Lecture 12

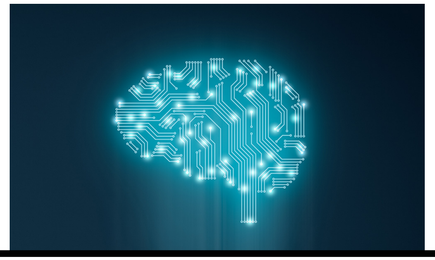
08.12.2025

Today's plan and announcements

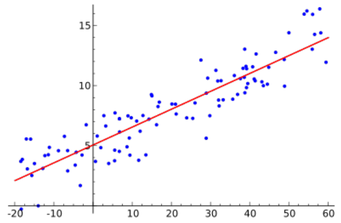
- Unsupervised learning
  - Dimensionality reduction through auto encoders - brief overview
  - Discussion on quiz 2
  - Clustering through k-means

- This week exercise hour
  - Python exercise for k-means
  - Questions about homework

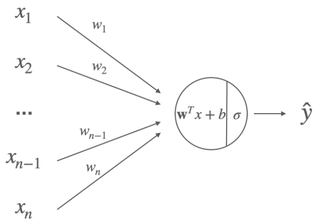
Introduction



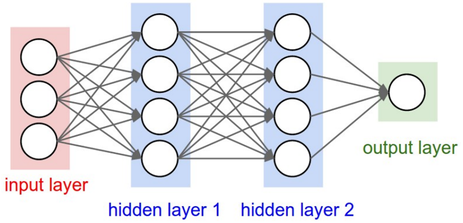
Linear regression



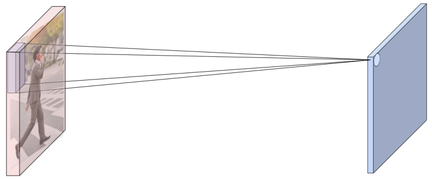
Logistic regression



Neural networks

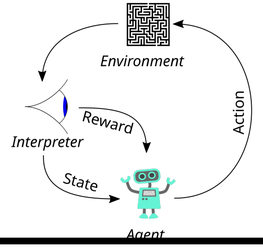


Convolutional neural networks

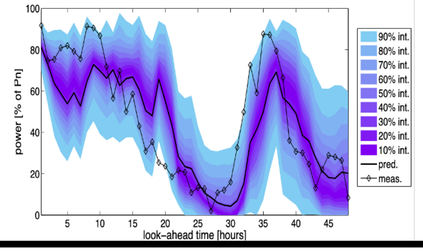


AI & sustainability

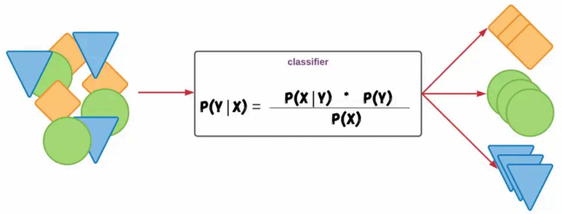
Reinforcement learning



Recurrent neural networks

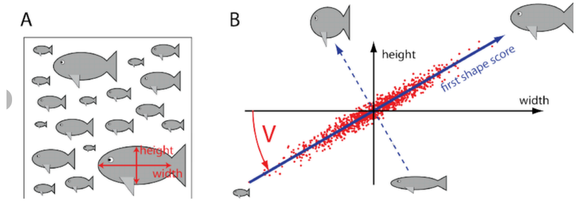


Naive Bayes

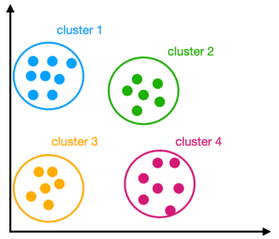


KNN

