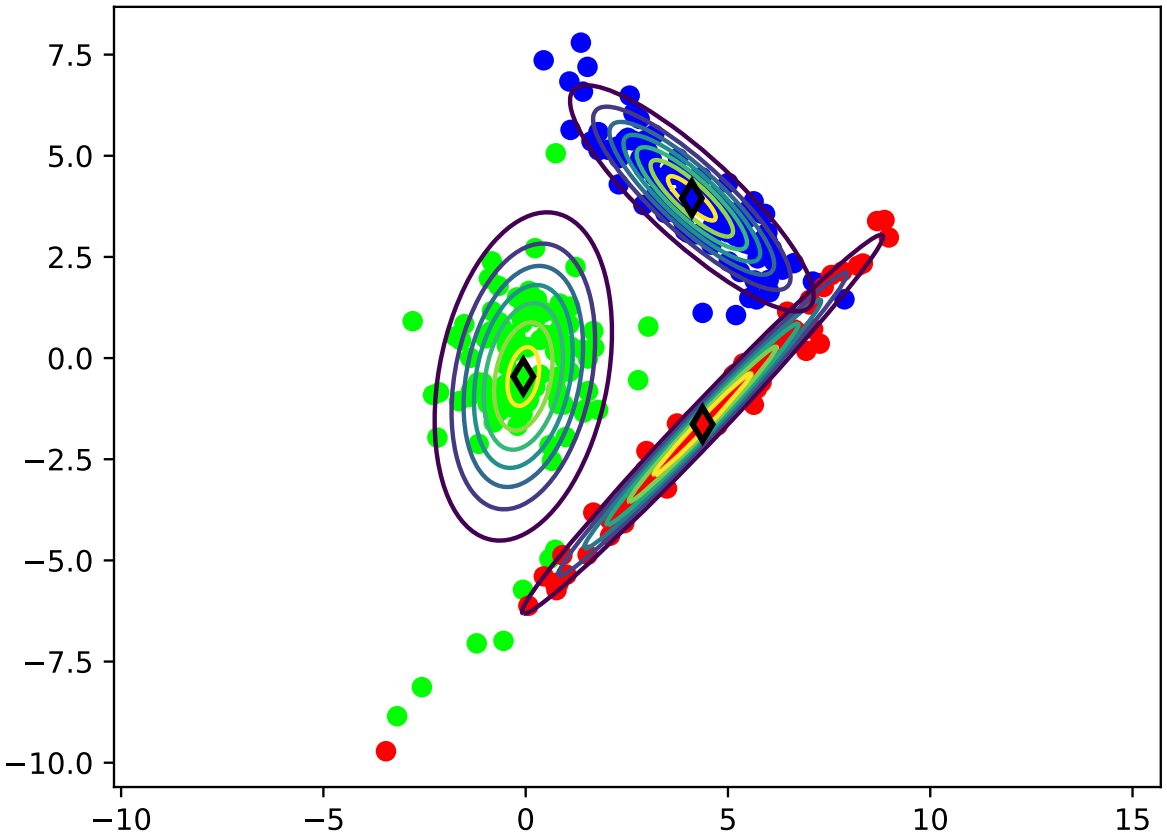
Interpolating between k-means and GMM E-M (part 2)

We can interpolate algorithmically,
meaning we can create algorithms that have elements of both.
Here's something like k-means but with weights and covariances.



