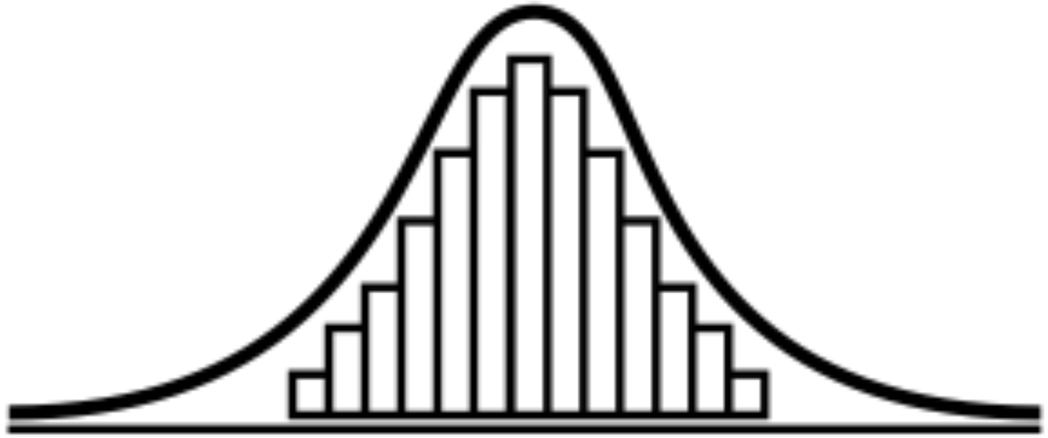
Gaussian Progress

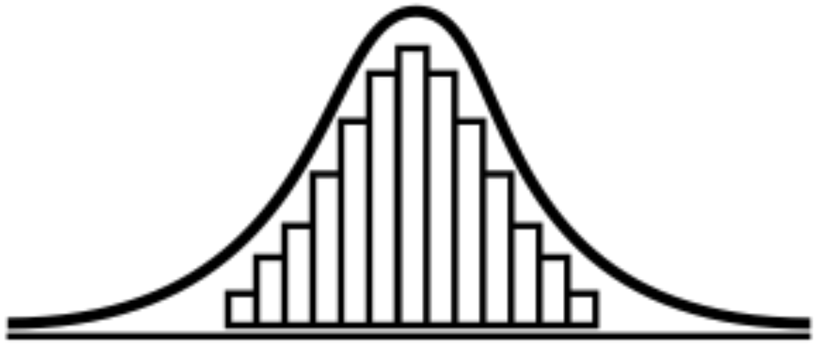
(an utmost normal topic)



a homage to gaussian methods, tricks and progress

When you're a machine learning professional you might feel like you need to learn so many algorithms that it can be hard to keep up. It can be very demotivating.

This talk is not about downplaying this feeling but it is about demonstrating a lovely hack; understanding a mother algorithm.



It turns out that if you appreciate what the gaussian distribution can do then there are lot's of algorithms that are much easier to grasp. This talk is an attempt at explaining the power of the Gaussian[tm] by stepping up the ladder of complexity of algorithms.

Today

1. Literally Introduce Gauss
2. Explain *the* guassian trick for
   — Classification, Clusters and Outliers
3. Explain normal neural benefits
   — Mixed Density Networks/Mixture Encoders
4. Live Code a Gaussian Process
   — A really meta trick

The problem

Teacher wants to read a newspaper so he comes up with a mathematical challenge for the kids to keep 'em busy.

So the teacher said ...

"Add all the numbers 1 till 100."

... and then continued reading his newspaper.

This is the task:

```
1   +   2 + ... +  99 + 100 = ?
```

This is the task:

```
1   +   2 + ... +  99 + 100 = ?
```

But here's a rewrite.

```
1   +   2 + ... +  49 +  50
100 +  99 + ... +  52 +  51
--------------------------- +
101 + 101 + ... + 101 + 101 = 101 * 50 = 5050
```

This is the task:

```
1   +   2 + ... +  99 + 100 = ?
```

But this task is similar.

```
1   +   2 + ... +  99 + 100
100 +  99 + ... +   2 +   1
------------------------------ +
101 + 101 + ... + 101 + 101 -> 101 * 100/2 = 5050
```

This is the task:

```
1 + 2 + ... + (n-1) + n = ?
```

This is the task:

```
1 + 2 + ... + (n-1) + n = ?
```

This is what math is!

```
1       +     2   + ... + (n-1) +   n
n       + (n-1) + ... +     2   +   1
------------------------------------ +
(n+1) + (n+1) + ... + (n+1) + (n+1) -> (n+1) * n/2
```

Math is a compiler for numbers.

But math is best explained with stories.

The story I just told is more commonly told via this formula:

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$$

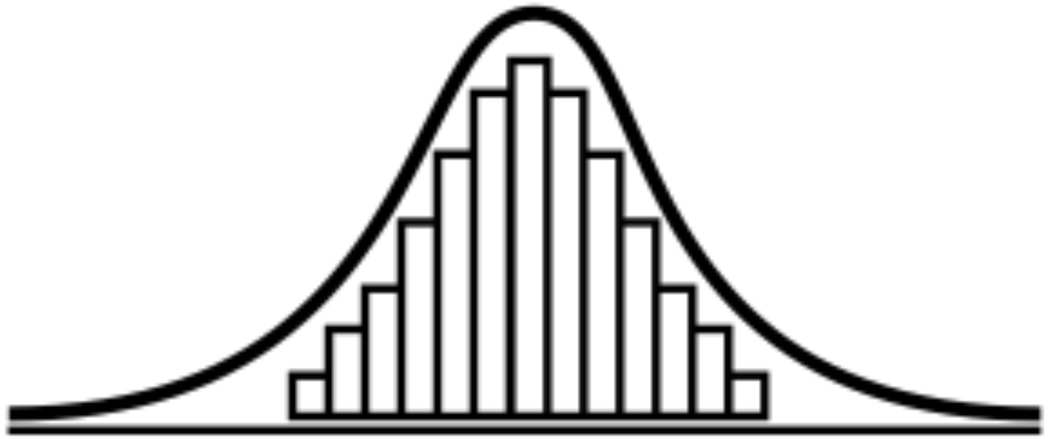
I personally think the story of the boy who bested his teacher is a better way of explaining things. Especially since this story is something that *actually* happened.

It supposedly actually happened to a kid named "Carl Friedrich Gauss".

And something very popular got named after him;

— the gaussian distribution
— the normal distribution
— the bell curve

These are all the same thing.

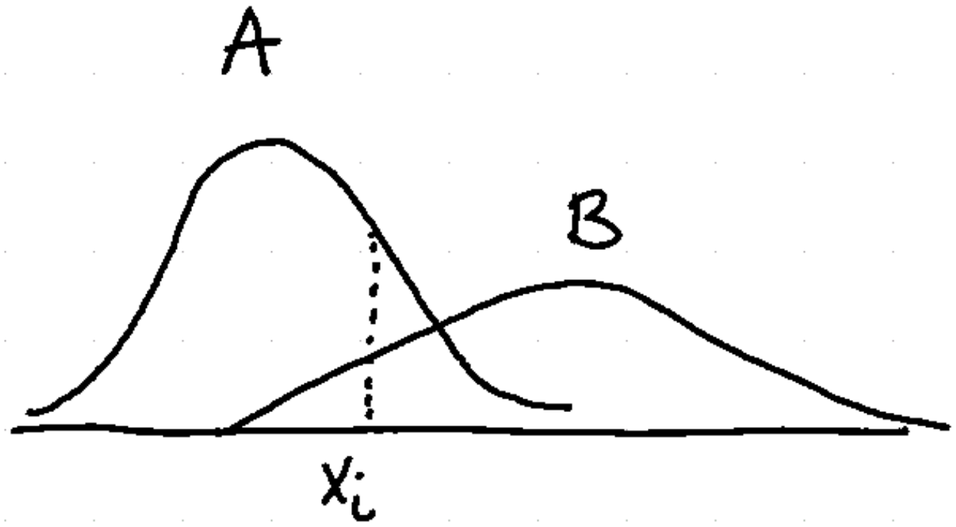


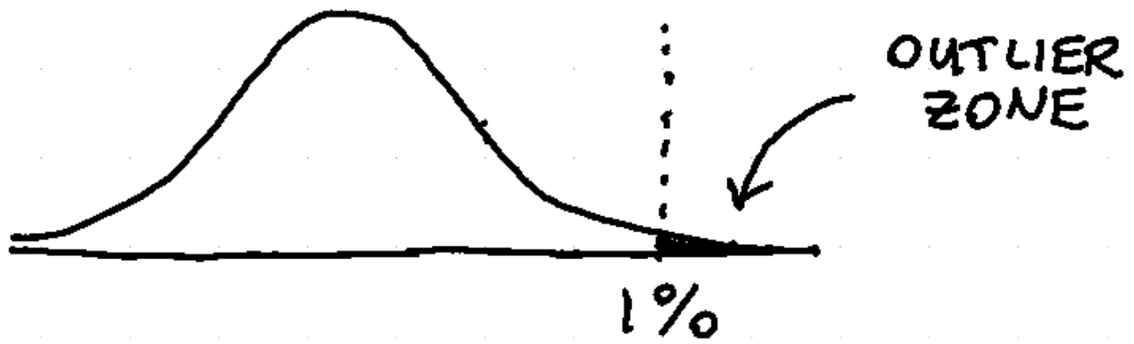
Gauss As A Building Block

This is the maths in one dimension.

$$f\left(x|\mu,\sigma^2\right) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

But here's some usecases:





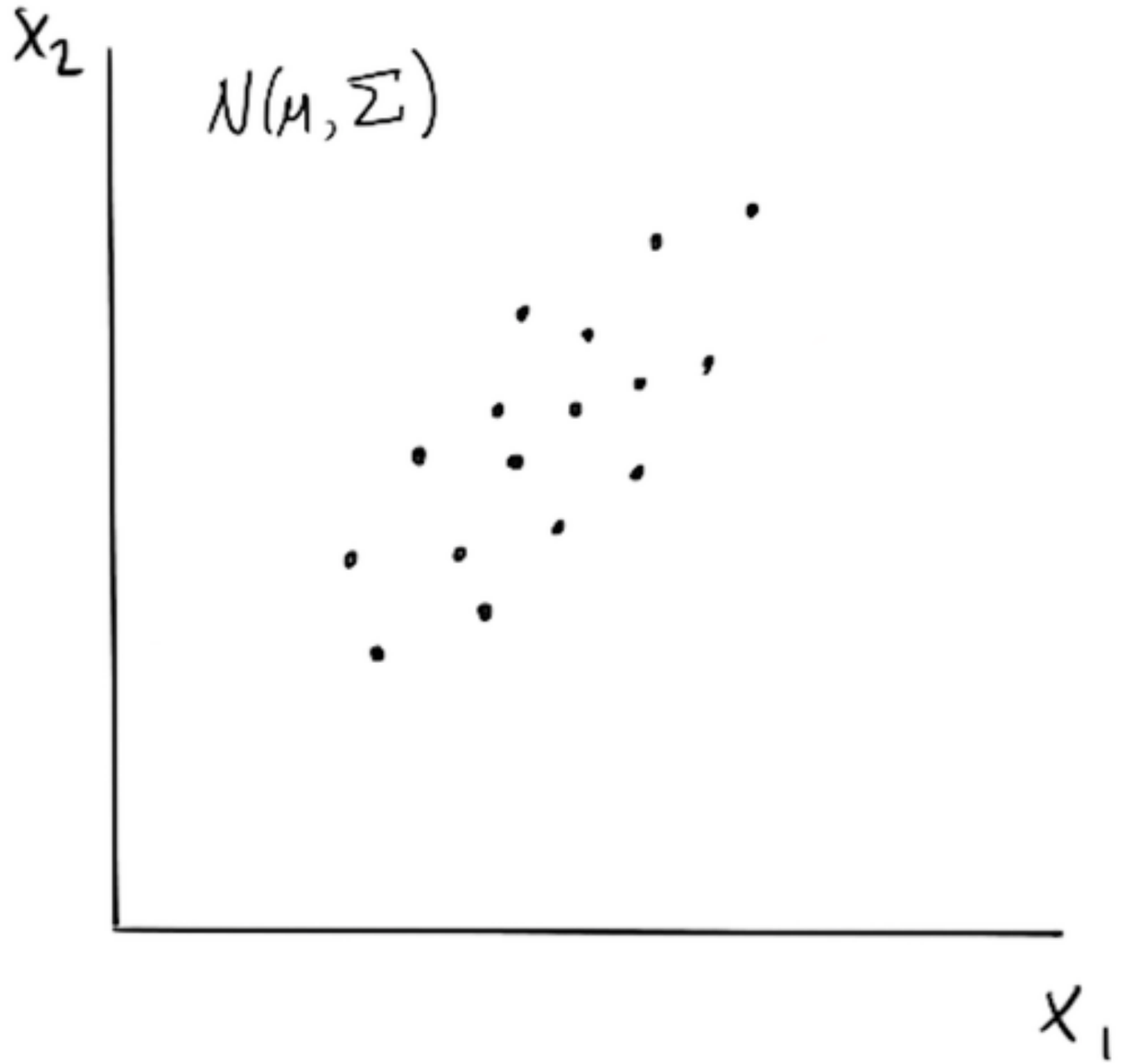
$$p(x_i = A) = \frac{f_A(x_i)}{f_A(x_i) + f_B(x_i)}$$

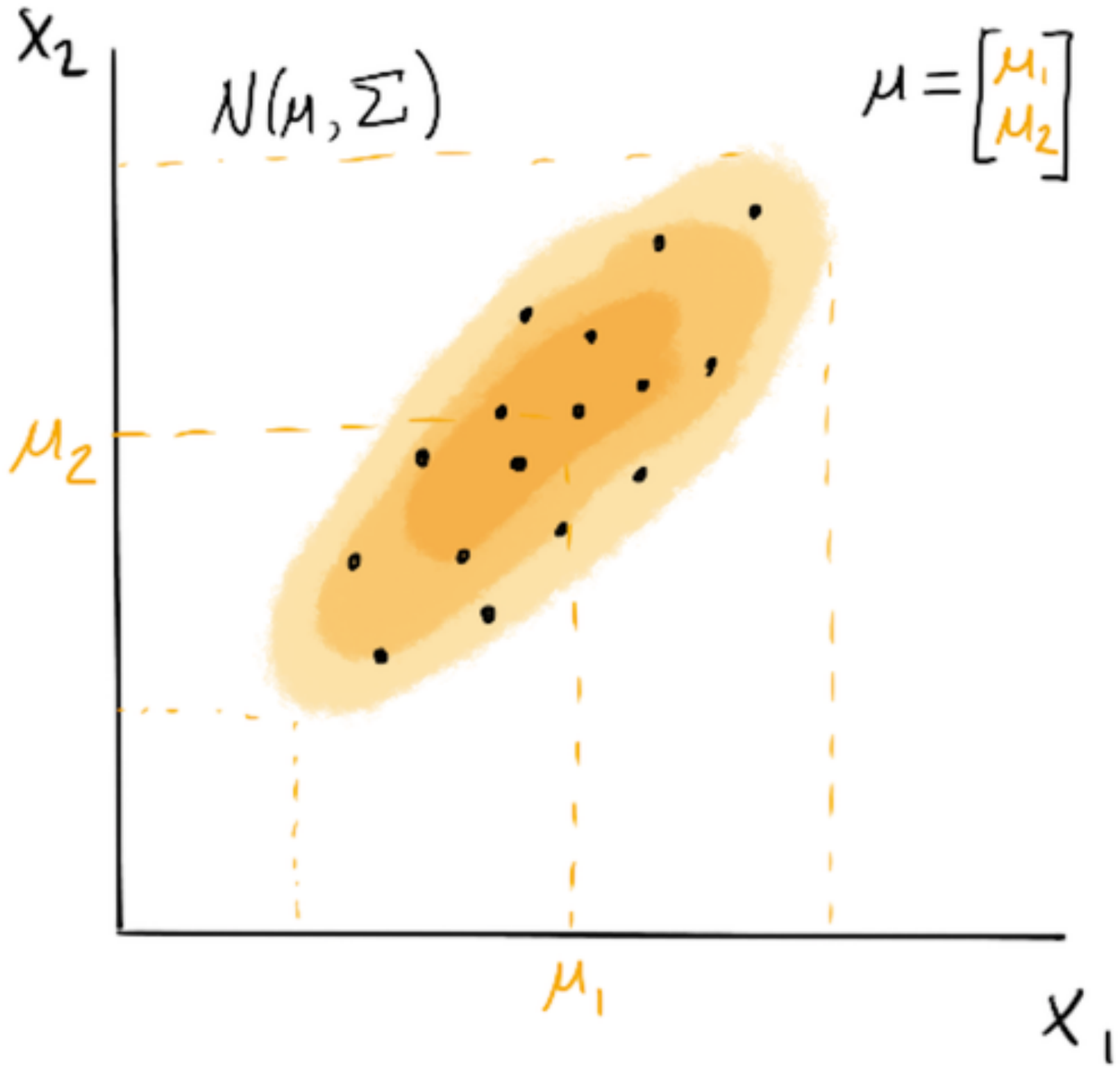
This is the maths in one dimension.

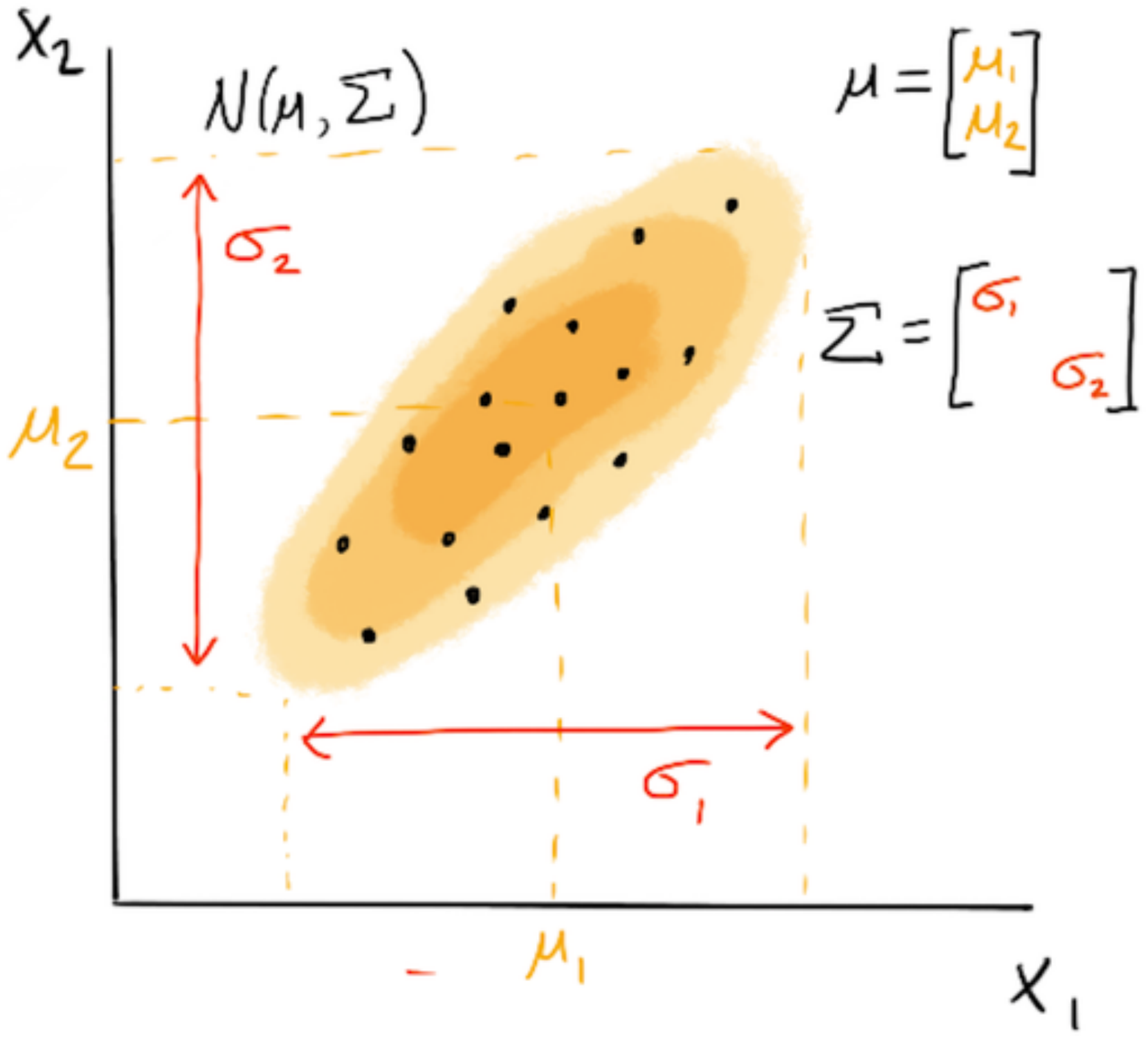
$$f\left(x|\mu,\sigma^2\right) = \frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

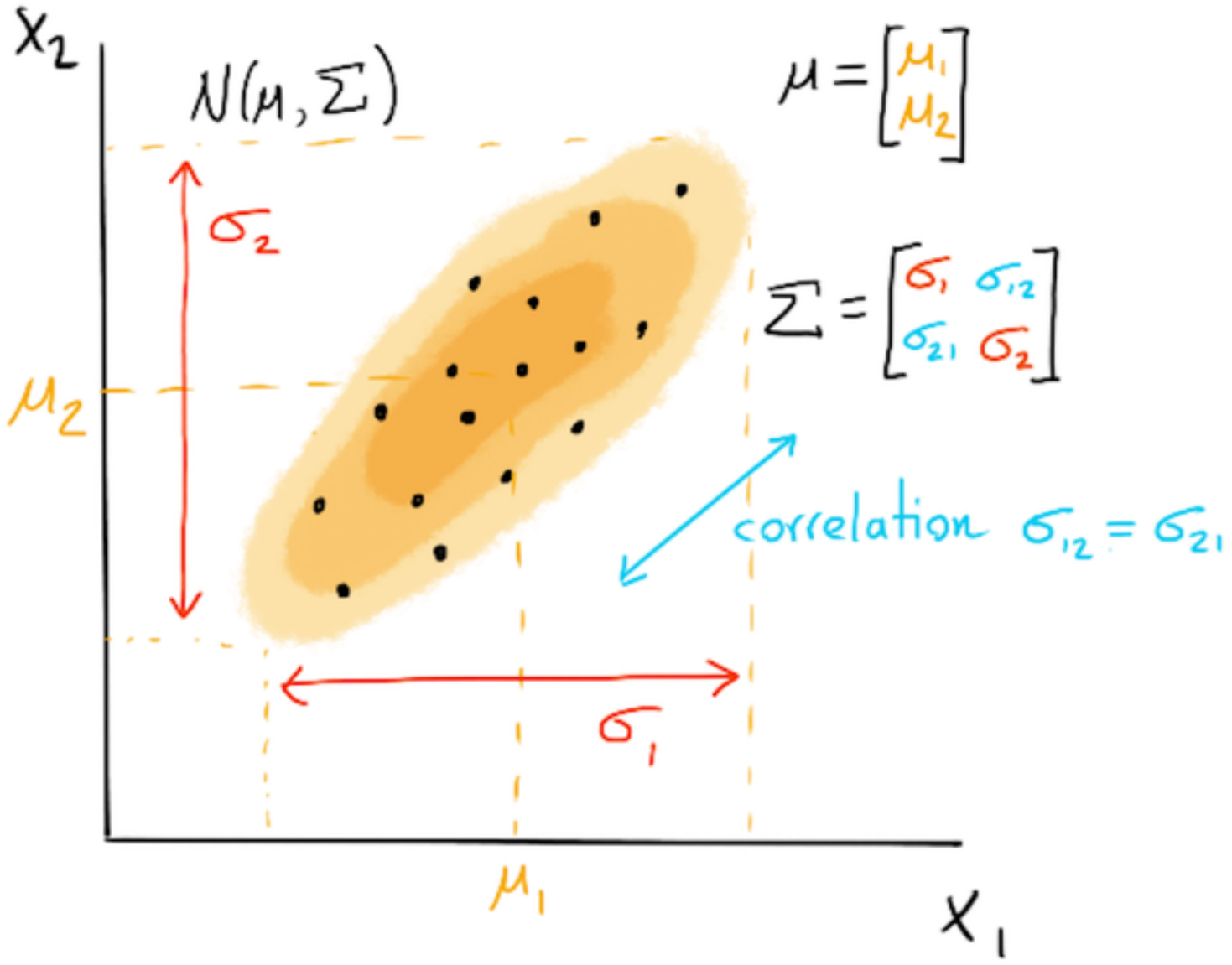
This is the formula for 2+ dimensions.