Dimensionality reduction



Clustering



Review of data statistics through quiz 2 grades

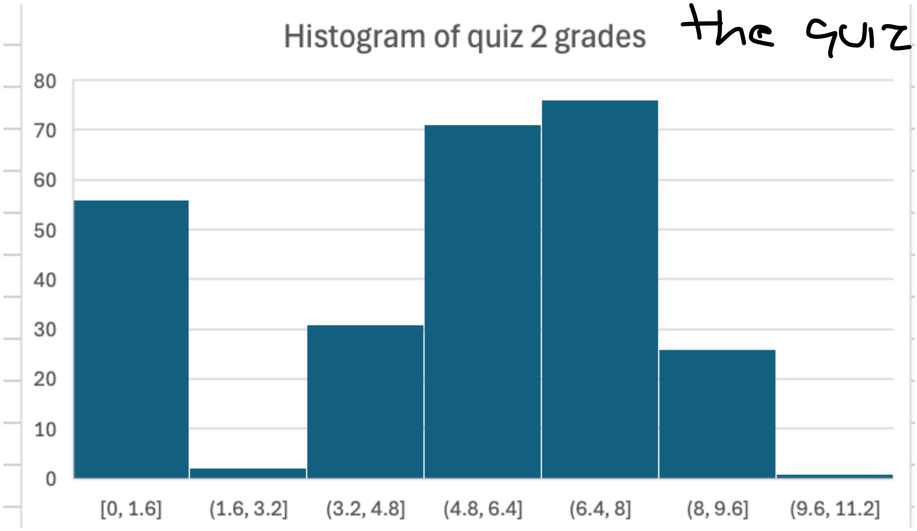
$\{x^i\}_{i=1}^N$ x^i : quiz 2 grade, N : number of students who took the quiz

Mean: 6.38, Standard deviation: 1.56, Mode: 7

Note: we are ignoring the 0 grades (those who didn't take the exam)

Quartiles:

- **1st: 5.50,** 25% of data points is below this number
- **2nd (median): 6.25,** 50% of data points is below this number
- **3rd: 7.50,** 75% of grades is below this number



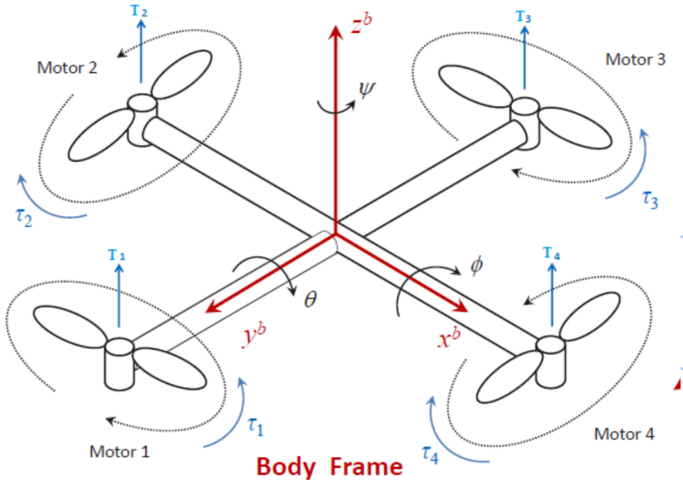
Correlation with quiz 1 grades: 0.294

$\mu = \frac{1}{N}\sum_{i=1}^{N} x^i$, if we consider $x^i \in \mathbb{R}^2$, x_1^i : quiz 1 grades, x_2^i : quiz 2 grades

$Cov(x_1, x_2) = \frac{1}{N}\sum_{i=1}^{N}(x_1^i - \mu_1)(x_2^i - \mu_2)$

$Corr(x_1, x_2) = \frac{cov(x_1, x_2)}{cov(x_1, x_1)\, cov(x_2, x_2)}$, $cov(x_j, x_j) = \sigma_j^2$: variance.