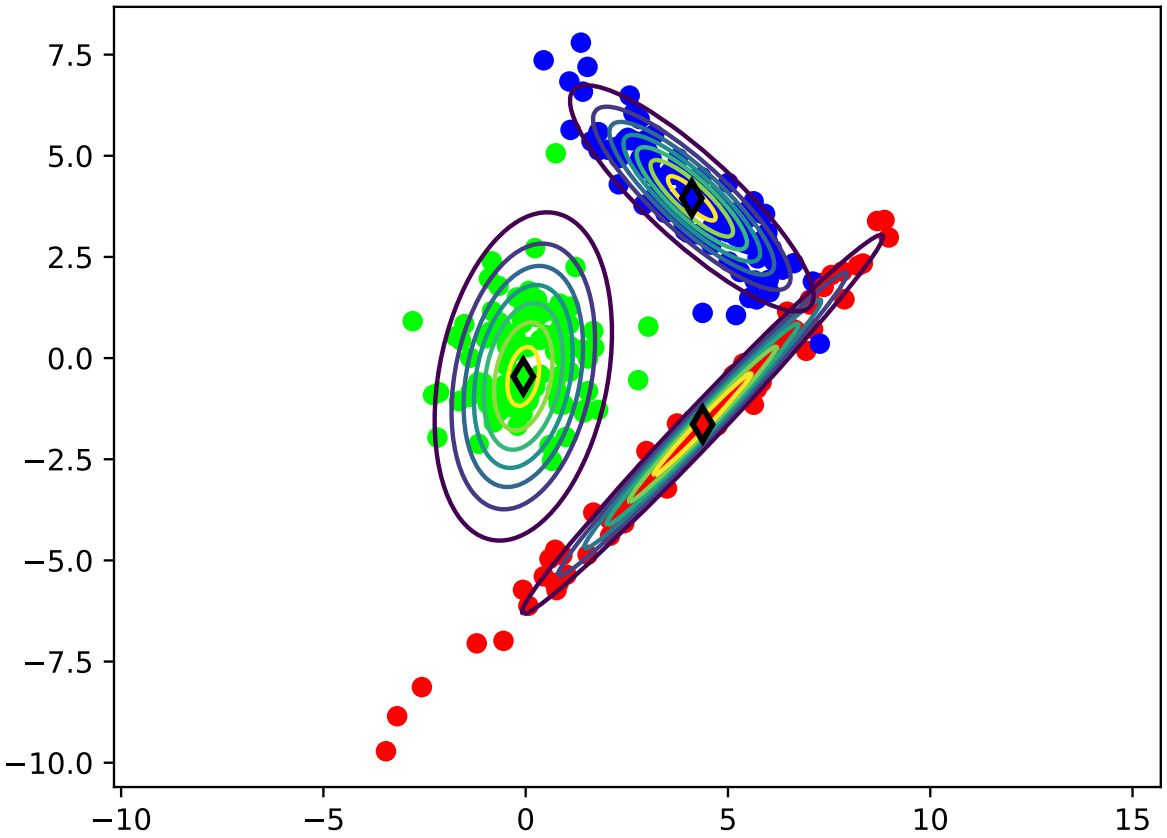
Interpolating between k-means and GMM E-M (part 2)

We can interpolate algorithmically,
meaning we can create algorithms that have elements of both.
Here's something like k-means but with weights and covariances.