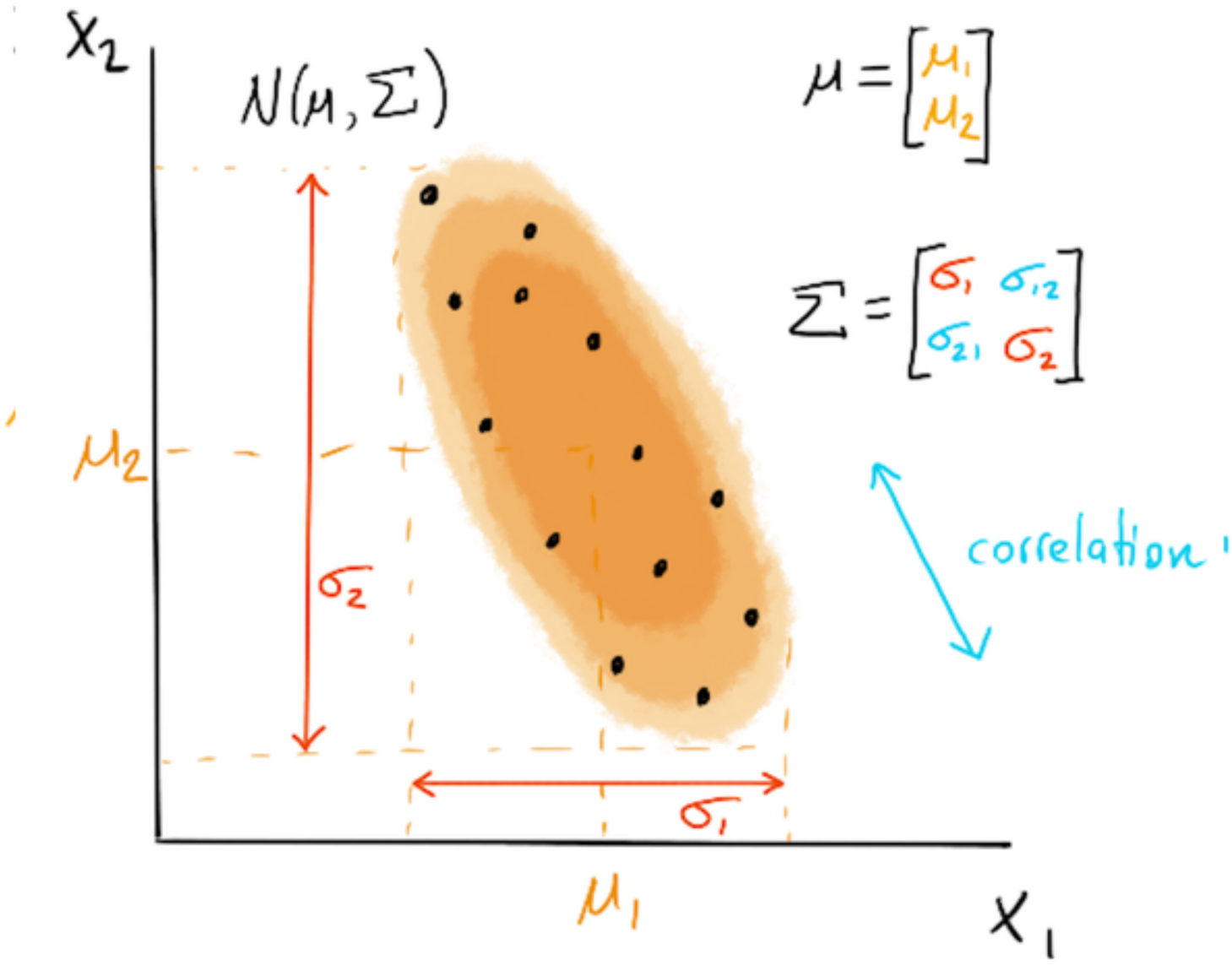
$$f_{\mathbf{x}}\left(x_1,\ldots,x_k\right) = \frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)}{\sqrt{(2\pi)^k|\boldsymbol{\Sigma}|}}$$

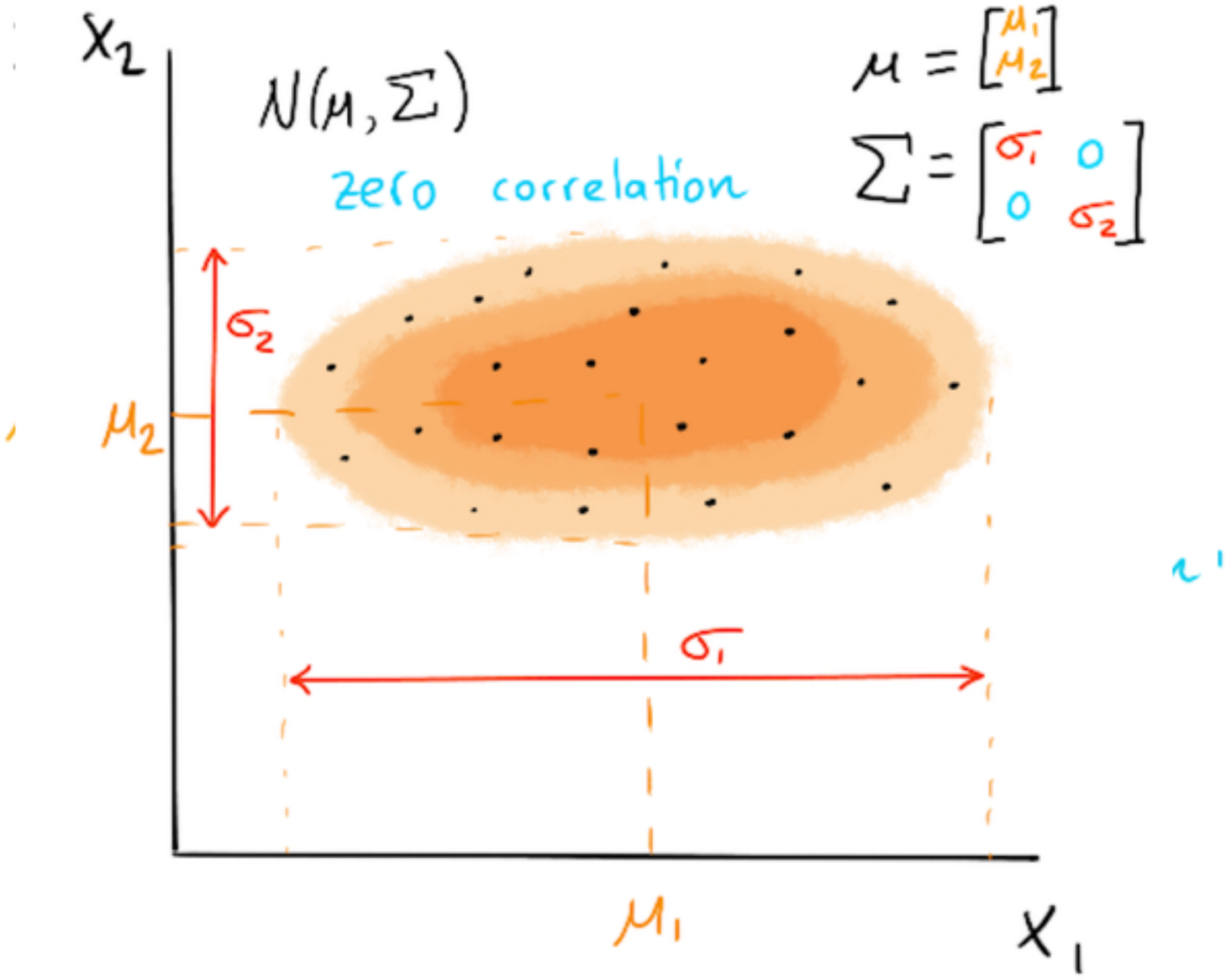

X_2
$N(\mu, \Sigma)$
X_1


X_2
$N(\mu, \Sigma)$
$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$
μ_2
μ_1
X_1


X_2
$N(\mu, \Sigma)$
$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$
σ_2
$\Sigma = \begin{bmatrix} \sigma_1 & \\ & \sigma_2 \end{bmatrix}$
μ_2
σ_1
μ_1
X_1

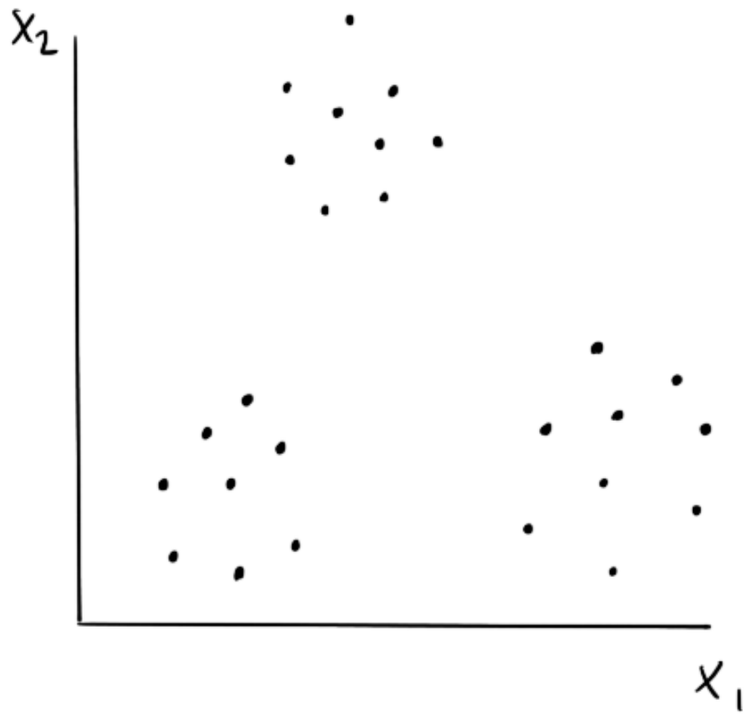

X_2
$N(\mu, \Sigma)$
$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$
σ_2
μ_2
$\Sigma = \begin{bmatrix} \sigma_1 & \sigma_{12} \\ \sigma_{21} & \sigma_2 \end{bmatrix}$
correlation $\sigma_{12} = \sigma_{21}$
σ_1
μ_1
X_1

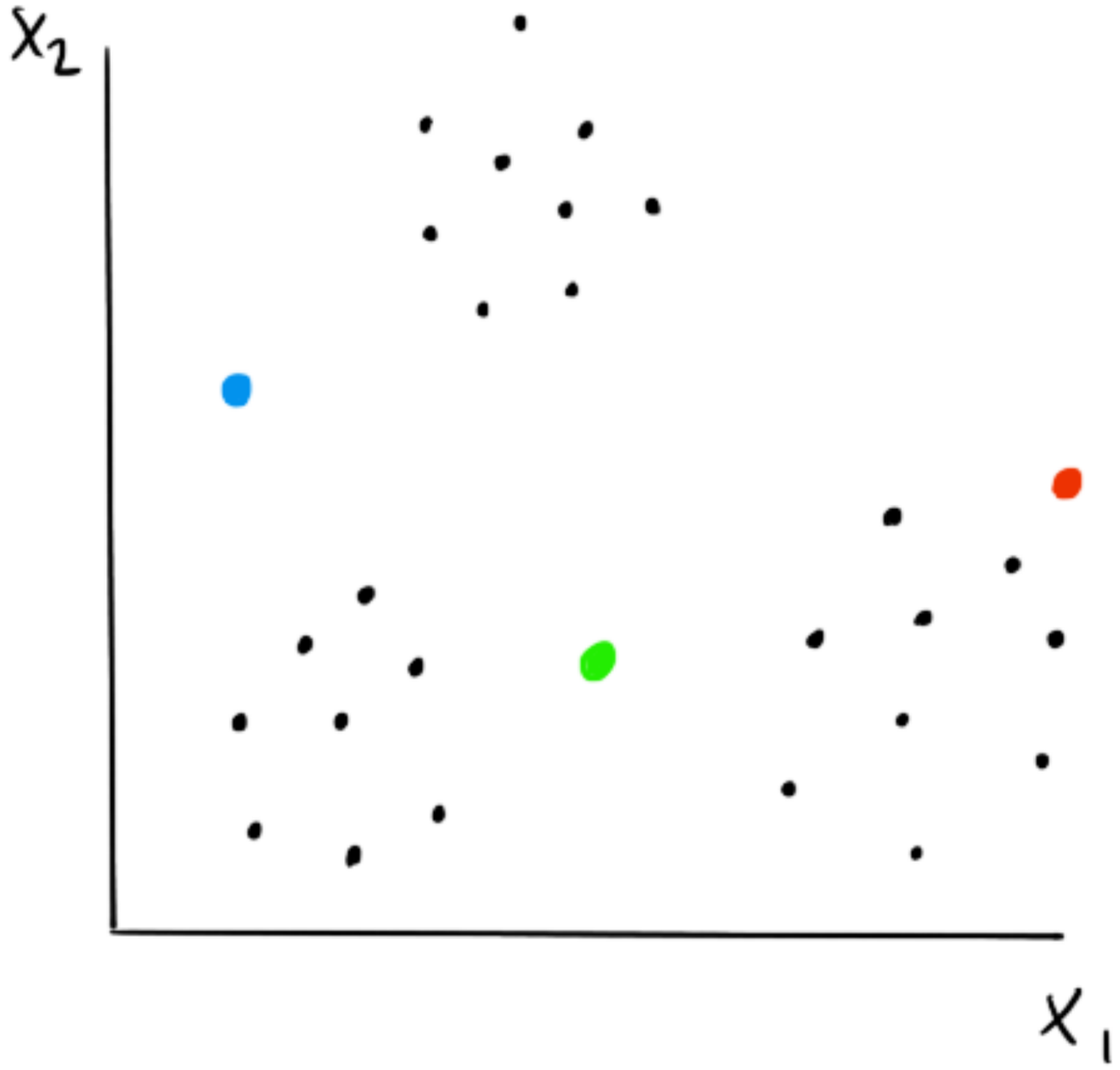

X_2
$N(\mu, \Sigma)$
$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$
$\Sigma = \begin{bmatrix} \sigma_1 & \sigma_{12} \\ \sigma_{21} & \sigma_2 \end{bmatrix}$
μ_2
σ_2
correlation!
σ_1
μ_1
X_1

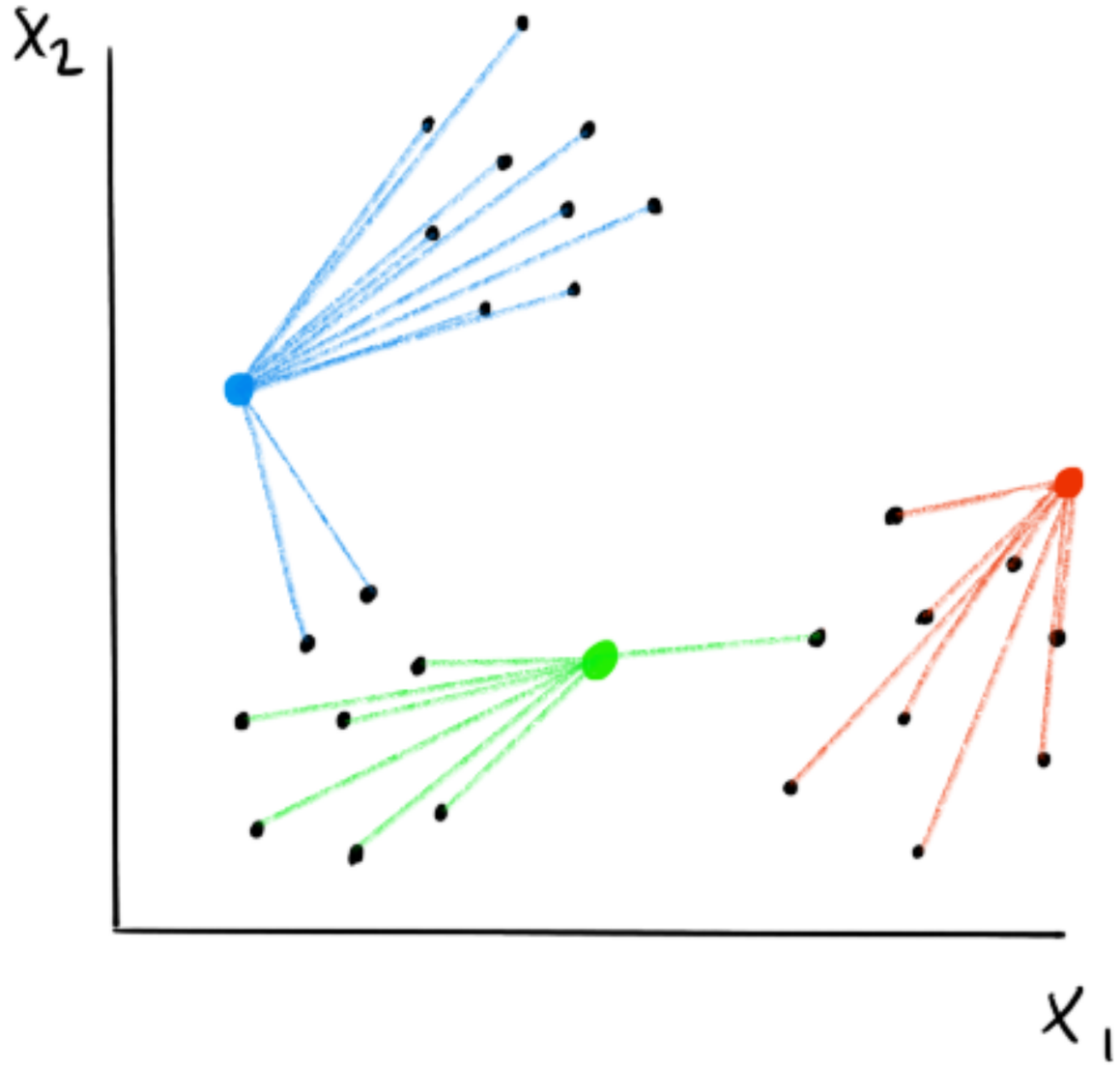

X_2
$N(\mu, \Sigma)$
zero correlation
$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$
$\Sigma = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix}$
σ_2
μ_2
σ_1
μ_1
X_1

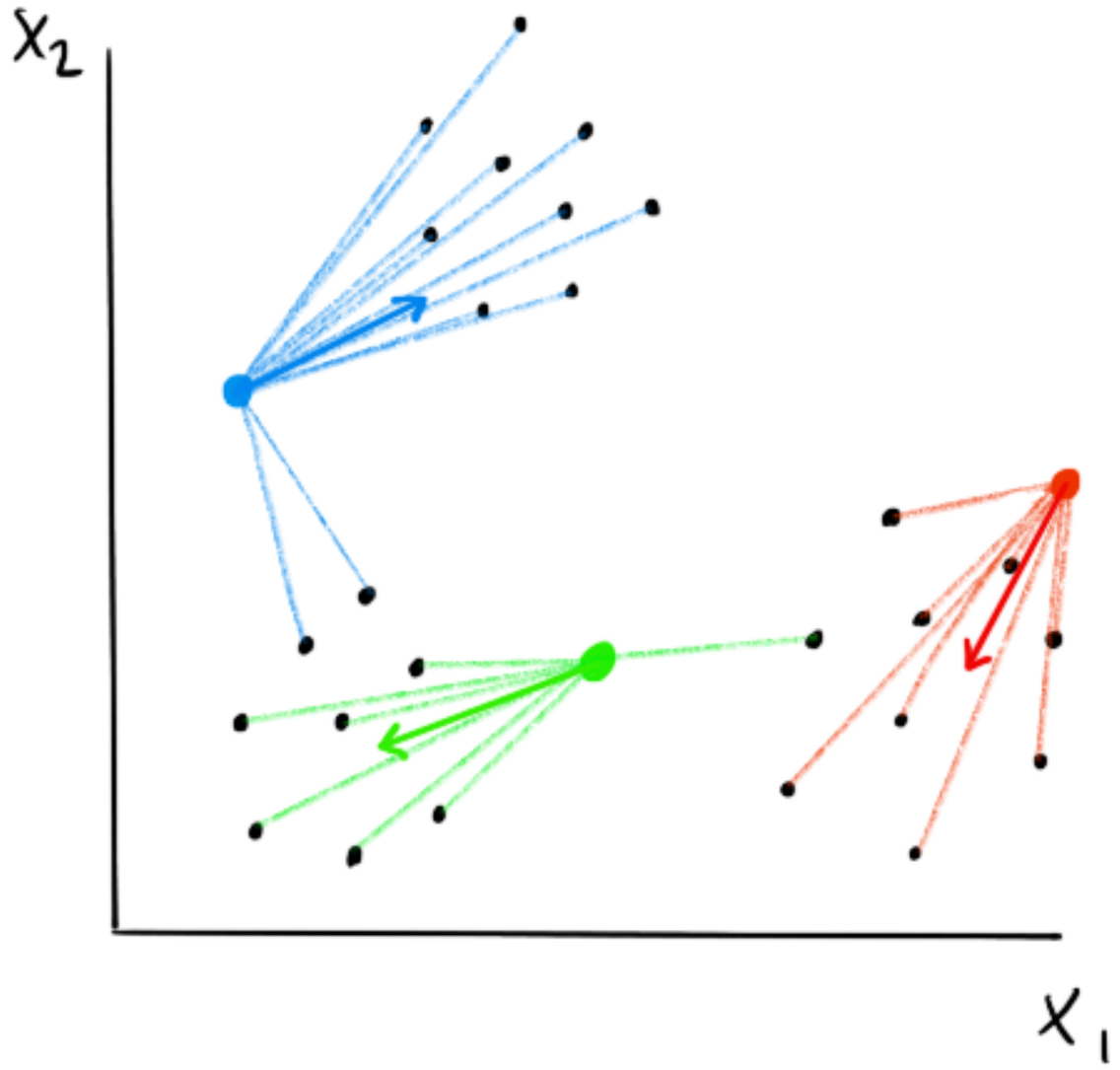
So what can you do with this? Plenty!