Quiz 2, problem 1



- Quadrotor states

$s \in \mathbb{R}^{12}$

$(x, y, z, \dot{x}, \dot{y}, \dot{z}, \theta, \phi, \varphi, \dot{\theta}, \dot{\phi}, \dot{\varphi})$

pitch, roll, yaw

- Quadrotor inputs

$a \in \mathbb{R}^4$ voltage for each 4 motors

- Policy $\pi : S \to A$. $S = \mathbb{R}^{12}$, $A = \mathbb{R}^4$
  - Linear policy

$\pi : s \mapsto Ks$, $K \in \begin{bmatrix} K_{1,1} & K_{1,2} & \cdots & K_{1,12} \\ K_{2,1} & K_{2,2} & \cdots & K_{2,12} \\ \vdots & & & \\ K_{4,1} & K_{4,2} & \cdots & K_{4,12} \end{bmatrix} \in \mathbb{R}^{4\times 12}$

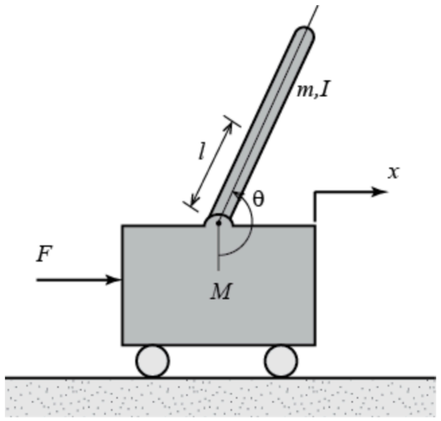
- Objective

$\frac{1}{2}(s_T - s_G)^2$: bring the state close to s_G at time T.

More examples on RL - Problem set 3

states : $(x, \dot{x}, \theta, \dot{\theta}) =: s \in \mathbb{R}^4$

- x: position of cart
- $\dot{x}$: velocity of cart
- θ: pendulum angle
- $\dot{\theta}$: angular velocity



input : force applied to the cart $F \in \mathbb{R}$

$\frac{ds}{dt} = f(s_t, a_t)$ { 1. Euler discretization

2. policy, $\pi : \mathbb{R}^4 \to \mathbb{R}$

- what would a linear policy look like ?

$[k_1 \; k_2 \; k_3 \; k_4] \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix}$. $\pi(s) = ks$, $k \in \mathbb{R}^{1\times 4}$, $s \in \mathbb{R}^4$

Review principal component analysis

- Recall PCA objective $\min_{\Theta \in \mathbb{R}^{d \times r}} \|X - X\Theta\Theta^T\|_F$ with $\Theta = \begin{bmatrix} \theta_1 & \theta_2 & \dots & \theta_r \end{bmatrix} \in \mathbb{R}^{d \times r}$
  - Given normalized data matrix $X \in \mathbb{R}^{N \times d}$, PCA objective: Finding the r-dimensional subspace ($r < d$) spanned by $\{\theta_i\}_{i=1}^r$ that minimizes average distance of data points to it

- Solution through eigen-decomposition of $C = X^T X$

$\{\theta_i^*\}_{i=1}^r$ are the eigenvectors corresponds to r largest eigenvalues of C. $\Theta^* = [\theta_1^*, \dots, \theta_r^*]$

- Reduced dimensional data: $A = X\Theta^* \in \mathbb{R}^{N \times r}$

- Reconstruction of data: $\hat{X} = X\Theta^*\Theta^{*T}$ $\in \mathbb{R}^{N \times d}$