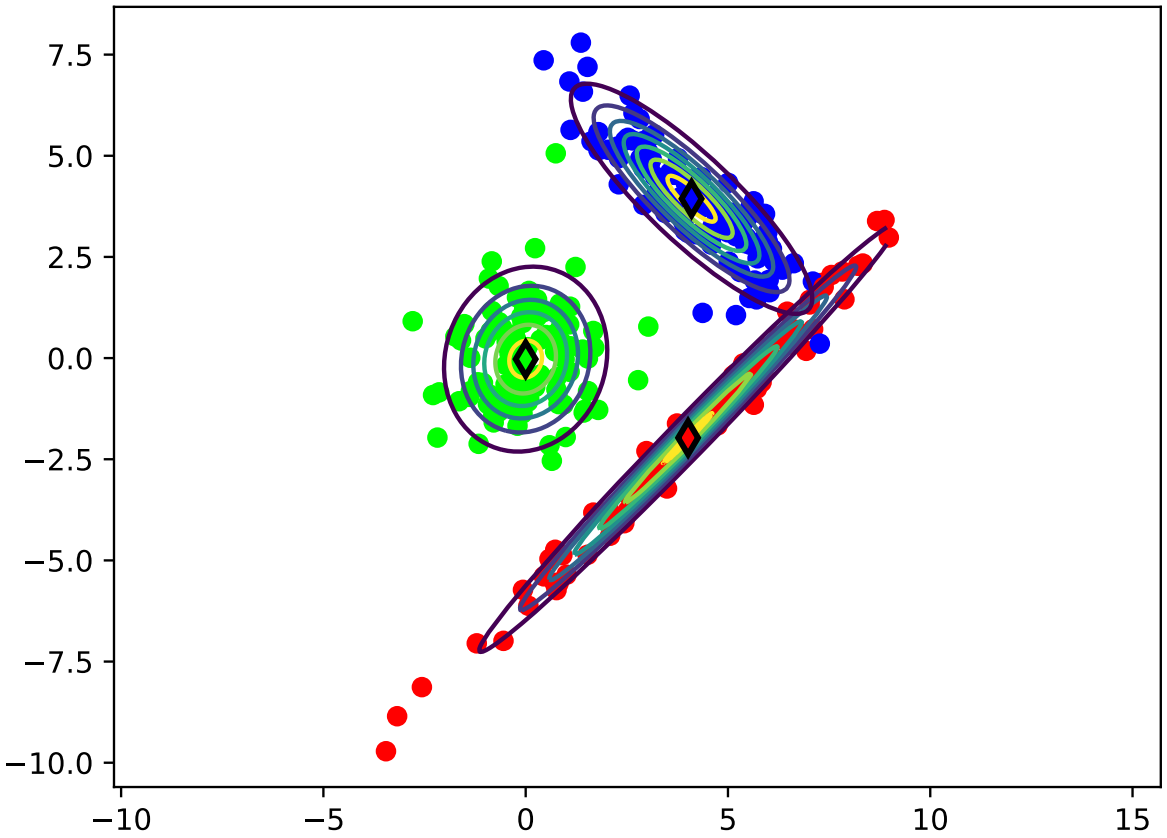
Interpolating between k-means and GMM E-M (part 2)

We can interpolate algorithmically,
meaning we can create algorithms that have elements of both.
Here's something like k-means but with weights and covariances.



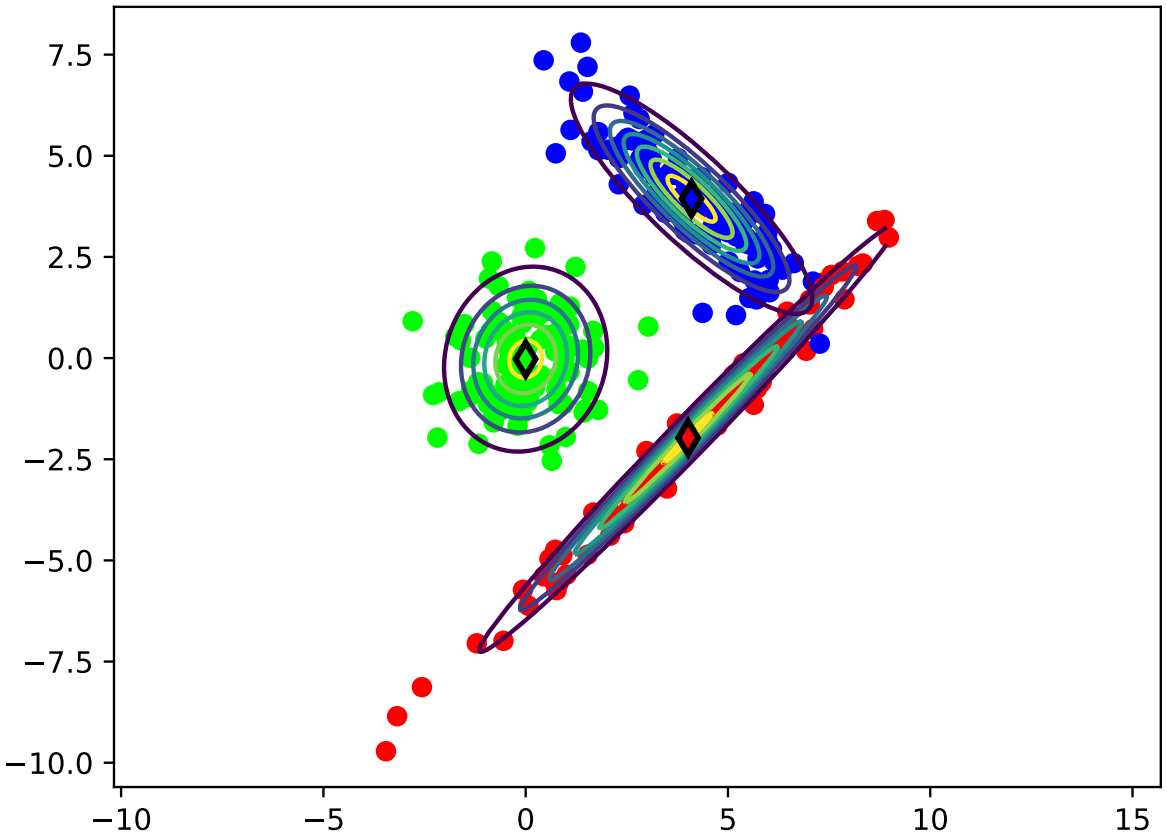
Interpolating between k-means and GMM E-M (part 2)