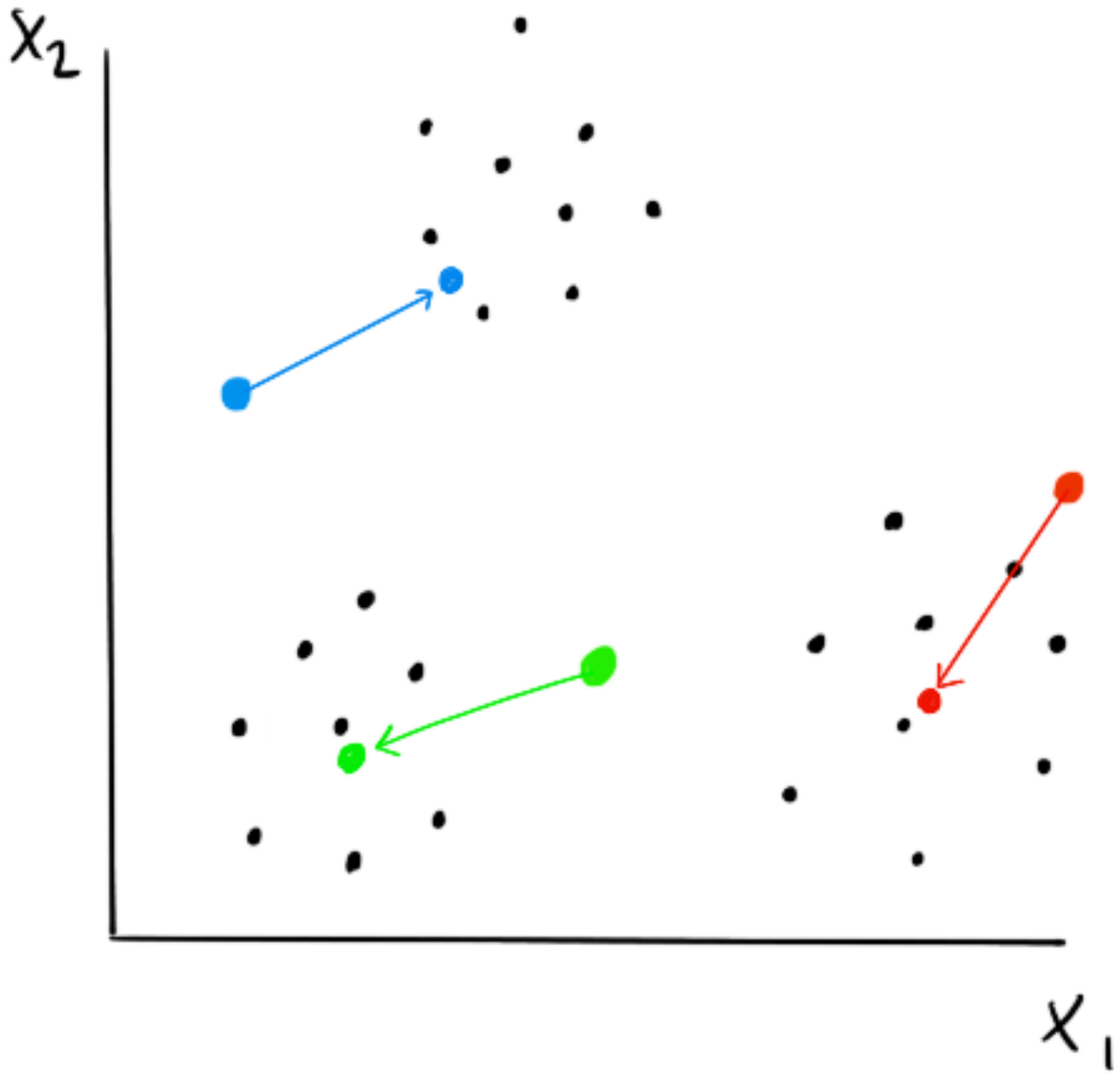
For starters; we can take k-means and make it better.

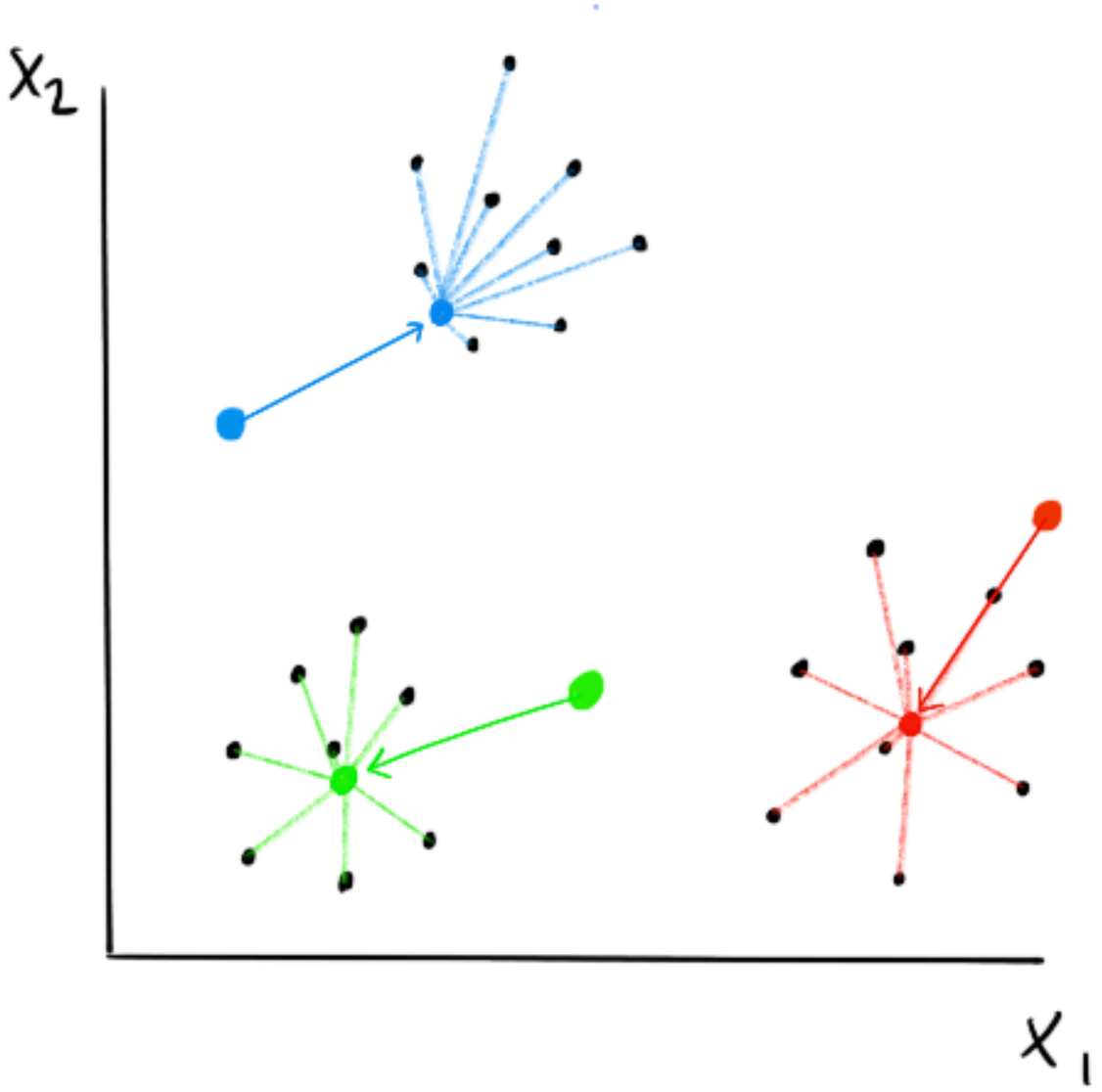


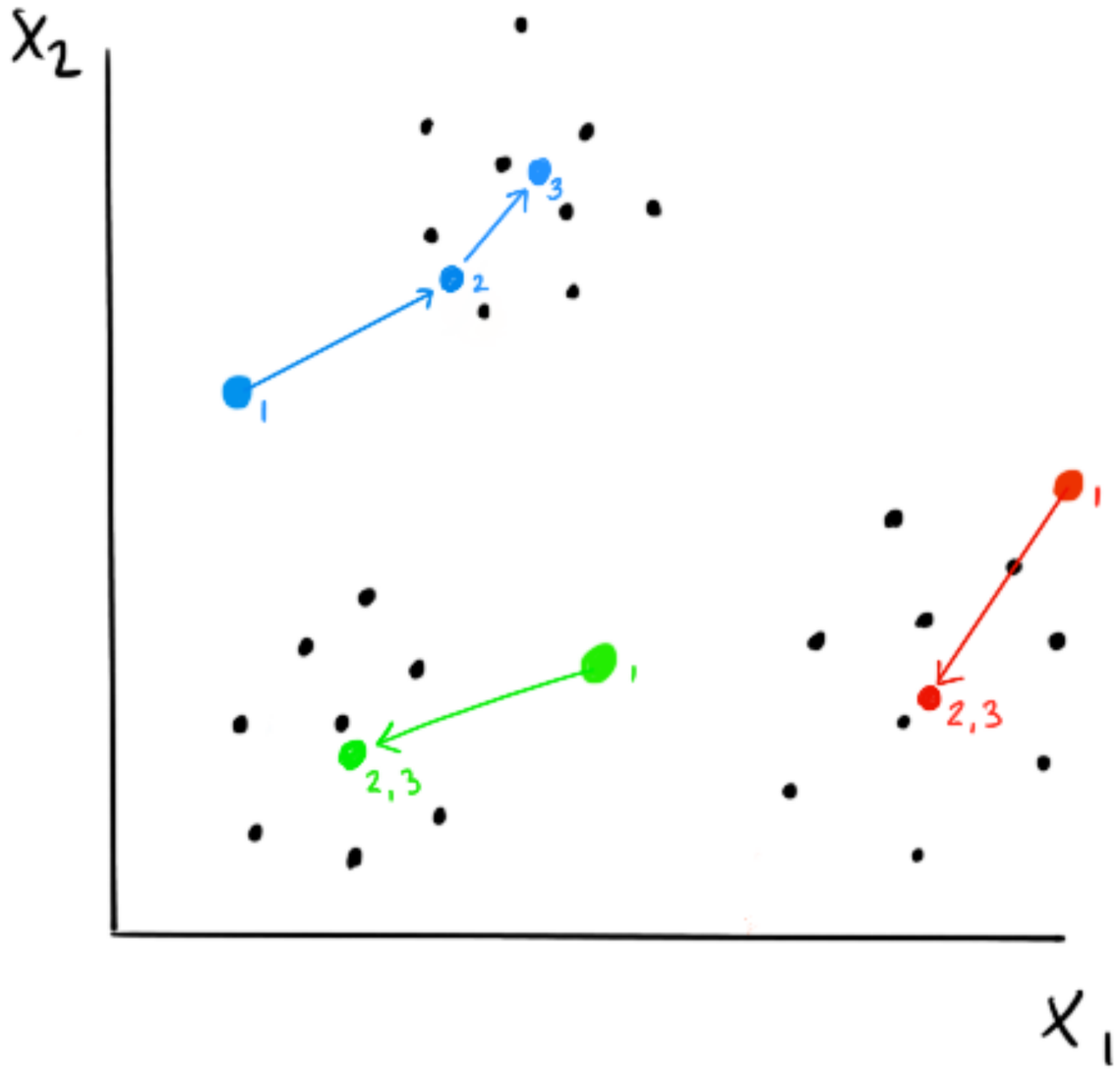

X_2
X_1


X_2
X_1


X_2
X_1


X2
X1


X_2
X_1

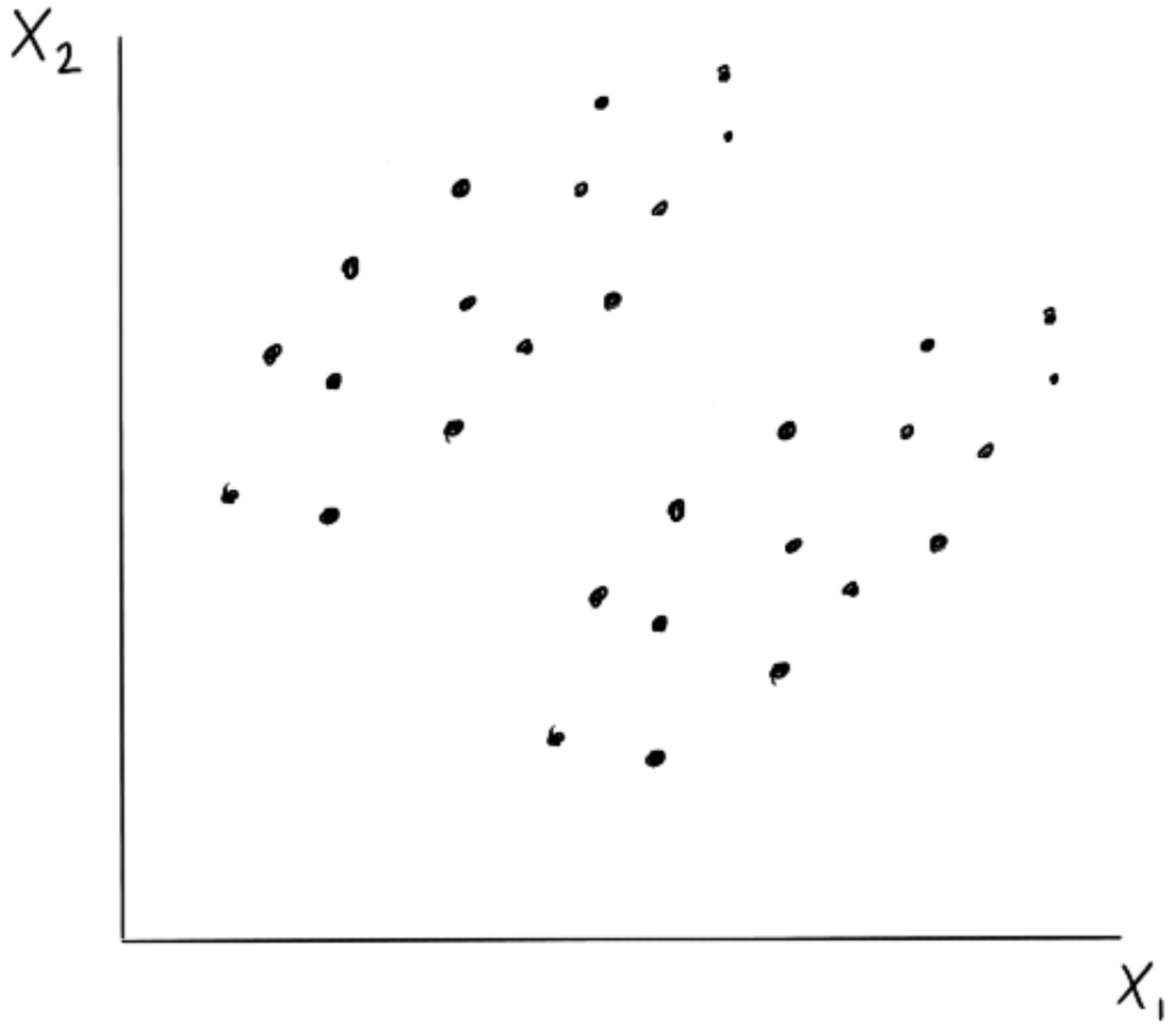

X2
3
2
1
1
2,3
1
2,3
X1

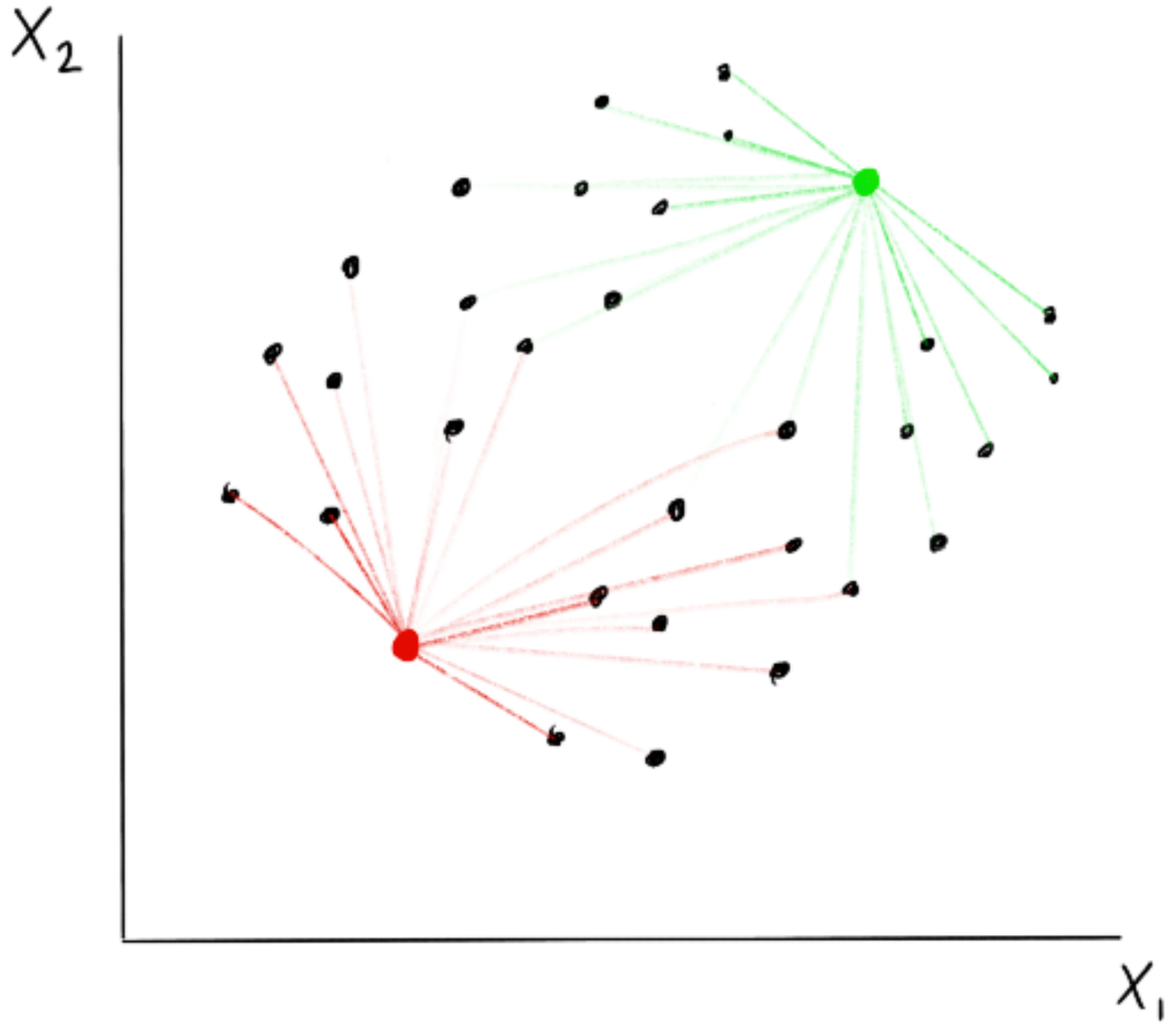
The algorith has two steps:

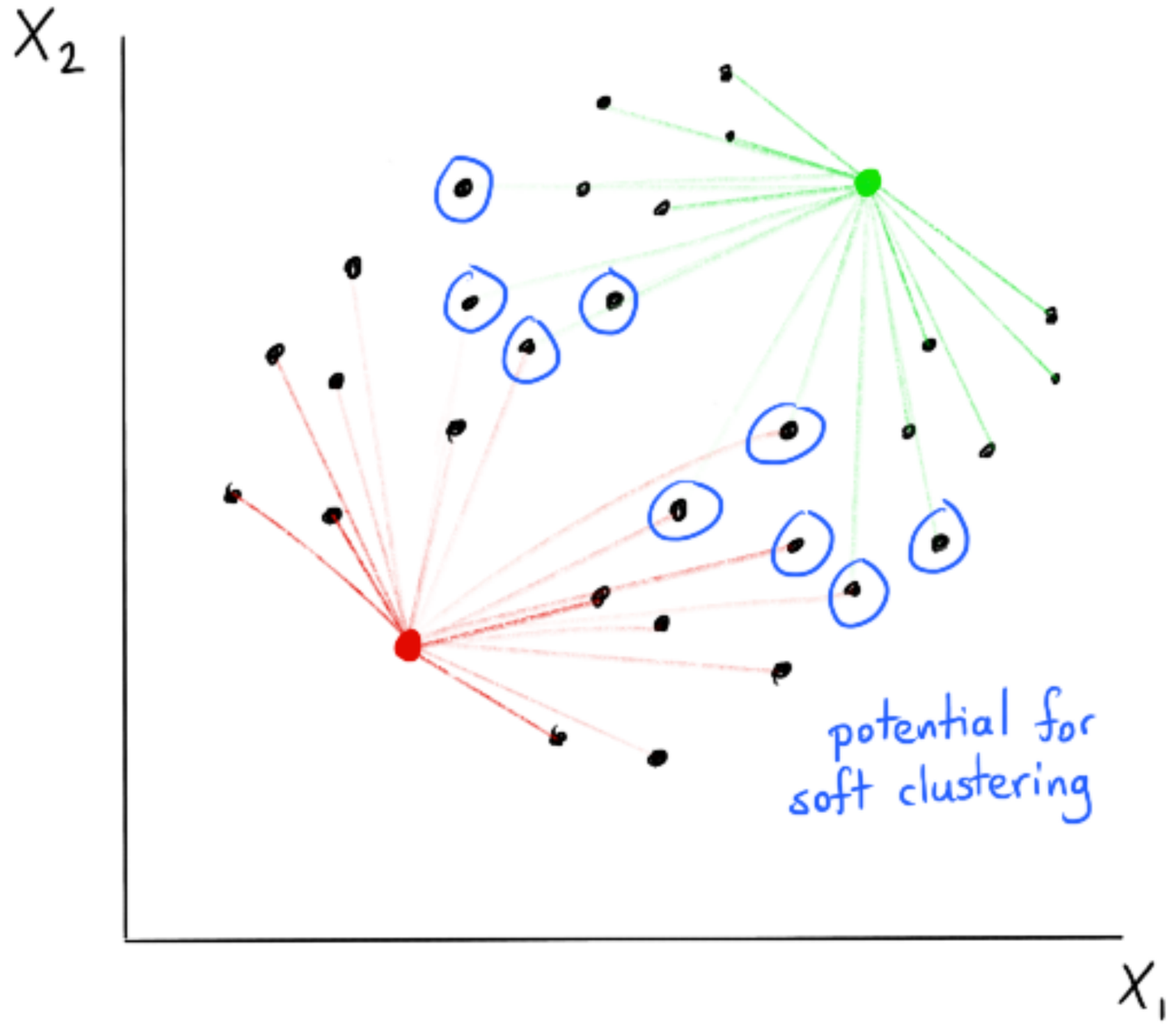
1. a step where we check distances (E)
2. a step where we move centers (M)

This (EM-algorithm) gives it a nice property: it **must** converge! The next step will always have less total distance between centroids and points (otherwise the centroids would not move).