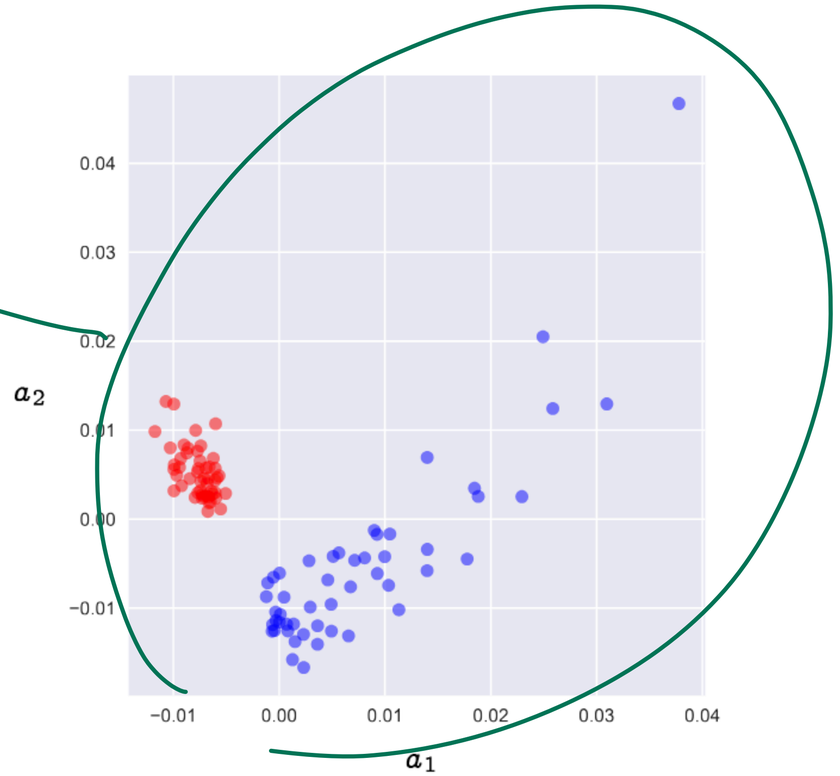
PCA example applied to two texts

PCA to reduce $X \in \mathbb{R}^{100\times 2262}$ to $A \in \mathbb{R}^{100\times 2}$, from Stanford EE104

1) *The* Critique of Pure Reason by Immanuel Kant

2) The Problems of Philosophy by Bertrand Russell

100 data points in reduced dimension 2 (from 2262)



EPFL

Autoencoder for dimensionality reduction

- PCA assumes the data is approximately on a lower dimensional subspace.
- Generally, the data might be approximately on a lower dimensional surface (not a subspace). In this case, other dimensionality reduction techniques such as autoencoder can be used
- An autoencoder is a type of neural network used for dimensionality reduction

encoder $g_\phi : \mathbb{R}^d \to \mathbb{R}^r$, $\{\phi, \theta\}$ are parameters (weights & biases of the 2 neural networks)

decoder $f_\theta : \mathbb{R}^r \to \mathbb{R}^d$,

- Training: minimizing the reconstruction error / loss $\frac{1}{N}\sum_{i=1}^{N} L\left(x^i, \hat{x}^i\right)$

$L(x^i, \hat{x}^i) = \frac{1}{2}(x^i - \hat{x}^i)^2$, $\hat{x}^i = f_\theta(g_\phi(x^i))$