We can interpolate algorithmically,
meaning we can create algorithms that have elements of both.
Here's something like k-means but with weights and covariances.



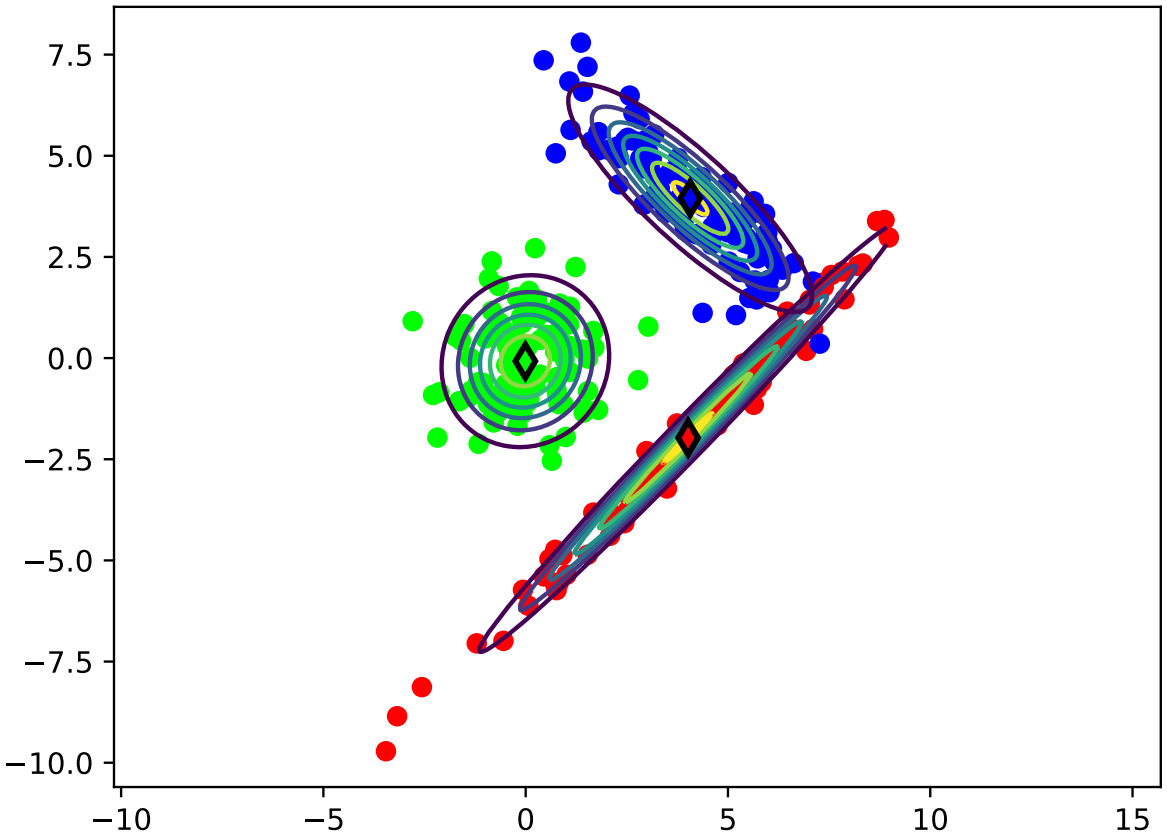
Interpolating between k-means and GMM E-M (part 2)

We can interpolate algorithmically,
meaning we can create algorithms that have elements of both.
Here's something like k-means but with weights and covariances.