But there's something "iffy" about all this.

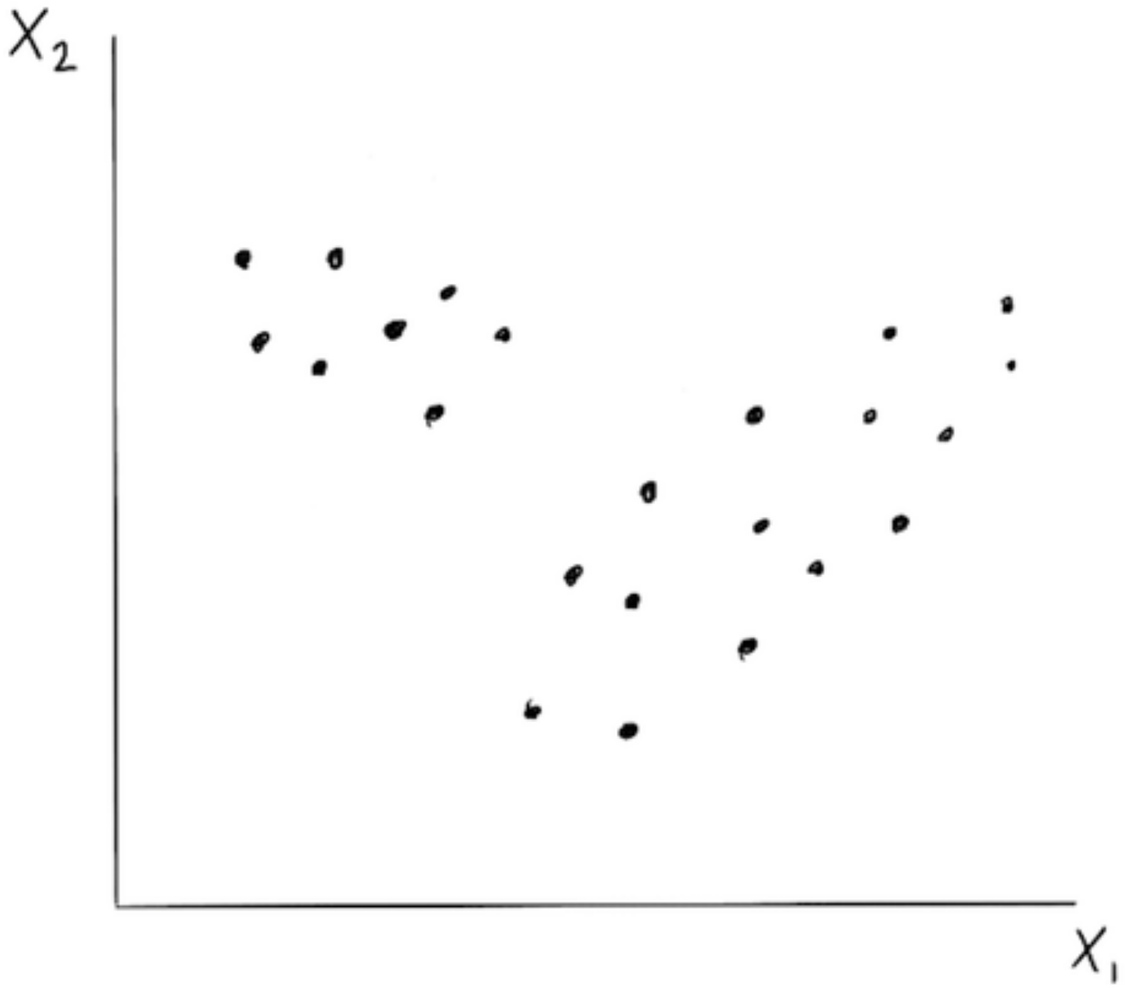

X_2
X_1

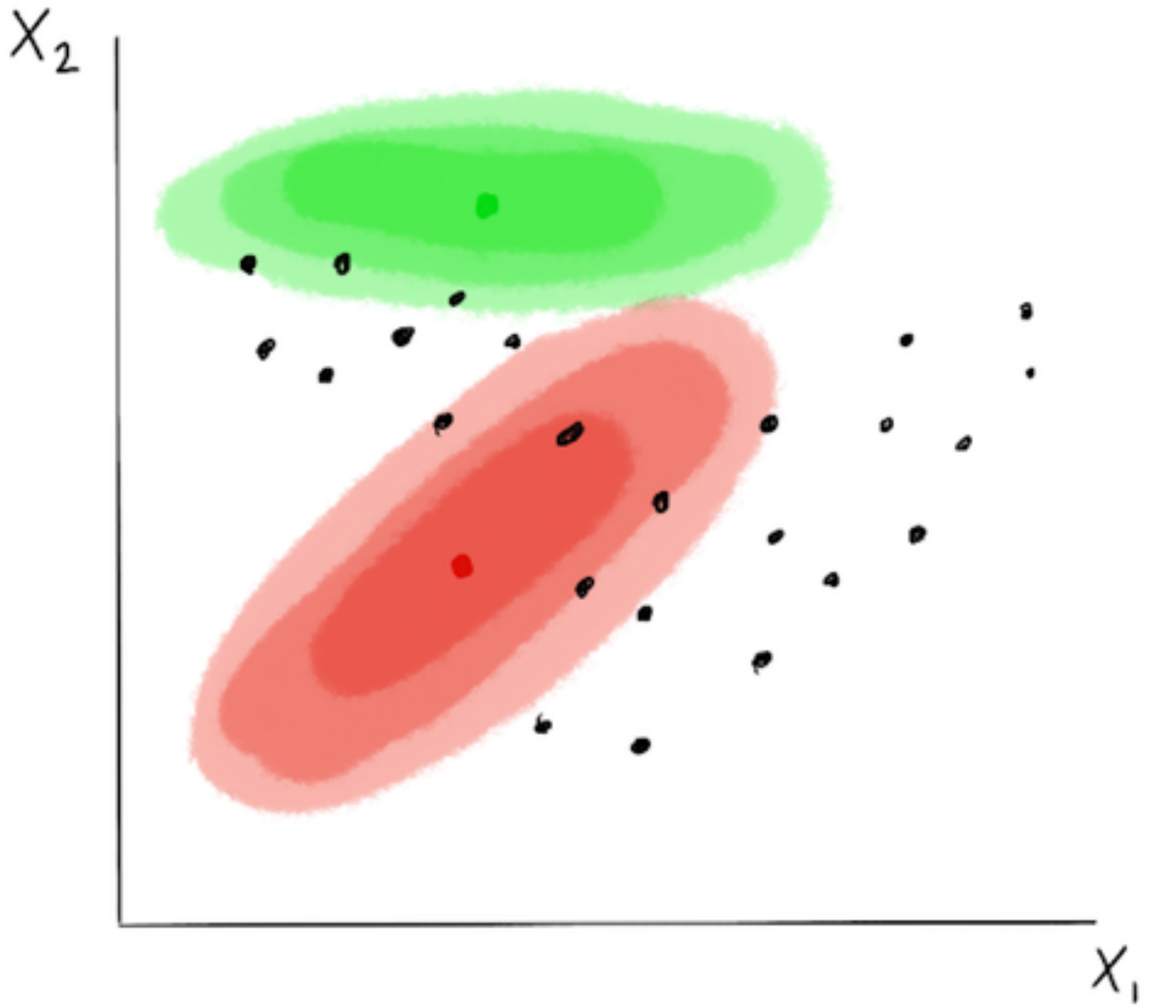

X_2
X_1

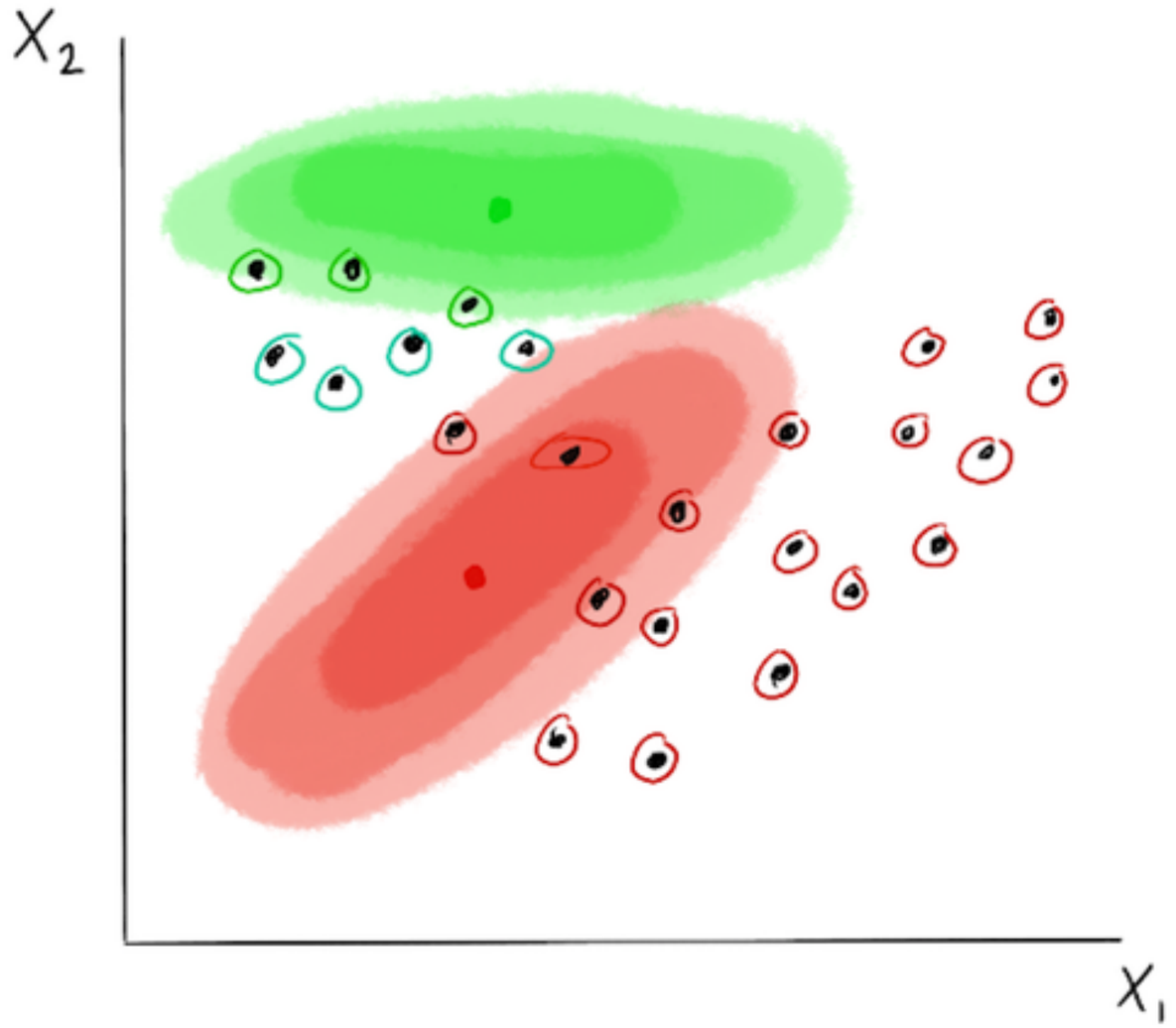

X_2
potential for
soft clustering
X_1

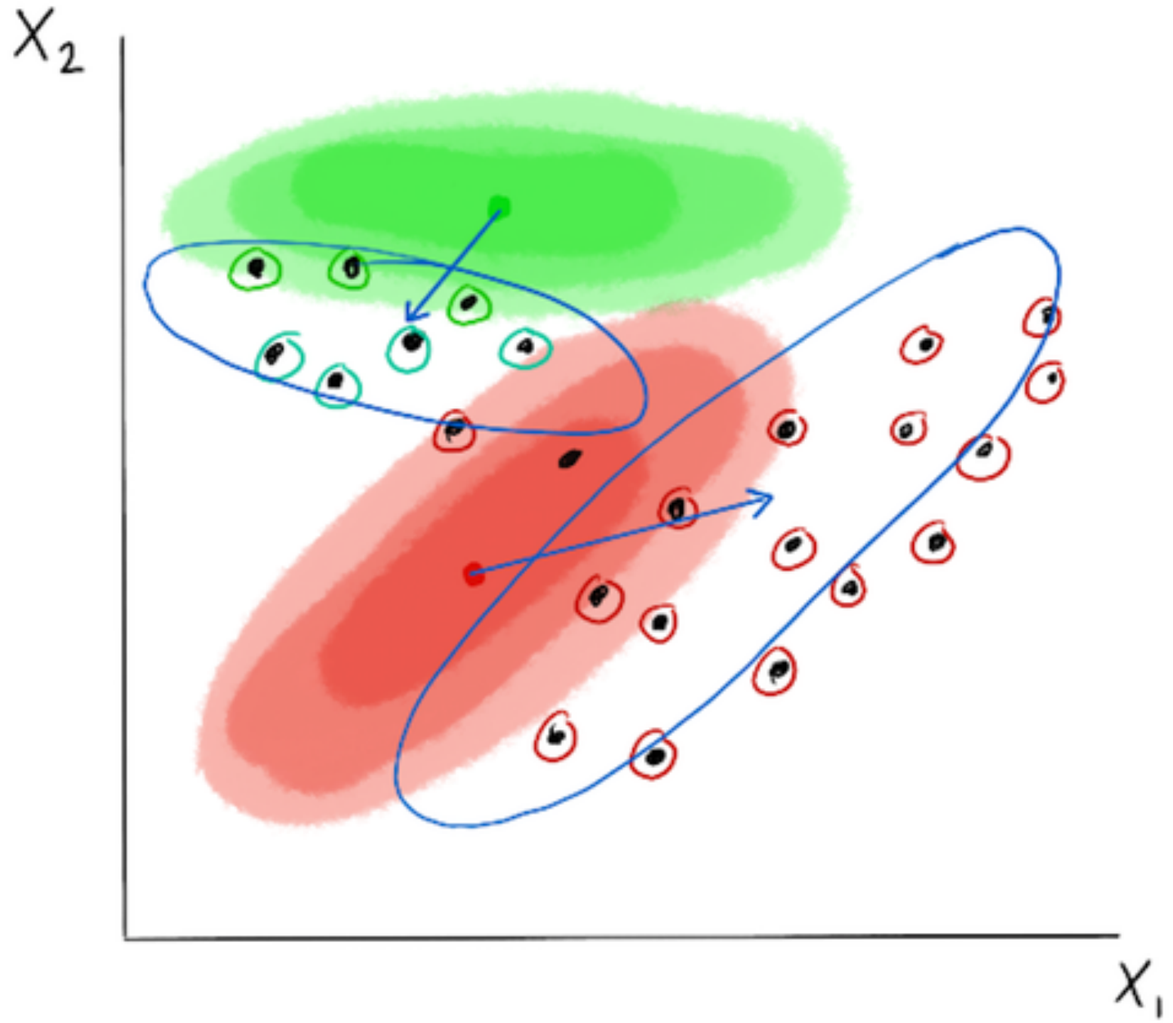
So the idea is; we'll try to do "kmeans" but instead of making hard clusters we'll try to fit multiple gaussians that can perform "soft" clusters. There will be a probability that a point belongs to a certain cluster.

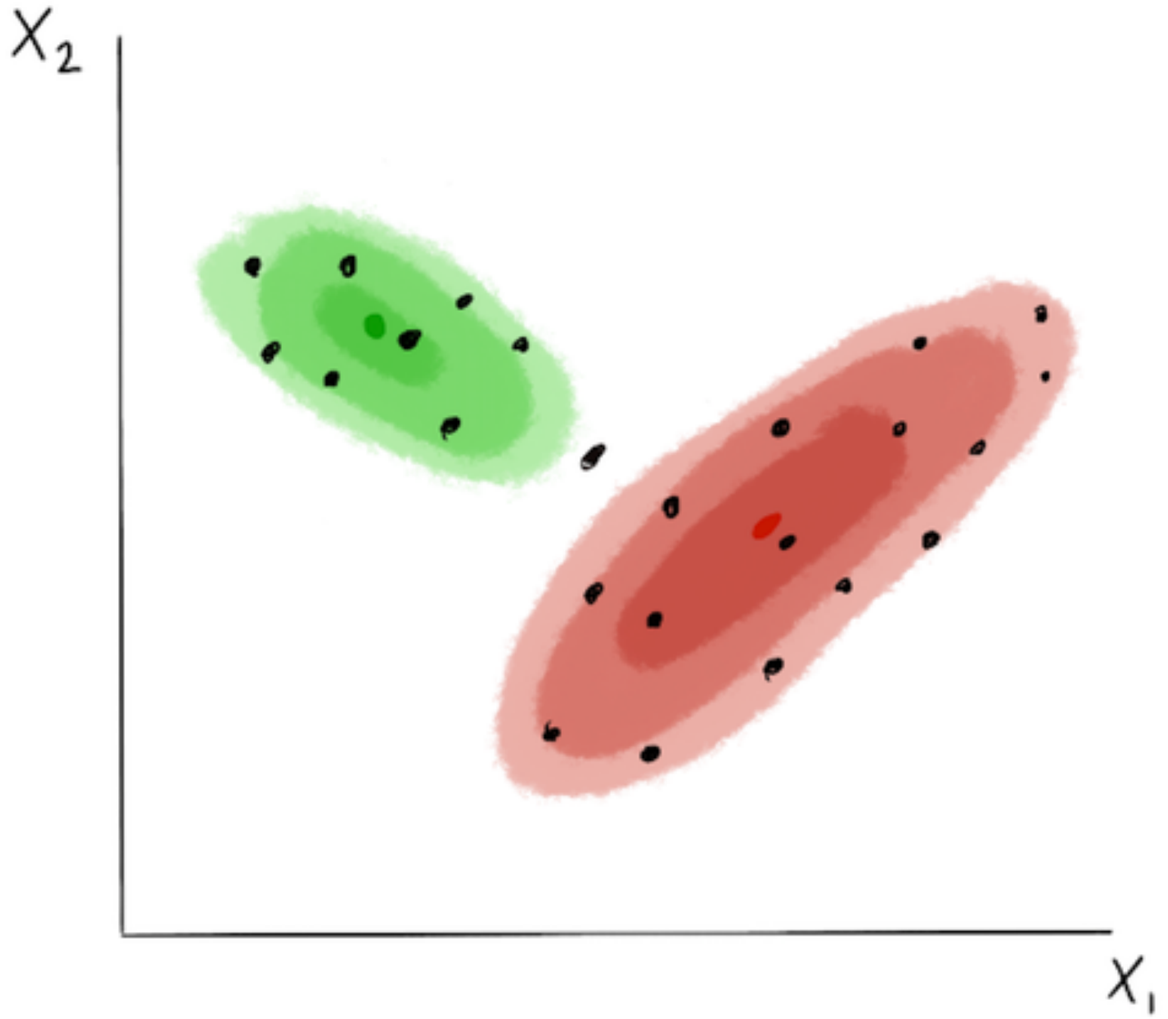
The further a point is from a center the less it influences it. Not a bad idea.