ϕ, θ

Autoencoder

Autoencoder vs. PCA

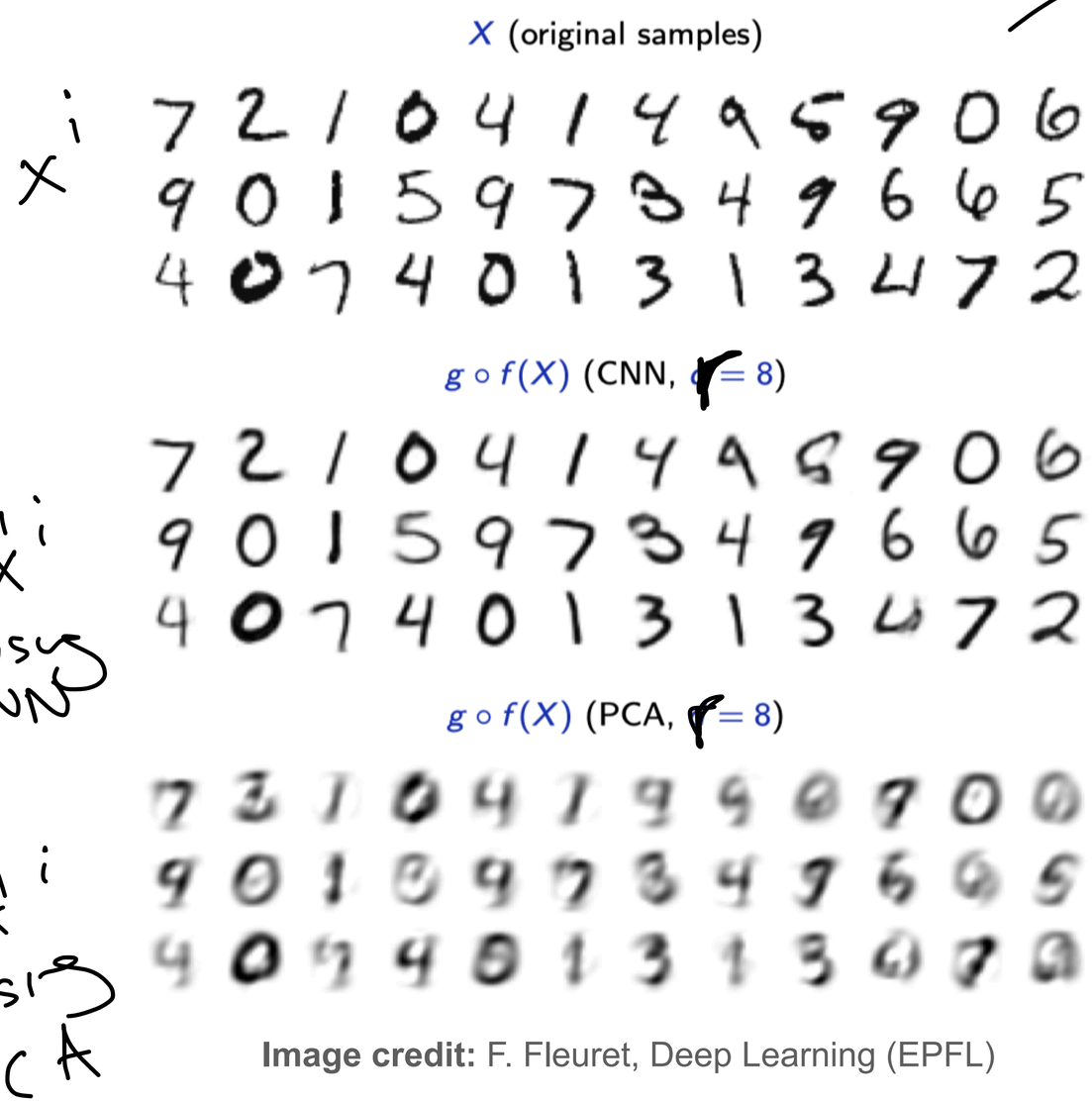


Image credit: F. Fleuret, Deep Learning (EPFL)

Top: Some examples of the original MNIST test samples

Middle: Reconstructed output from an auto-encoder with a latent space of 8 dimensions