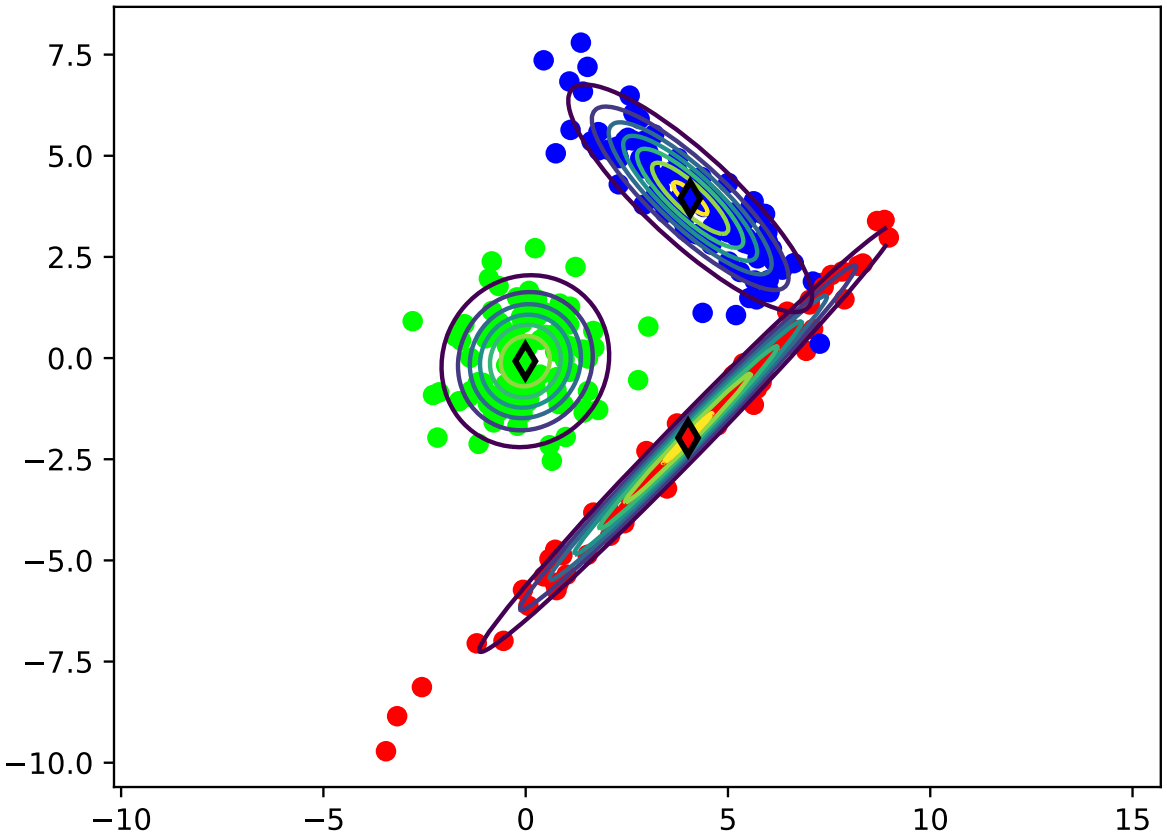
Interpolating between k-means and GMM E-M (part 2)

We can interpolate algorithmically,
meaning we can create algorithms that have elements of both.
Here's something like k-means but with weights and covariances.



Interpolating between k-means and GMM E-M (part 2)

We can interpolate algorithmically,
meaning we can create algorithms that have elements of both.
Here's something like k-means but with weights and covariances.



Summary of MLE part 2