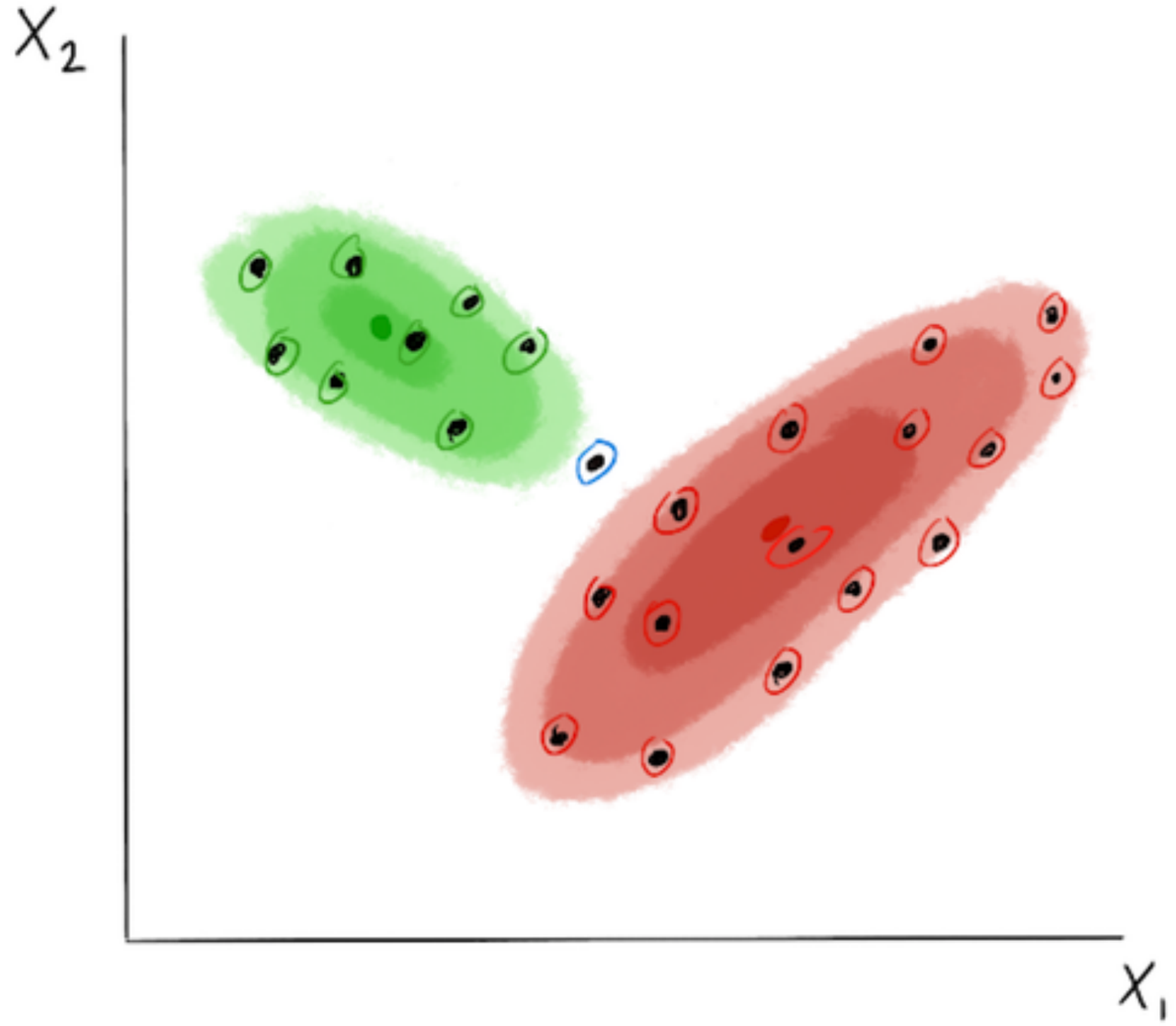

X_2
X_1

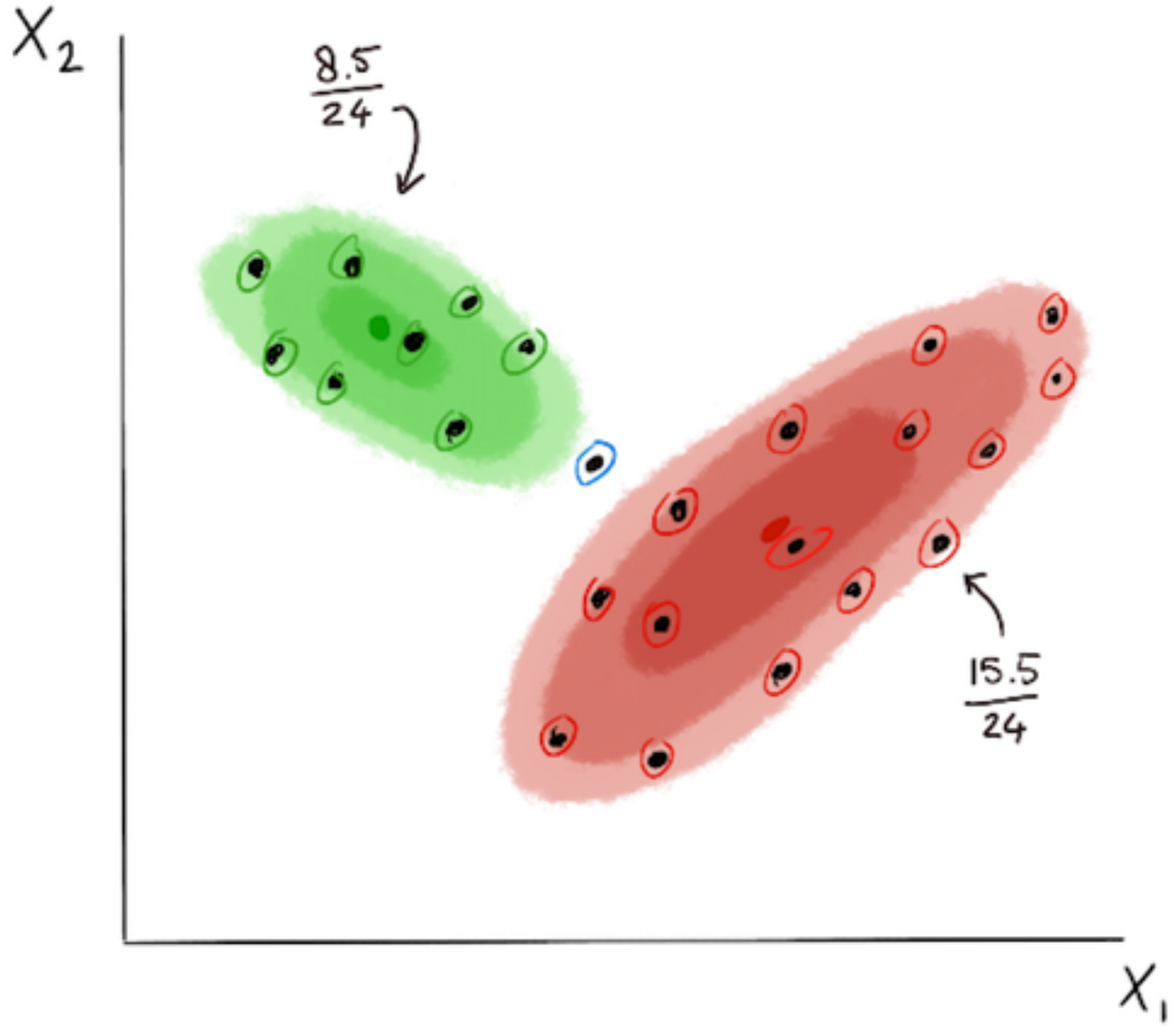

X_2
X_1


X_2
X_1

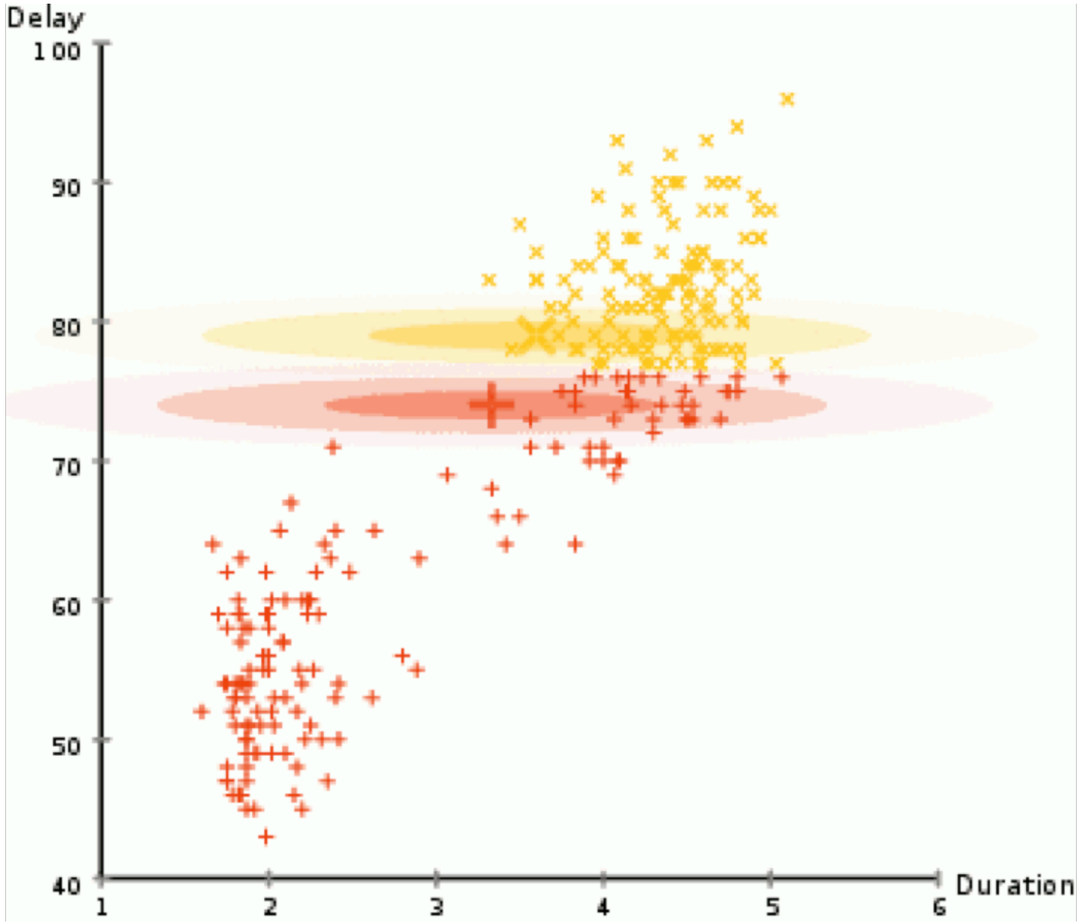

X_2
X_1


X_2
X_1


X_2
X_1


X2
8.5
24
15.5
24
X1

The GIF from Wikipedia



The Math from Wikipedia

And again, this story is usually summarised like this;

$$p(\mathbf{x}) = \sum_{i=1}^{k} \pi_i \times N(\mu_i, \sigma_i)$$

There k gaussians that we'll try to fit and each gaussian gets an associated weight k.

For every centroid k we'll calculate;

$$n_k = \sum_{i=1}^{n} p_k(i)$$

$$\mu_k = \frac{1}{n_k} \sum_{i=1}^{n} p_k(i) x_i$$

$$\sigma_k^2 = \frac{1}{n_k} \sum_{i=1}^{n} p_k(i) (x_i - \mu_k)^2$$

$$\pi_k = \frac{n_k}{n}$$

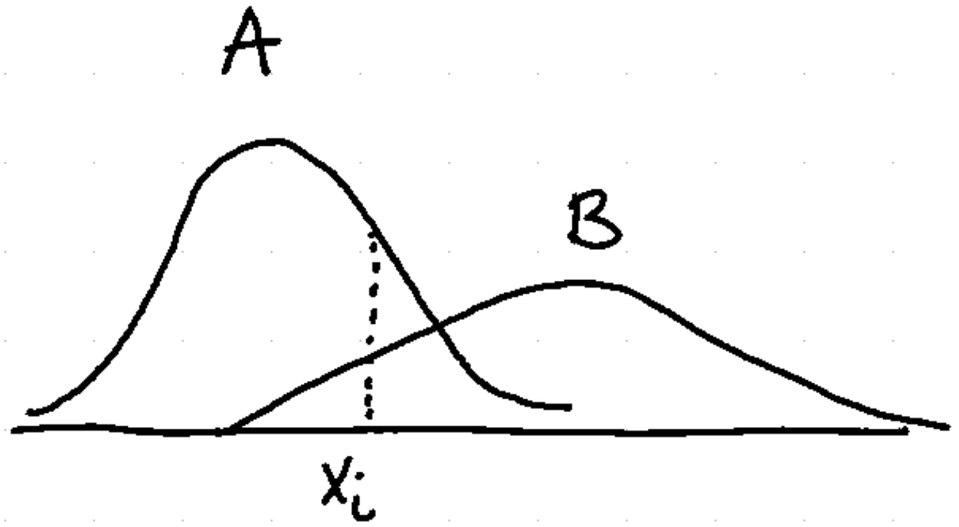
The downside of this approach of explaining things is that it distracts you from the coolest observation:

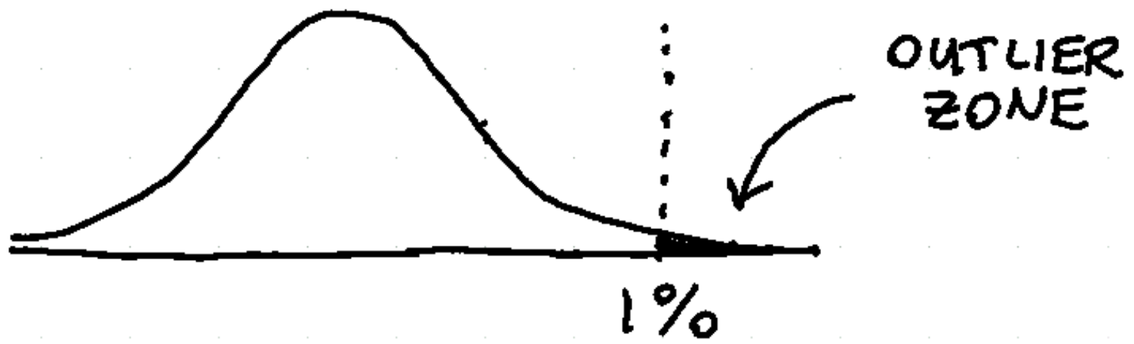
The downside of this approach of explaining things is that it distracts you from the coolest observation:

THIS HAS SO MANY APPLICATIONS

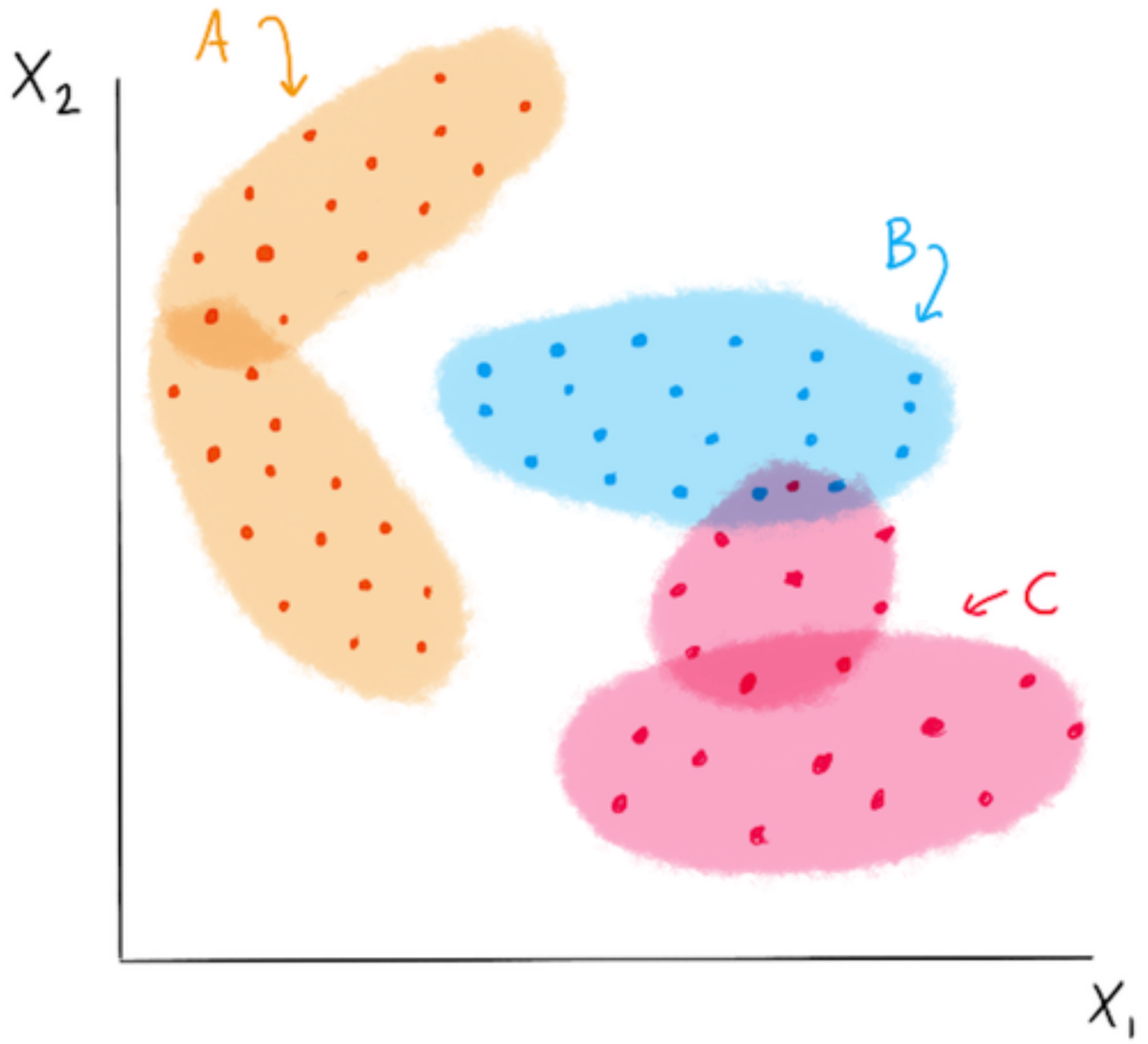
Applications beyond clustering help motivate learning.

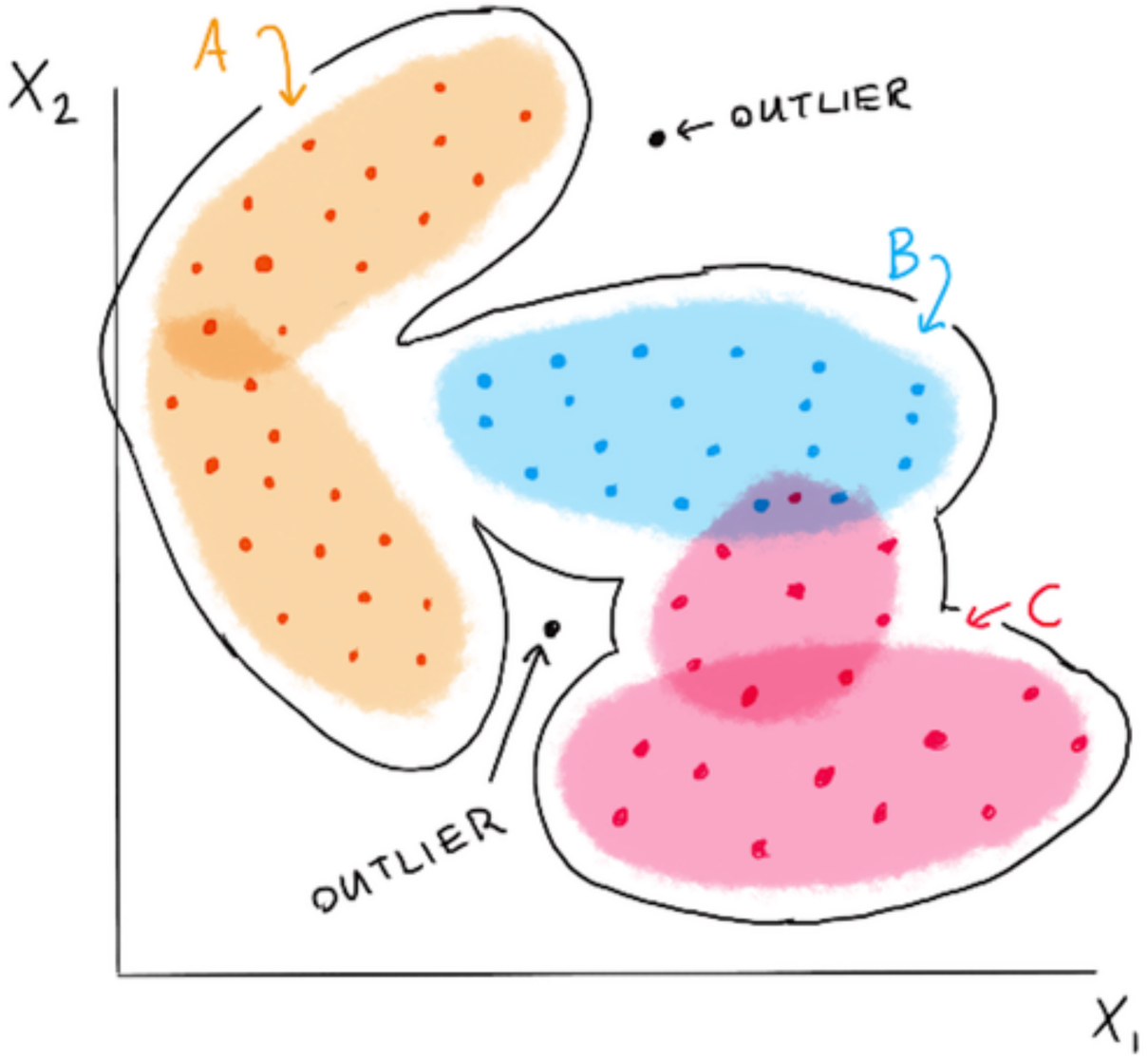
Let's do this idea but now in higher dimensions.

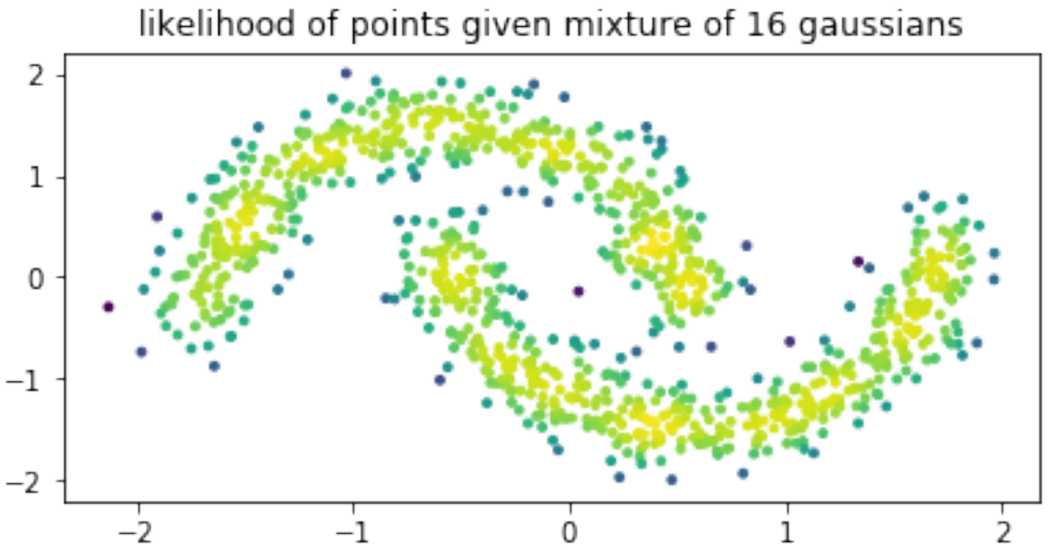


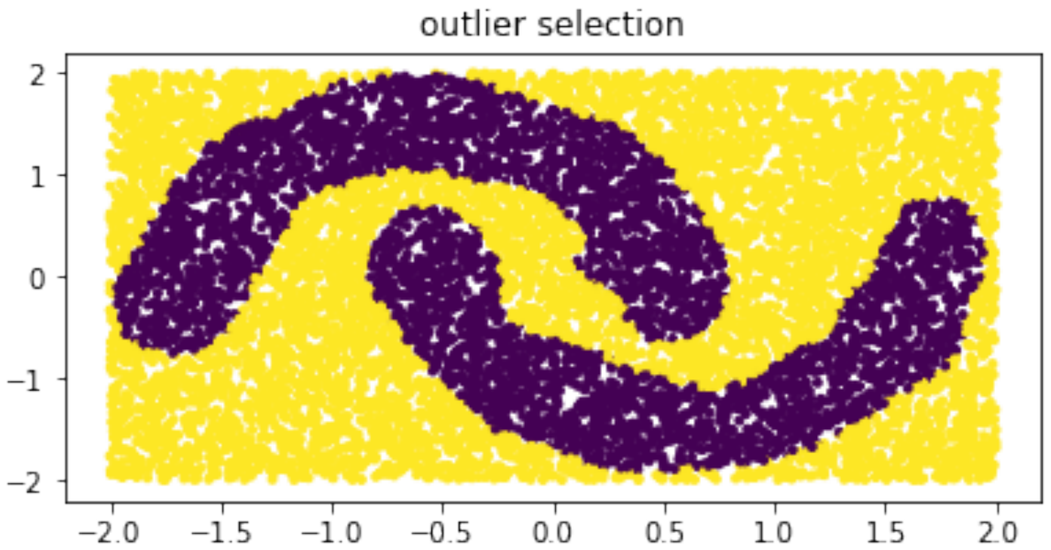


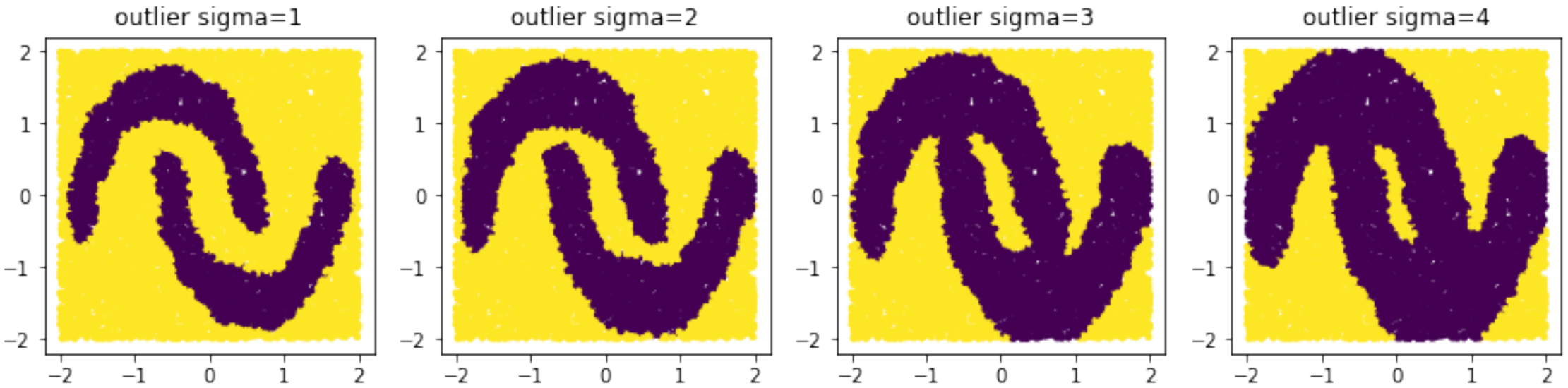
$$p(x_i = A) = \frac{f_A(x_i)}{f_A(x_i) + f_B(x_i)}$$