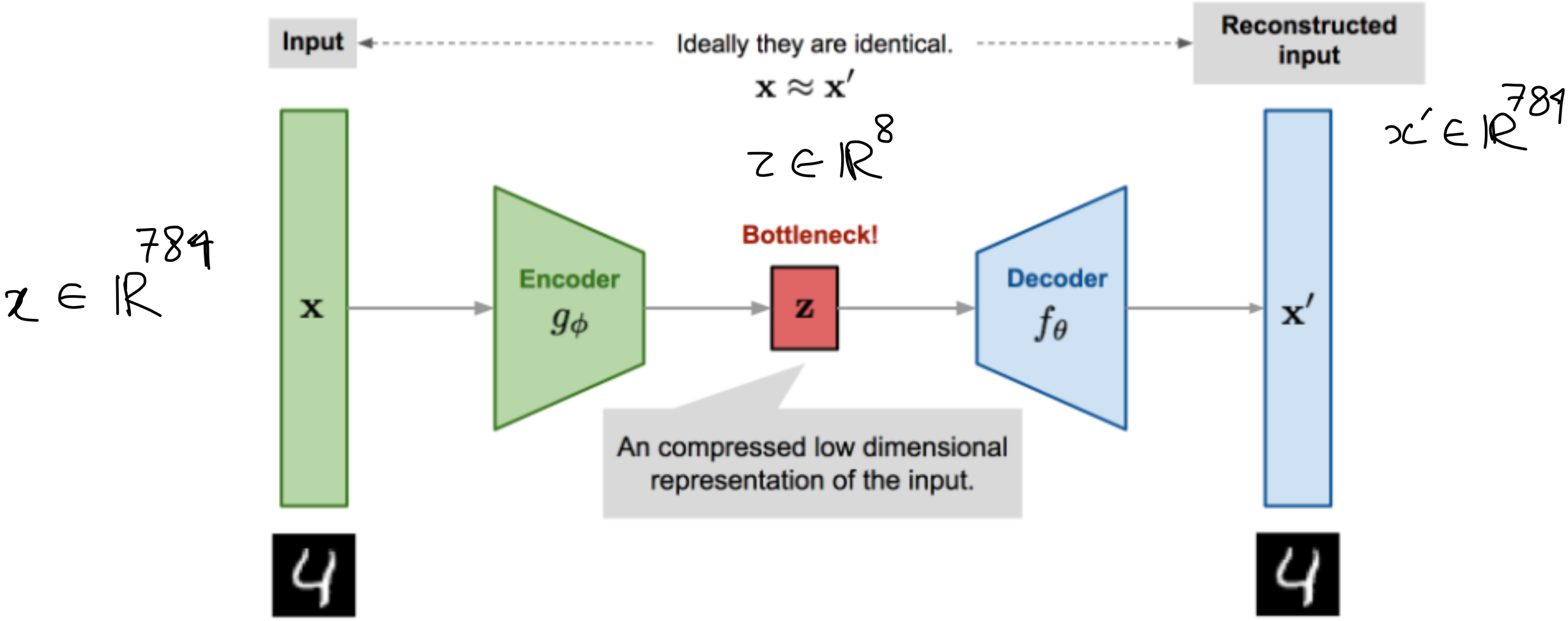
This auto-encoder uses convolutional layers, and was trained on the MNIST training set