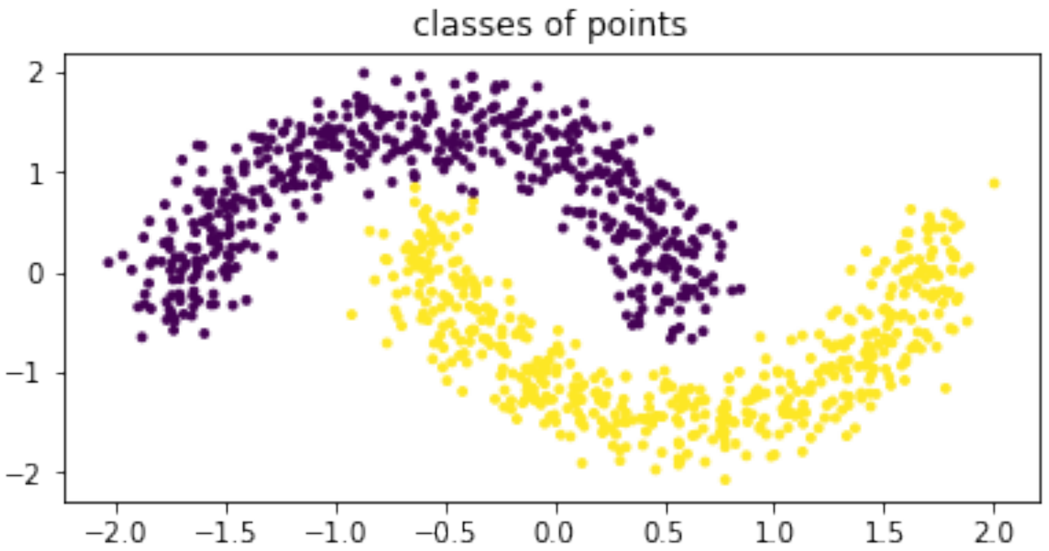

A
B
C
X_2
X_1

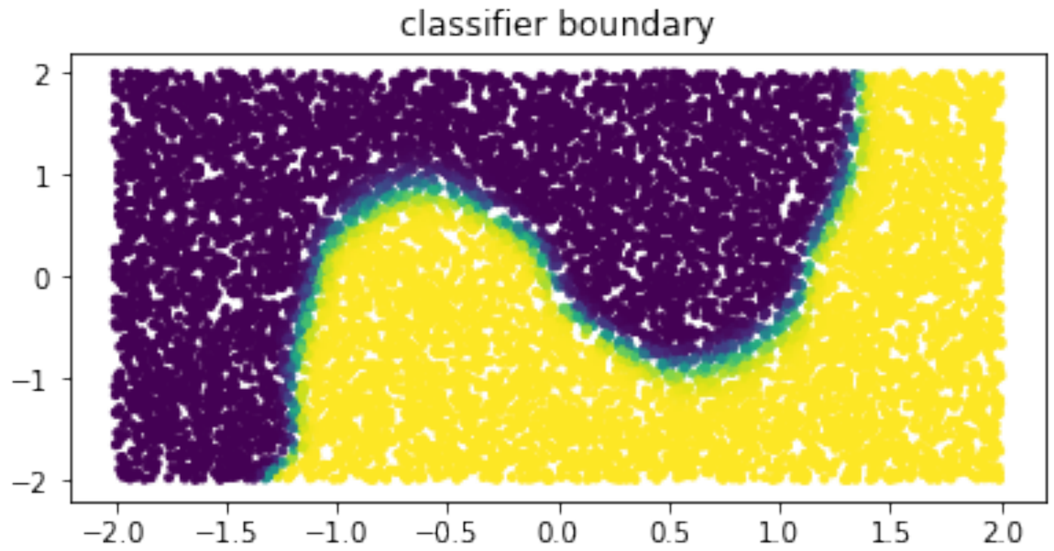

A
X_2
OUTLIER
B
C
OUTLIER
X_1


likelihood of points given mixture of 16 gaussians
2
1
0
−1
−2
−2
−1
0
1
2

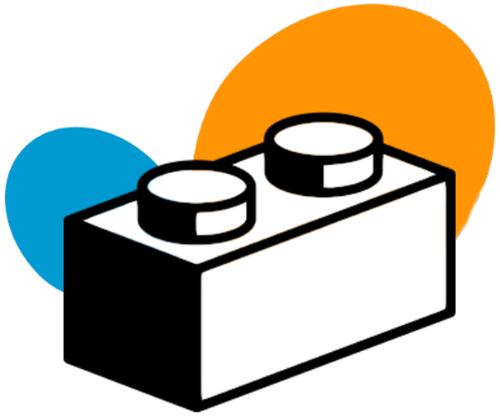

outlier selection
2
1
0
−1
−2
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


outlier sigma=1
outlier sigma=2
outlier sigma=3
outlier sigma=4
2
1
0
−1
−2
−2
−1
0
1
2


classes of points
2
1
0
−1
−2
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


classifier boundary
2
1
0
−1
−2
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

scikit lego

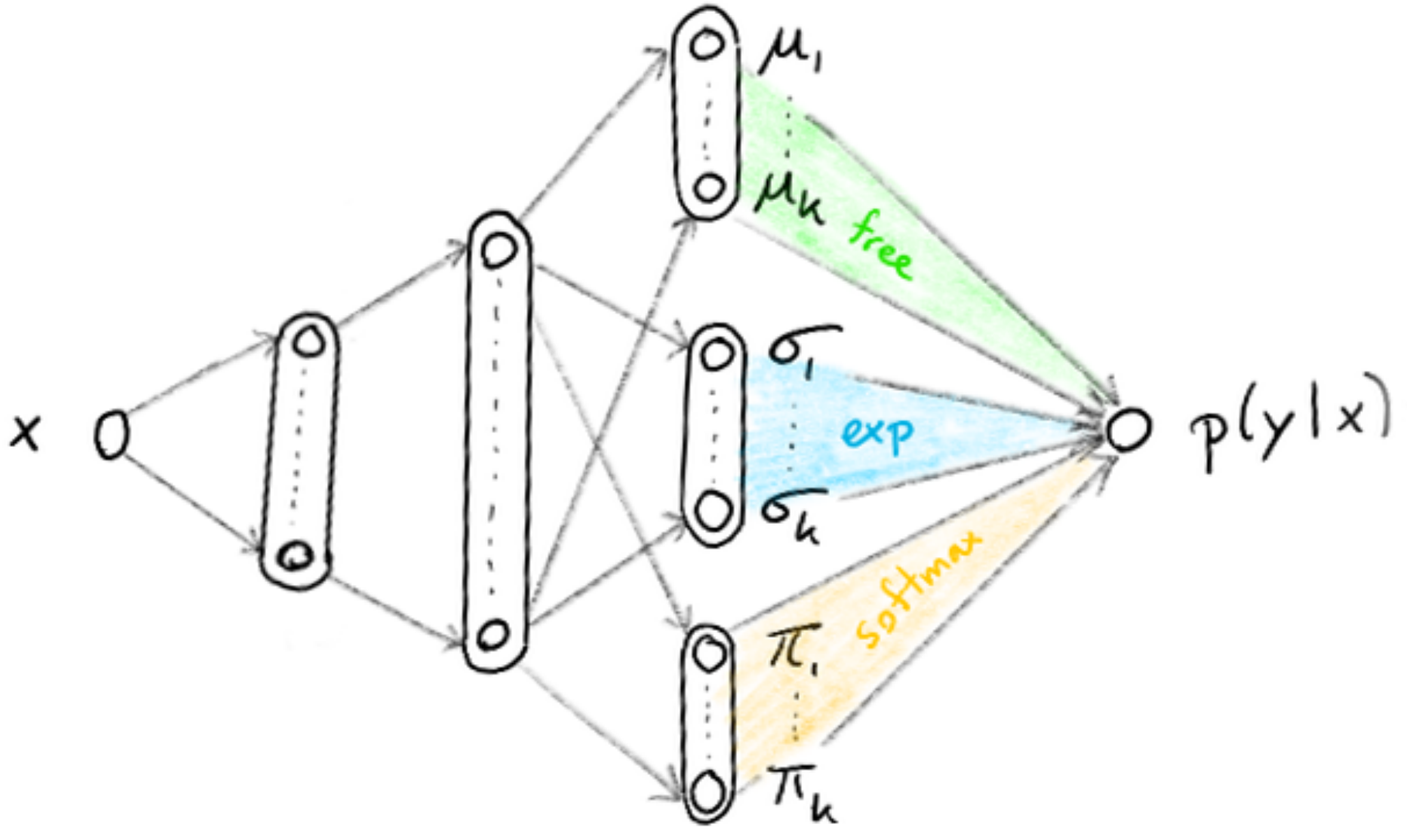


```
from sklego.mixture import GMMClassifier
from sklego.mixture import GMMOutlierDetector
```

Open sourced! Feel free to find us on github.

But what about deep learning?

Well ... we can add useful properties to neural networks by glueing a gaussian on top of it.


x
μ_1
μ_k
free
σ_1
exp
σ_k
softmax
π_1
π_k
p(y|x)

$$p(\mathbf{y}|\mathbf{x}) = \sum_{i=1}^{k} \pi_i \times N(\mu_i, \sigma_i)$$

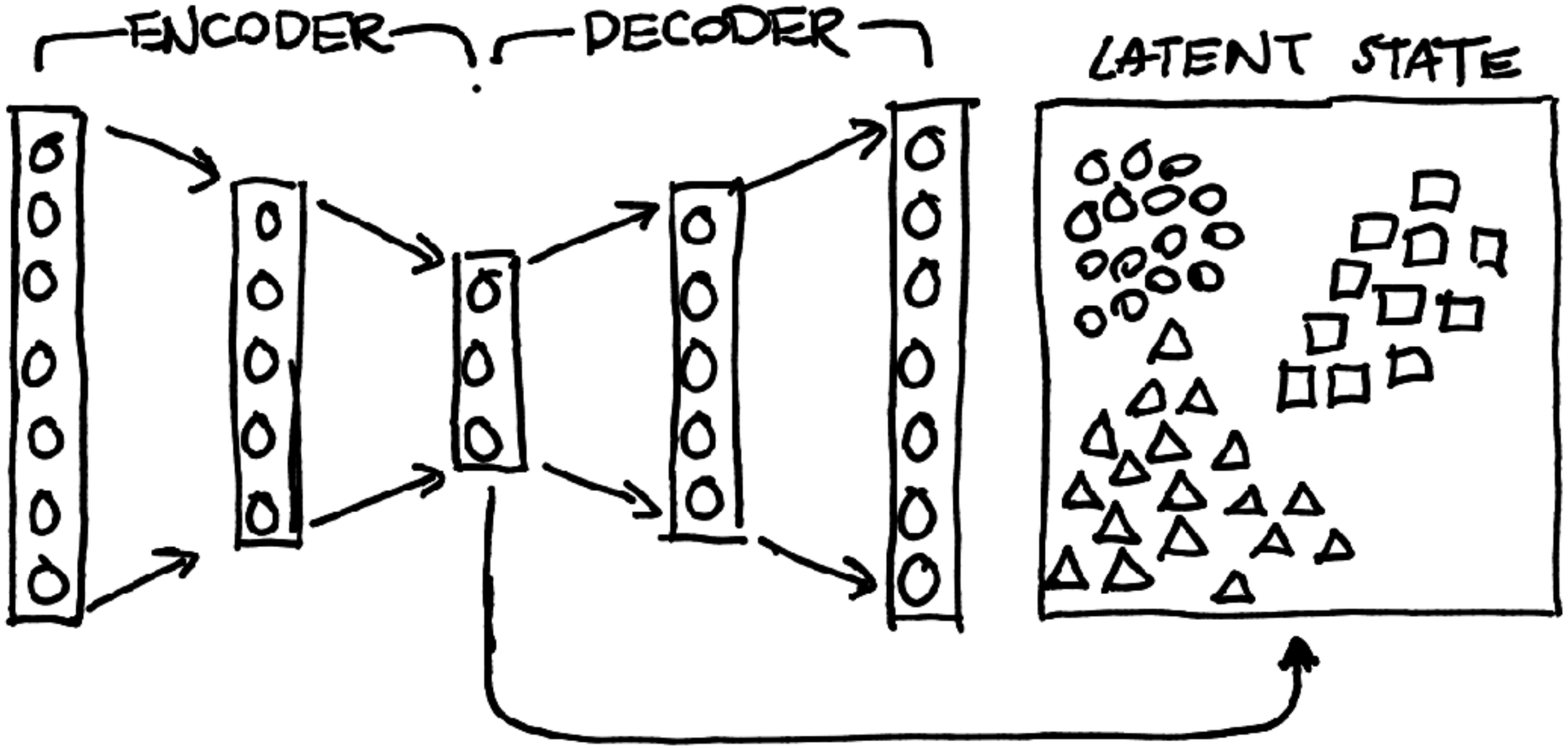
— a $\boldsymbol{\mu}$ layer with nodes $\{\mu_1 \dots \mu_k\}$ which will denote the mean of a normal distribution

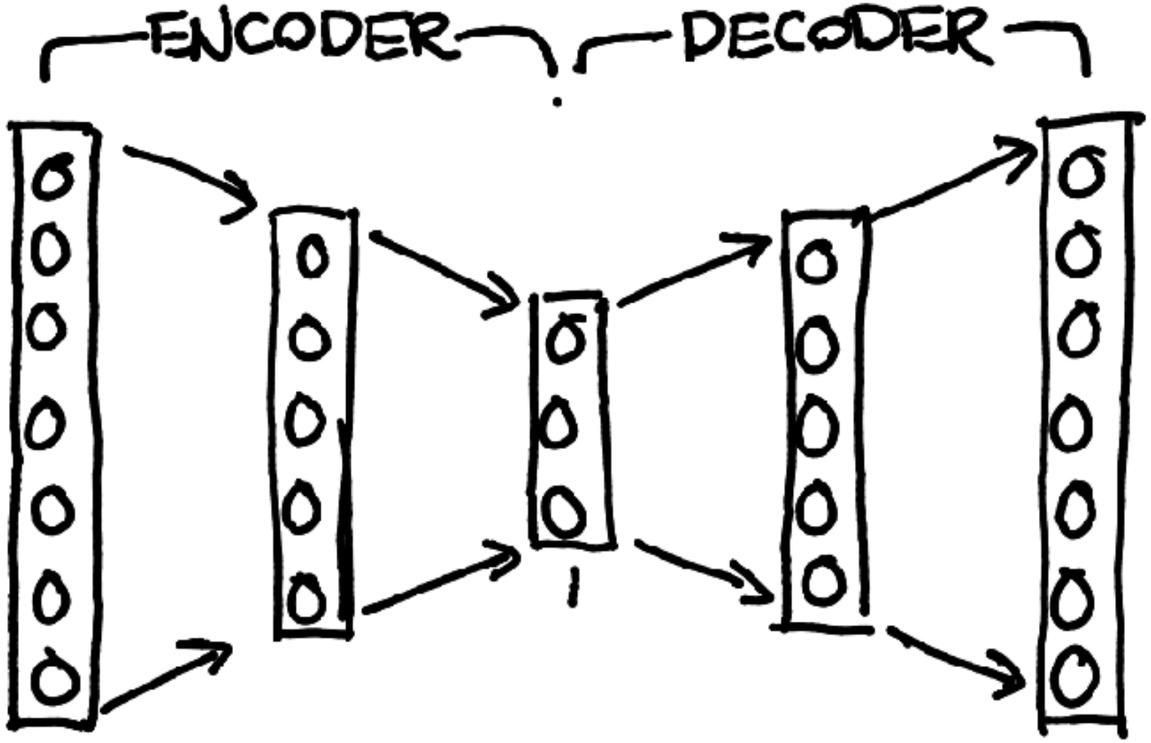
— a $\boldsymbol{\sigma}$ layer with nodes $\{\sigma_1 \dots \sigma_k\}$ which will denote the deviation of a normal distribution

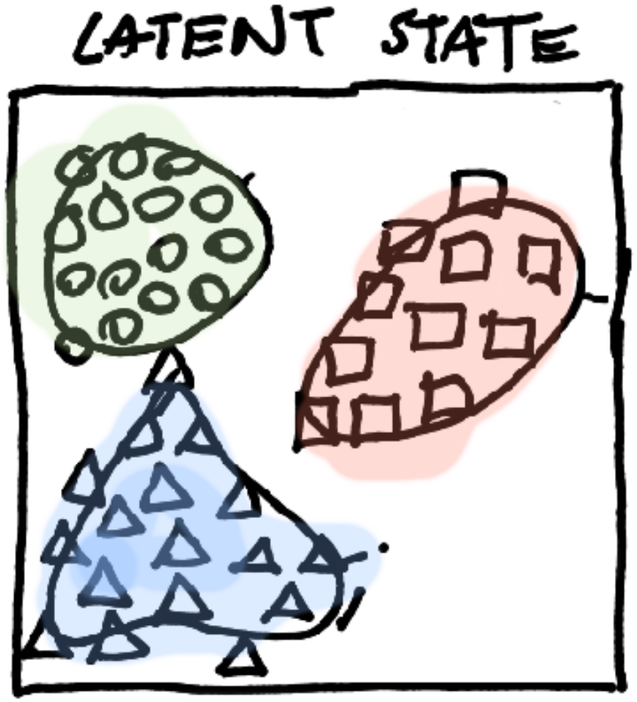
— a $\boldsymbol{\pi}$ layer with nodes $\{\pi_1 \dots \pi_k\}$ which will denote the weight of the associated normal distribution in the resulting prediction

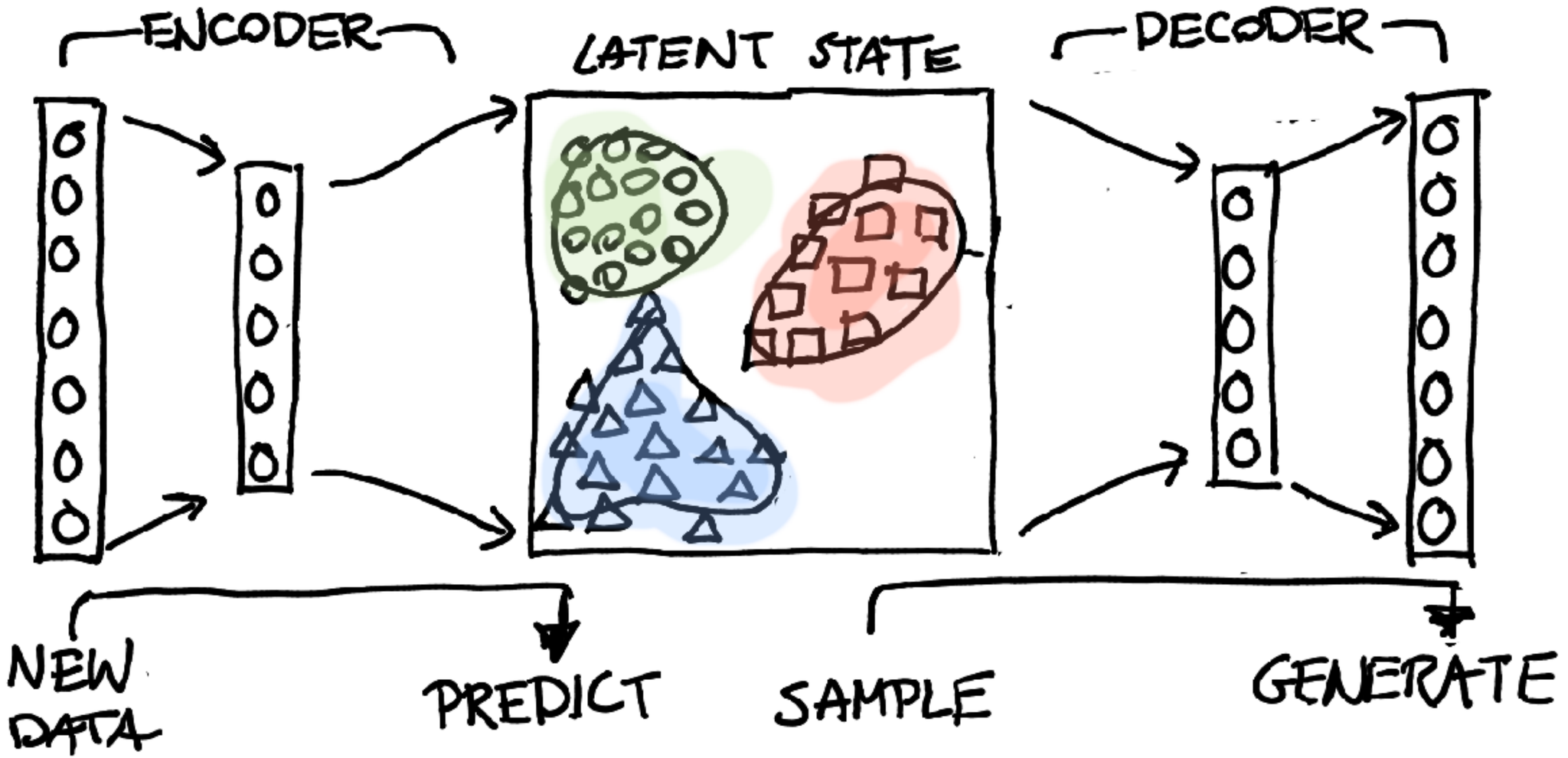
But what about deep learning?

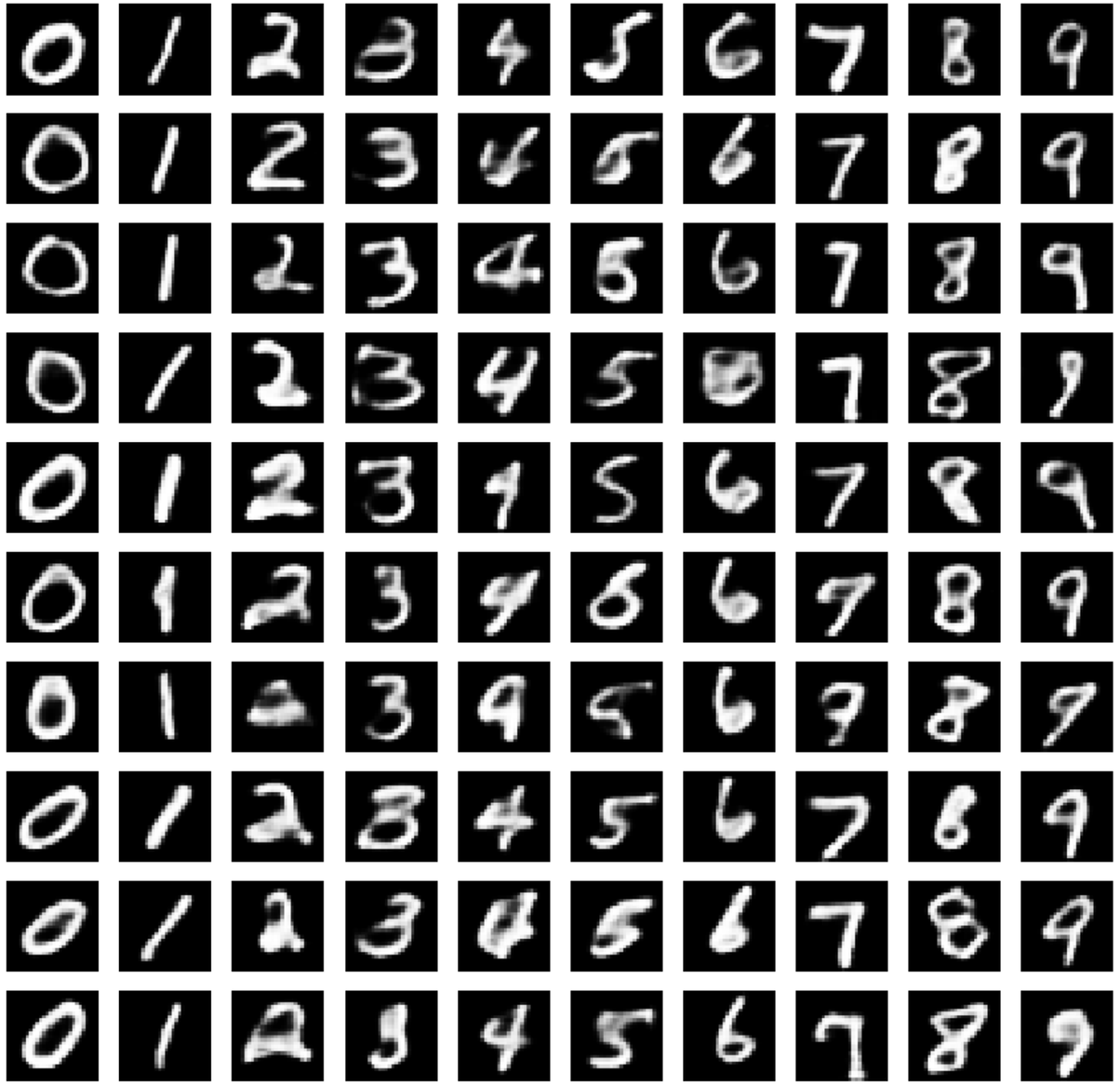
But we can also add different properties.

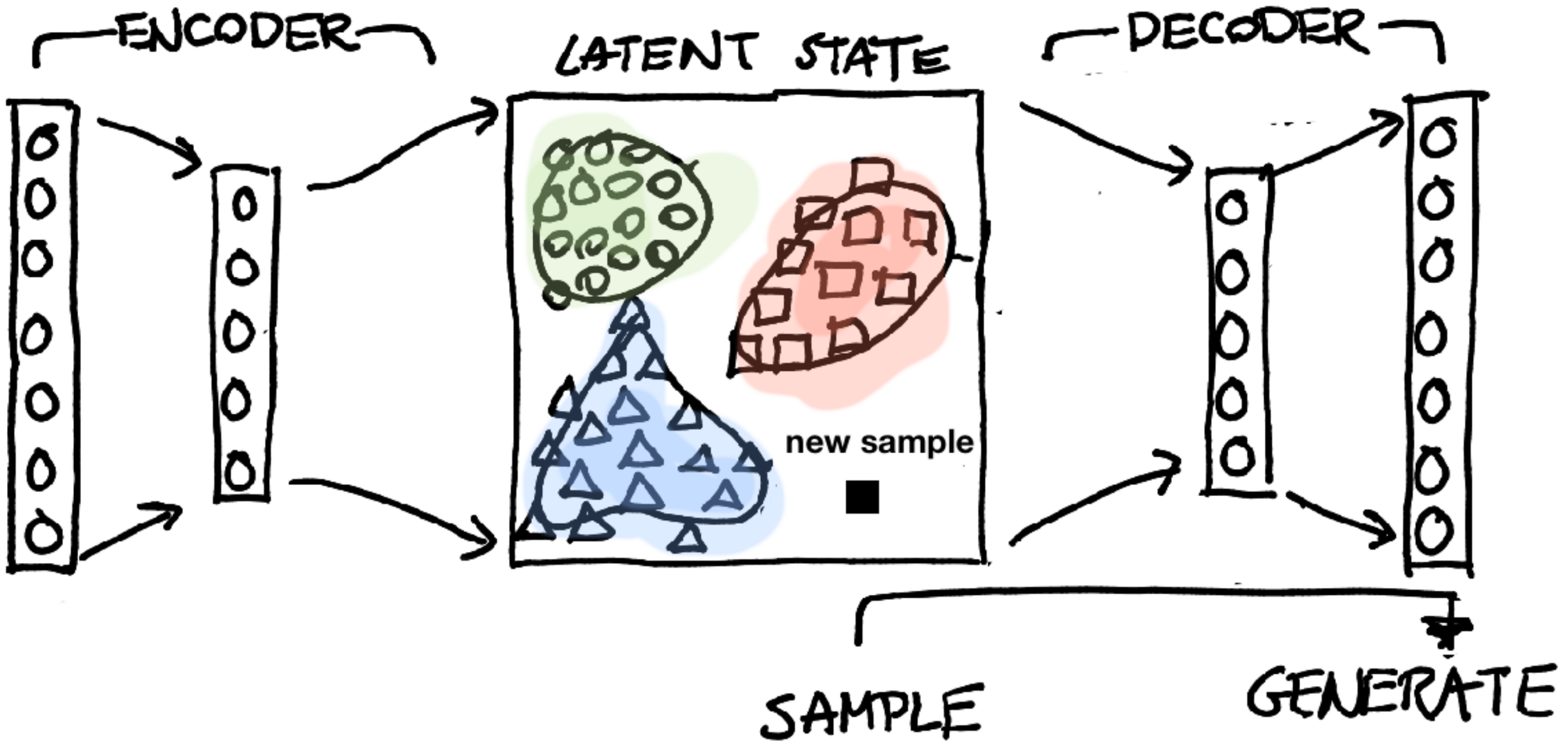

ENCODER
DECODER
LATENT STATE

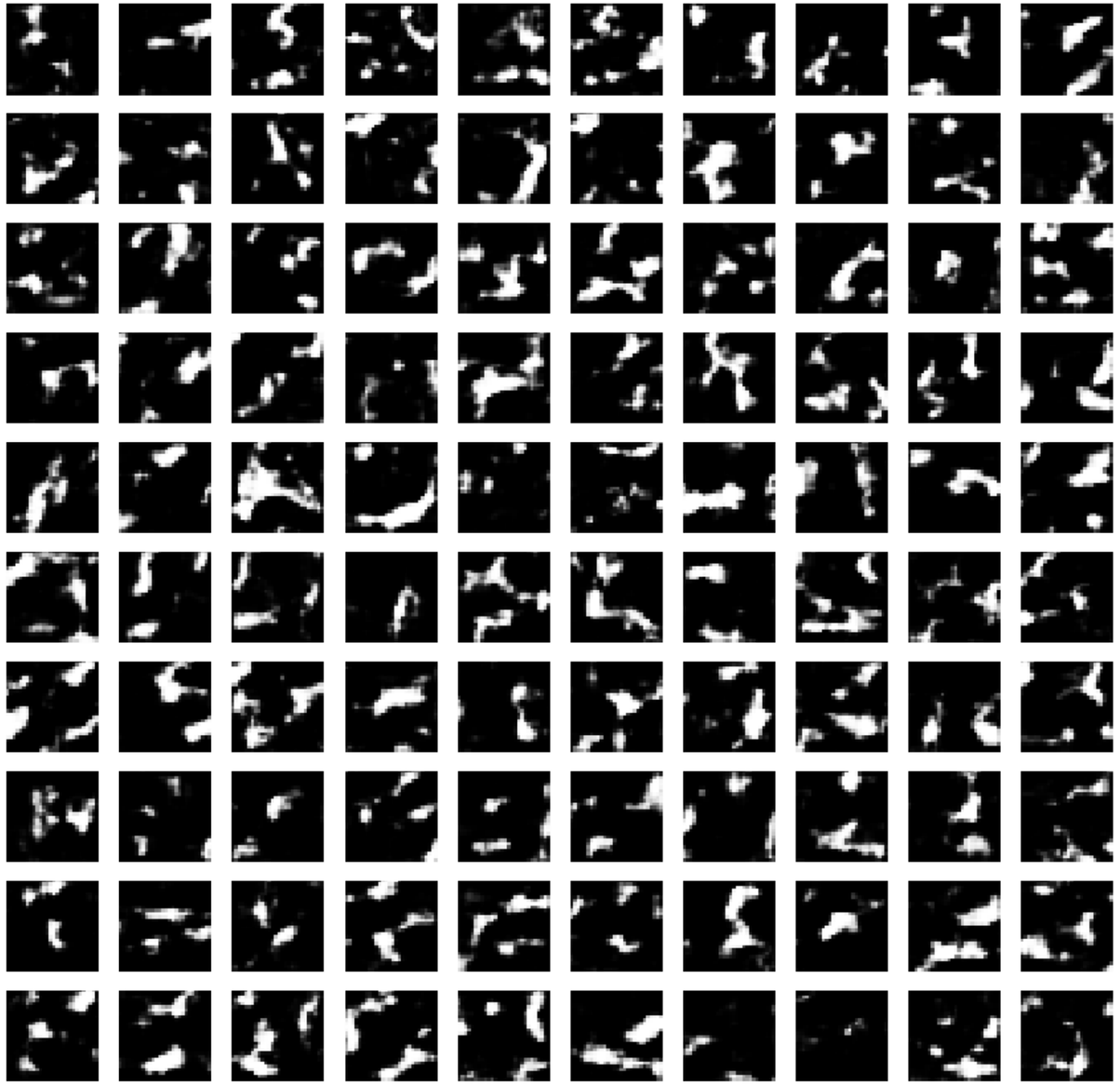

ENCODER
DECODER


LATENT STATE

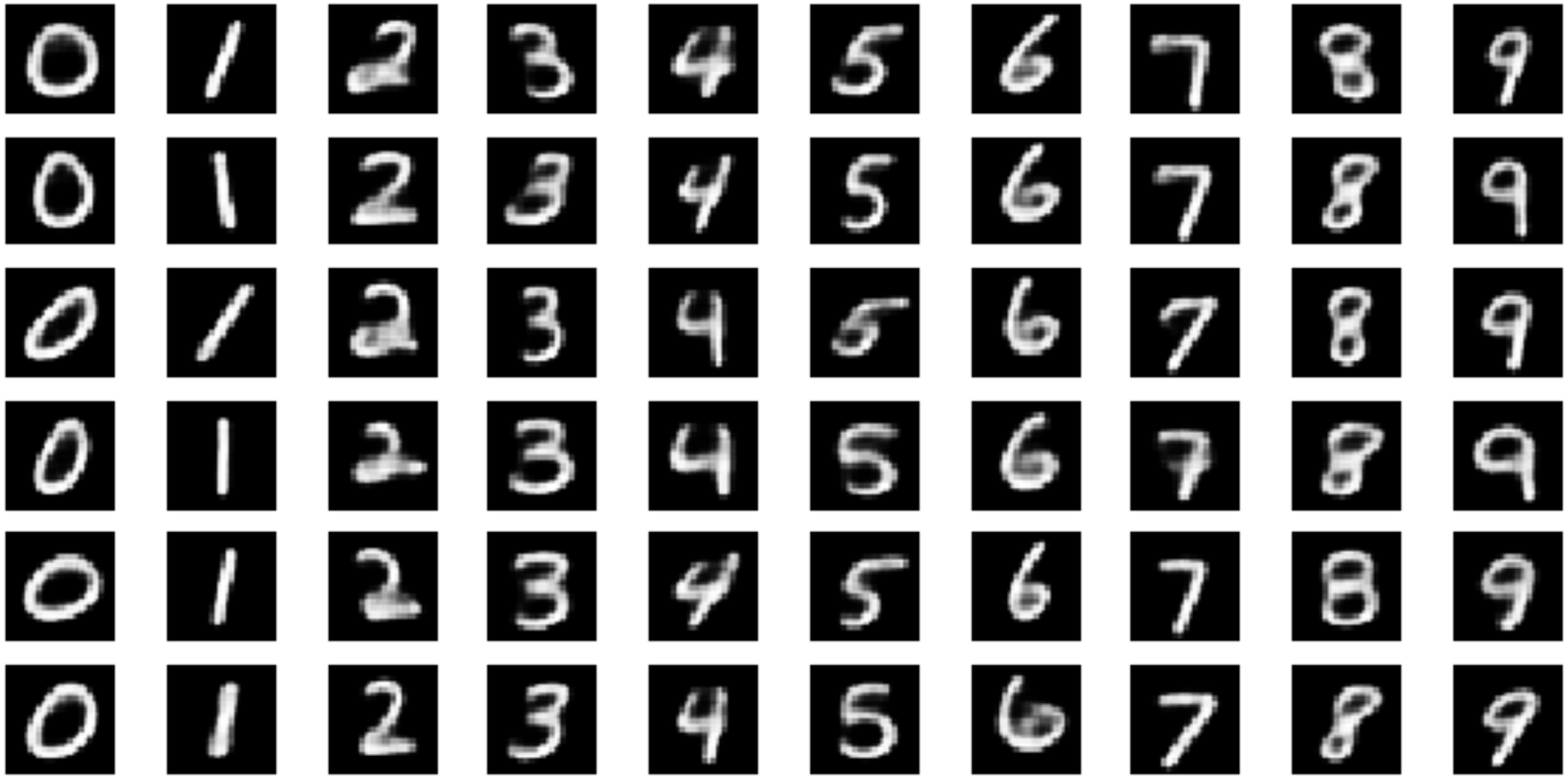

ENCODER
LATENT STATE
DECODER
NEW DATA
PREDICT
SAMPLE
GENERATE




ENCODER
LATENT STATE
DECODER
new sample
SAMPLE
GENERATE



Articulating Styles

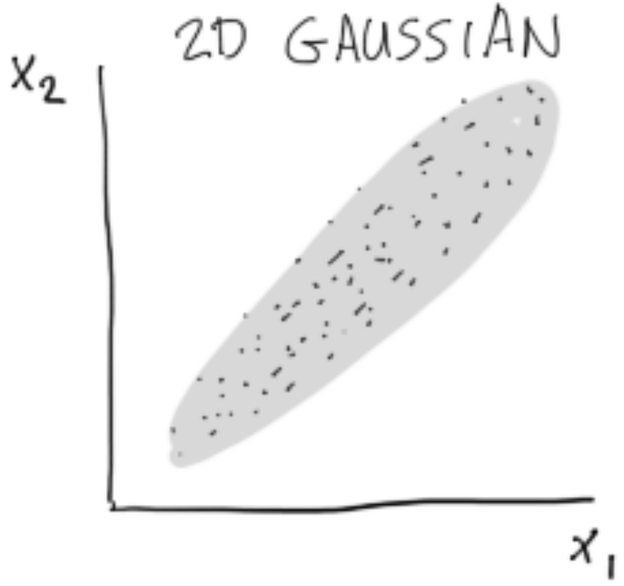


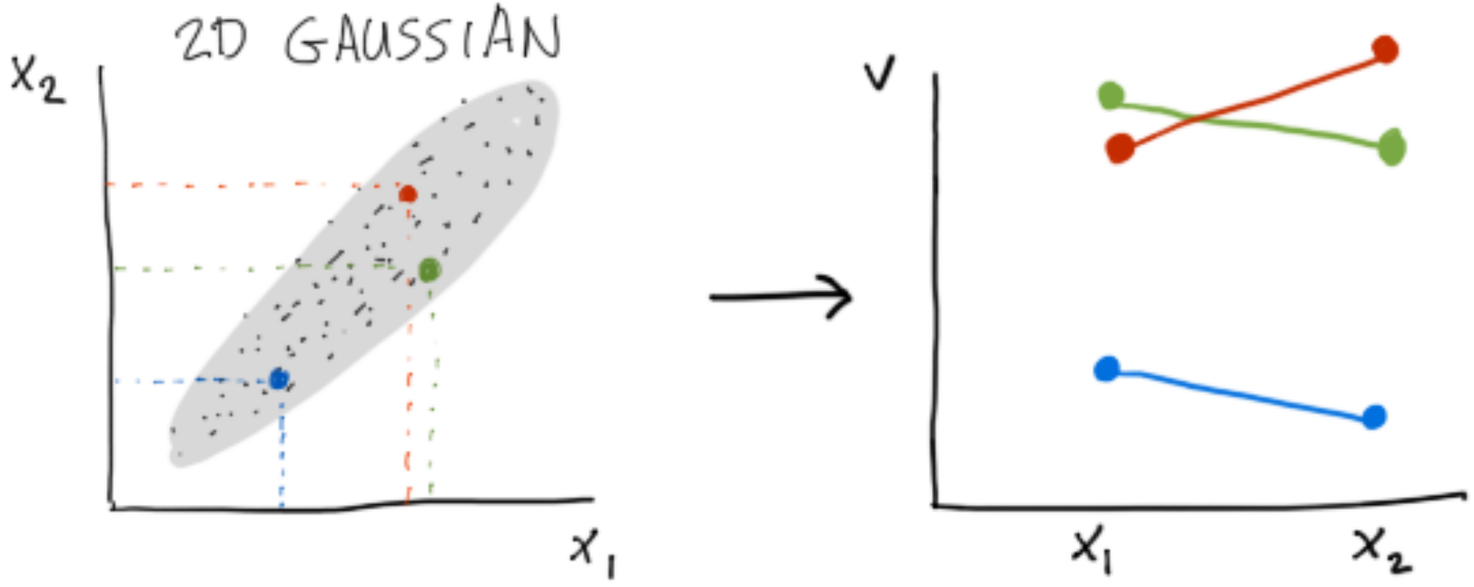
Gaussian Processes

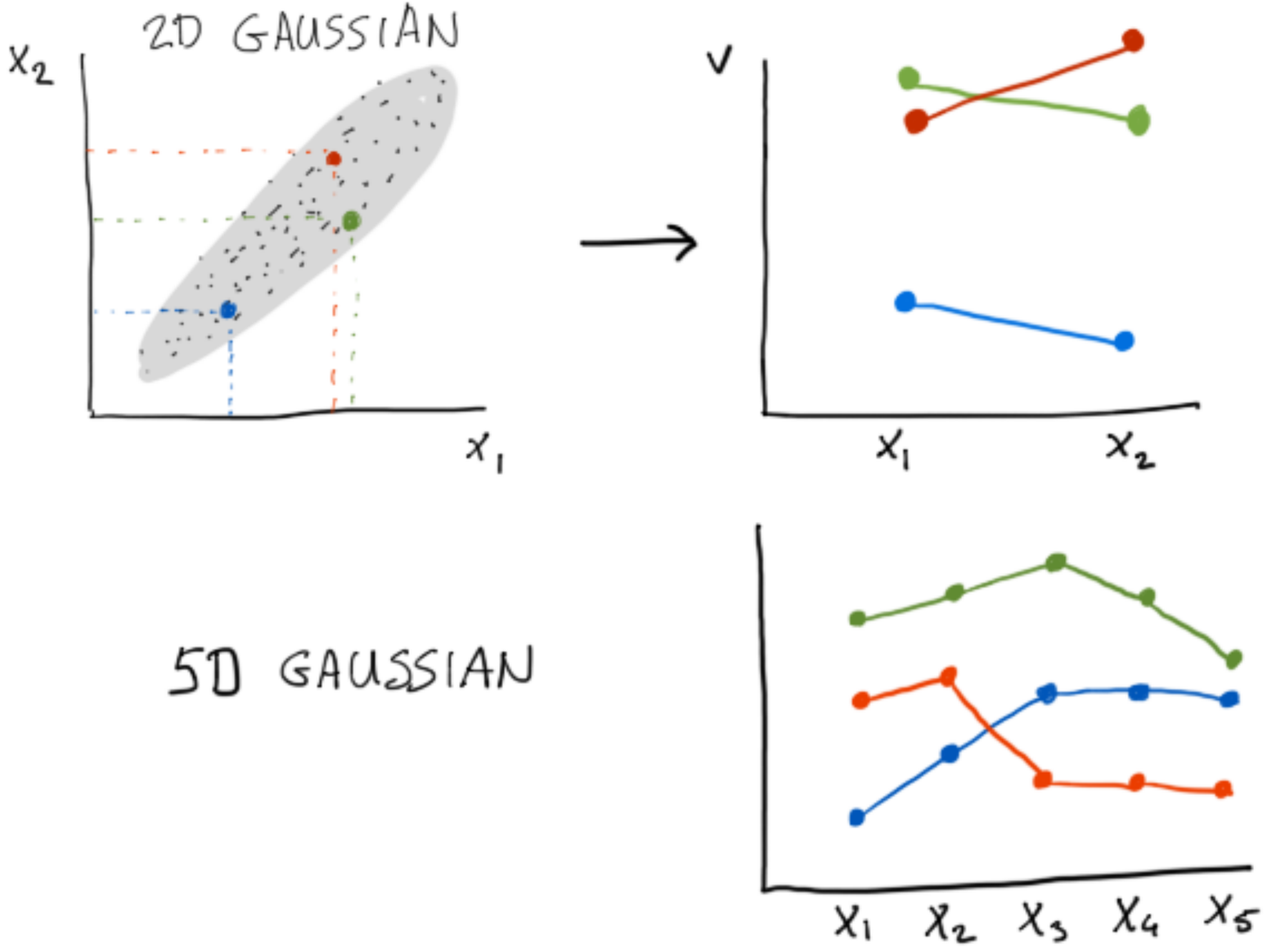
We're nearing the live coding part.

But I am going to attempt to get some intuition in you before I do.

If you've got some coffee, now is the time to drink it. We need to increase the dimensionality to understand this next part.

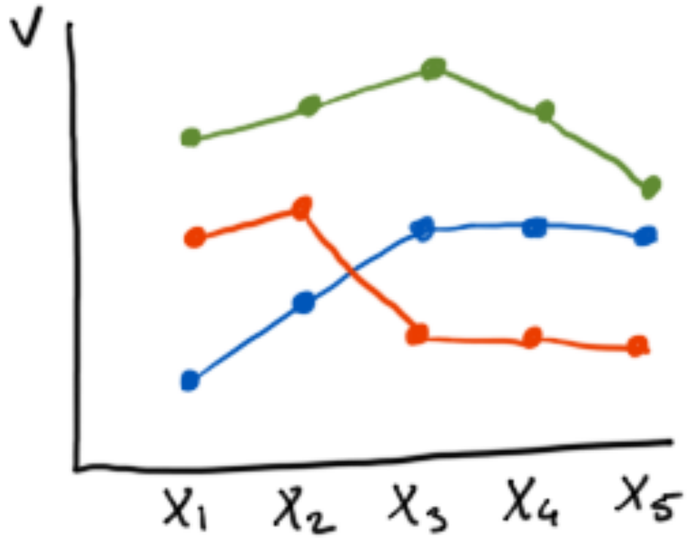

2D GAUSSIAN
x_2
x_1


2D GAUSSIAN
x_2
x_1
v
x_1
x_2


2D GAUSSIAN
x_2
x_1
v
x_1
x_2
5D GAUSSIAN
x_1
x_2
x_3
x_4
x_5

$$p(x) \sim N(\mu, \Sigma)$$

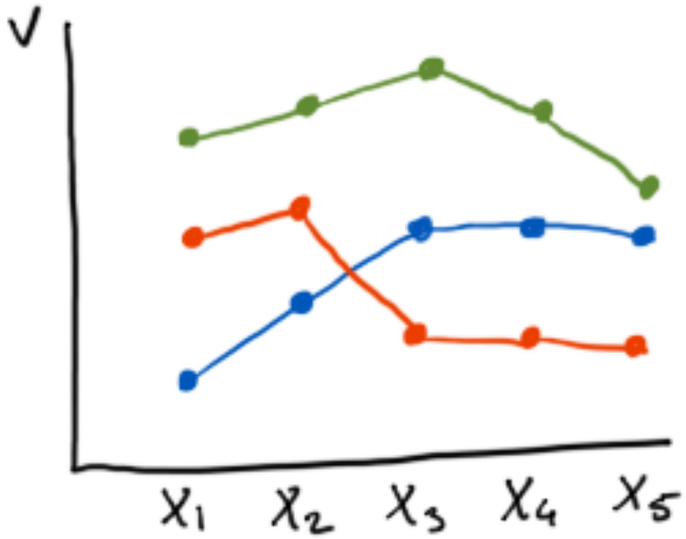
$$\Sigma = \begin{bmatrix} 5 \times 5 \end{bmatrix}$$



$$p(x) \sim N(\mu, \Sigma)$$

$$\Sigma = \begin{bmatrix} 5 \times 5 \end{bmatrix}$$

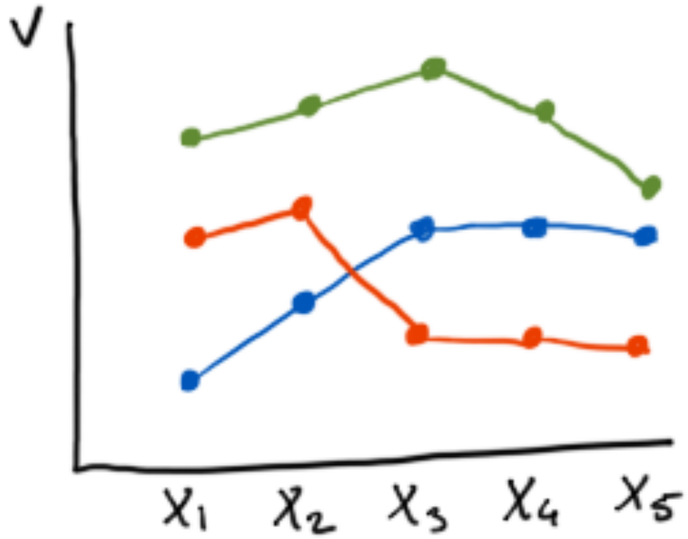
$$D = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 0 & 1 & 2 & 3 \\ 2 & 1 & 0 & 1 & 2 \\ 3 & 2 & 1 & 0 & 1 \\ 4 & 3 & 2 & 1 & 0 \end{bmatrix}$$



Idea: maybe the covariance between points should depend on the distance between them?

$$p(x) \sim N(\mu, \Sigma)$$

$$\Sigma = \begin{bmatrix} 5 \times 5 \end{bmatrix}$$



$$D = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 0 & 1 & 2 & 3 \\ 2 & 1 & 0 & 1 & 2 \\ 3 & 2 & 1 & 0 & 1 \\ 4 & 3 & 2 & 1 & 0 \end{bmatrix}$$

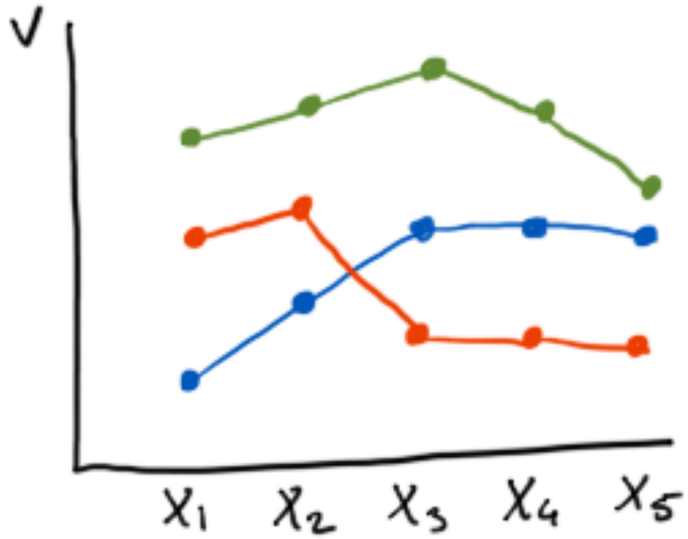
$f(d) =$ 

Gaussian! if dist is small then I'd like covariance!

$$p(x) \sim N(\mu, \Sigma)$$

$$\Sigma = \begin{bmatrix} & f(2) & \\ & \text{etc.} & \\ f(4) & & \end{bmatrix}$$

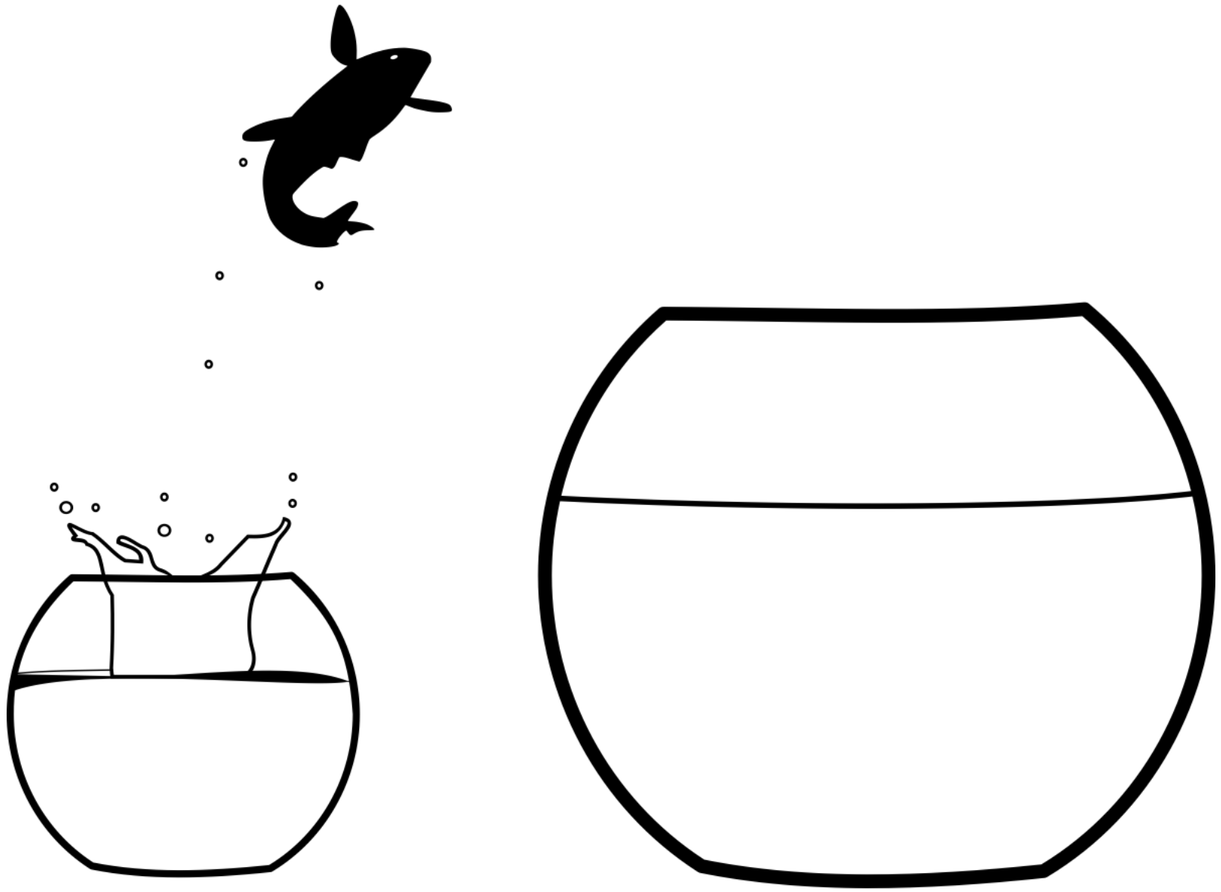
$$\Sigma \uparrow f() \uparrow D = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 0 & 1 & 2 & 3 \\ 2 & 1 & 0 & 1 & 2 \\ 3 & 2 & 1 & 0 & 1 \\ 4 & 3 & 2 & 1 & 0 \end{bmatrix}$$



$f(d) =$ 

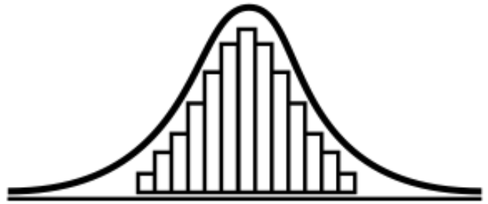
↑ Gaussian! if dist is small then I'd like covariance!

Live Coding Time



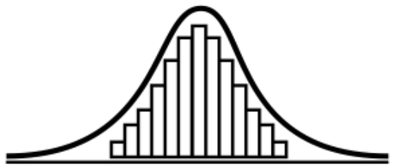
What did I just do?

Hopefully I was able to give some intuition on some things. The gaussian is everywhere in data science.



What did I just do?

It can be mind-blowing to witness the full applicability. Understanding this mathematical lego brick allows you to recognise algorithmic parts.



What did I just do?

If you still feel like you totally don't grasp it: **ThatsNormal[tm]**. I've skipped details in favor of intuition.

The best argument for this approach is considering the alternatives.

How to Demotivate: Wikipedia

Definition [edit]

A time continuous stochastic process $\{X_t; t \in T\}$ is Gaussian if and only if for every finite set of indices $t_1, \ldots, t_k$ in the index set T

$$\mathbf{X}_{t_1,\ldots,t_k} = (X_{t_1}, \ldots, X_{t_k})$$

is a multivariate Gaussian random variable.[2] That is the same as saying every linear combination of $(X_{t_1}, \ldots, X_{t_k})$ has a univariate normal (or Gaussian) distribution.

Using characteristic functions of random variables, the Gaussian property can be formulated as follows: $\{X_t; t \in T\}$ is Gaussian if and only if, for every finite set of indices $t_1, \ldots, t_k$, there are real-valued $\sigma_{\ell j}$, μ_ℓ with $\sigma_{jj} > 0$ such that the following equality holds for all $s_1, s_2, \ldots, s_k \in \mathbb{R}$

$$\mathrm{E}\left(\exp\left(i \sum_{\ell=1}^{k} s_\ell \, \mathbf{X}_{t_\ell}\right)\right) = \exp\left(-\frac{1}{2} \sum_{\ell,j} \sigma_{\ell j} s_\ell s_j + i \sum_{\ell} \mu_\ell s_\ell\right).$$

where i denotes the imaginary unit such that $i^2 = -1$.

The numbers $\sigma_{\ell j}$ and μ_ℓ can be shown to be the covariances and means of the variables in the process.[3]

How to Demotivate: Books

An alternative and equivalent way of reaching identical results to the previous section is possible by considering inference directly in function space. We use a Gaussian process (GP) to describe a distribution over functions. Formally:

Definition 2.1 *A* Gaussian process *is a collection of random variables, any finite number of which have a joint Gaussian distribution.* □

Gaussian process

A Gaussian process is completely specified by its mean function and covariance function. We define mean function $m(\mathbf{x})$ and the covariance function $k(\mathbf{x}, \mathbf{x}')$ of a real process $f(\mathbf{x})$ as

covariance and mean function

$$\begin{aligned} m(\mathbf{x}) &= \mathbb{E}[f(\mathbf{x})], \\ k(\mathbf{x}, \mathbf{x}') &= \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))], \end{aligned} \tag{2.13}$$

and will write the Gaussian process as

$$f(\mathbf{x}) \sim \mathcal{GP}\big(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\big). \tag{2.14}$$

Usually, for notational simplicity we will take the mean function to be zero, although this need not be done, see section 2.7.

How to Demotivate: Paper

Algorithm 1 Infinite-horizon Gaussian process (IHGP) inference. The GP prior is specified in terms of a state space model. After the setup cost on line 2, all operations are at most $\mathcal{O}(m^2)$.

1: **Input:** $\{y_i\}, \{\mathbf{A}, \mathbf{Q}, \mathbf{h}, \mathbf{P}_0\}, p(y \mid f)$ — targets, model, likelihood
2: Set up $\mathbf{P}^{\mathrm{p}}(\gamma)$, $\mathbf{P}^{\mathrm{s}}(\gamma)$, and $\mathbf{G}(\gamma)$ for $\gamma_{1:K}$ — solve DAREs for a set of likelihood variances, cost $\mathcal{O}(K\, m^3)$
3: $\mathbf{m}_0^{\mathrm{f}} \leftarrow \mathbf{0}$; $\quad \mathbf{P}_0^{\mathrm{p}} \leftarrow \mathbf{P}_0$; $\quad \gamma_0 = \infty$ — initialize
4: **for** $i = 1$ **to** n **do**
5: $\quad$ Evaluate $\mathbf{P}_i^{\mathrm{p}} \leftarrow \mathbf{P}^{\mathrm{p}}(\gamma_{i-1})$ — find predictive covariance
6: $\quad \tilde{\mu}_{\mathrm{f},i} \leftarrow \mathbf{h}^{\mathsf{T}} \mathbf{A}\, \mathbf{m}_{i-1}^{\mathrm{f}}$; $\quad \tilde{\sigma}_{\mathrm{f},i}^2 = \mathbf{h}^{\mathsf{T}} \mathbf{P}_i^{\mathrm{p}}\, \mathbf{h}$ — latent
7: $\quad$ **if** Gaussian likelihood **then**
8: $\quad\quad \eta_i \leftarrow y_i$; $\quad \gamma_i \leftarrow \sigma_{\mathrm{n},i}^2$ — if $\sigma_{\mathrm{n},i}^2 := \sigma_{\mathrm{n}}^2$, $\mathbf{k}_i$ and $\mathbf{P}_i^{\mathrm{f}}$ become time-invariant
9: $\quad$ **else**
10: $\quad\quad$ Match $\exp(\nu_i\, f_i - \tau_i\, f_i^2/2)\, \mathrm{N}(f_i \mid \tilde{\mu}_{\mathrm{f},i}, \tilde{\sigma}_{\mathrm{f},i}^2) \overset{\text{mom}}{=} p(y_i \mid f_i)\, \mathrm{N}(f_i \mid \tilde{\mu}_{\mathrm{f},i}, \tilde{\sigma}_{\mathrm{f},i}^2)$ — match moments
11: $\quad\quad \eta_i \leftarrow \nu_i / \tau_i$; $\quad \gamma_i \leftarrow \tau_i^{-1}$ — equivalent update
12: $\quad$ **end if**
13: $\quad \mathbf{k}_i \leftarrow \mathbf{P}_i^{\mathrm{p}}\, \mathbf{h} / (\tilde{\sigma}_{\mathrm{f},i}^2 + \gamma_i)$ — gain
14: $\quad \mathbf{m}_i^{\mathrm{f}} \leftarrow (\mathbf{A} - \mathbf{k}_i\, \mathbf{h}^{\mathsf{T}} \mathbf{A})\, \mathbf{m}_{i-1}^{\mathrm{f}} + \mathbf{k}_i\, \eta_i$; $\quad \mathbf{P}_i^{\mathrm{f}} \leftarrow \mathbf{P}_i^{\mathrm{p}} - \mathbf{k}_i\, \gamma_i\, \mathbf{k}_i^{\mathsf{T}}$ — mean and covariance
15: **end for**
16: $\mathbf{m}_n^{\mathrm{s}} \leftarrow \mathbf{m}_n^{\mathrm{f}}$; $\quad \mathbf{P}_n^{\mathrm{s}} \leftarrow \mathbf{P}^{\mathrm{s}}(\gamma_n)$ — initialize backward pass
17: **for** $i = n - 1$ **to** 1 **do**
18: $\quad \mathbf{m}_i^{\mathrm{s}} \leftarrow \mathbf{m}_i^{\mathrm{f}} + \mathbf{G}(\gamma_i)\, (\mathbf{m}_{i+1}^{\mathrm{s}} - \mathbf{A}\, \mathbf{m}_i^{\mathrm{f}})$; $\quad \mathbf{P}_i^{\mathrm{s}} \leftarrow \mathbf{P}^{\mathrm{s}}(\gamma_i)$ — mean and covariance
19: **end for**
20: **Return:** $\mu_{\mathrm{f},i} = \mathbf{h}^{\mathsf{T}} \mathbf{m}_i^{\mathrm{s}}$, $\sigma_{\mathrm{f},i}^2 = \mathbf{h}^{\mathsf{T}} \mathbf{P}_i^{\mathrm{s}}\, \mathbf{h}$, $\log p(\mathbf{y})$ — mean, variance, evidence

What Might Help

Mathematics is useful when there's context and intuition. It's downright terrible when it's just a bunch of symbols dumped on your retina.

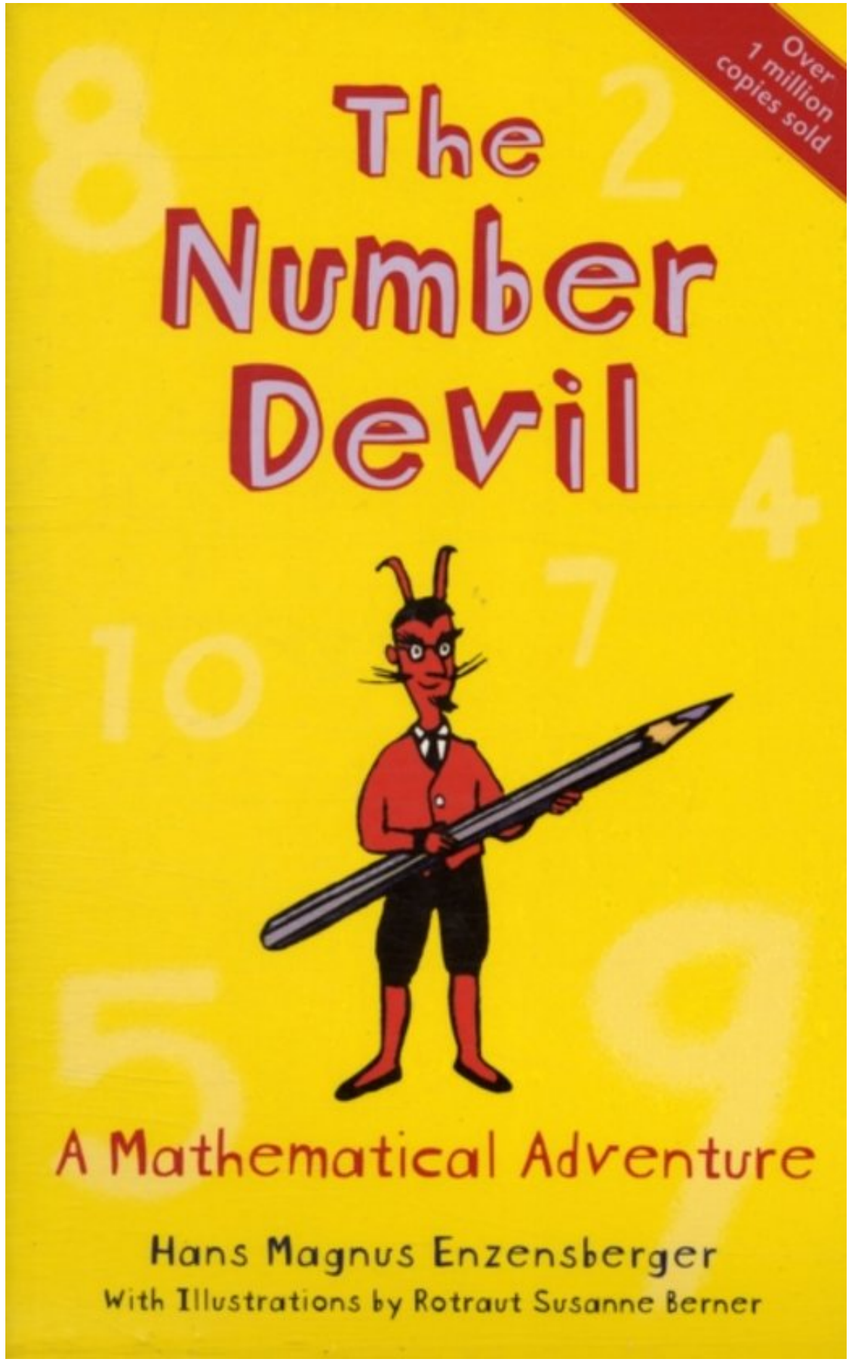
It's even worse if this occurs to you when you're young. I was lucky as a kid because my parents found a way to prevent this intellectual paralysis to me.

My Favourite Book

Remember the story in the beginning of this talk?

This book is about a boy who is afraid of maths. But as luck would have it, the number devil appears in his dreams. It's fairy tales about maths.

All who read this book when they're young ended up getting the best grades in my



Thanks for Listening