Bottom: Reconstructed output from PCA with 8 reduced dimensions

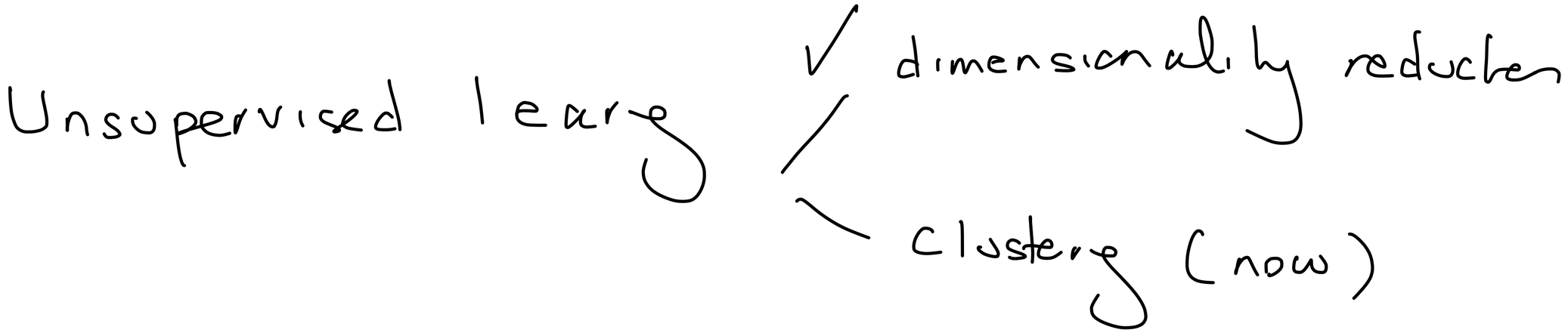
EPFL

Autoencoder

~~Introduction~~



EPFL

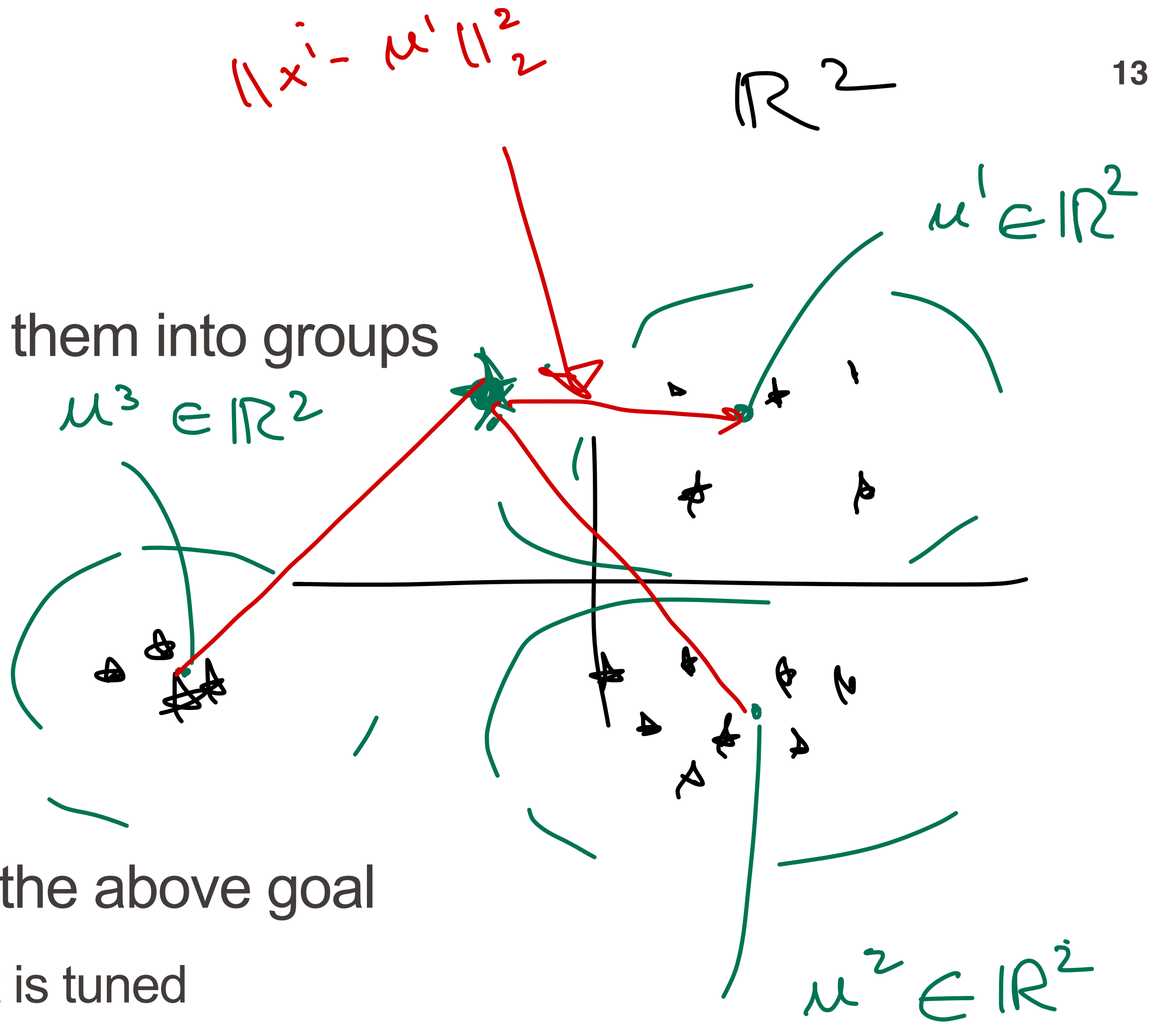


Clustering Through K-means

Clustering - motivation

$\{x^i\}_{i=1}^N$, $x^i \in \mathbb{R}^d$

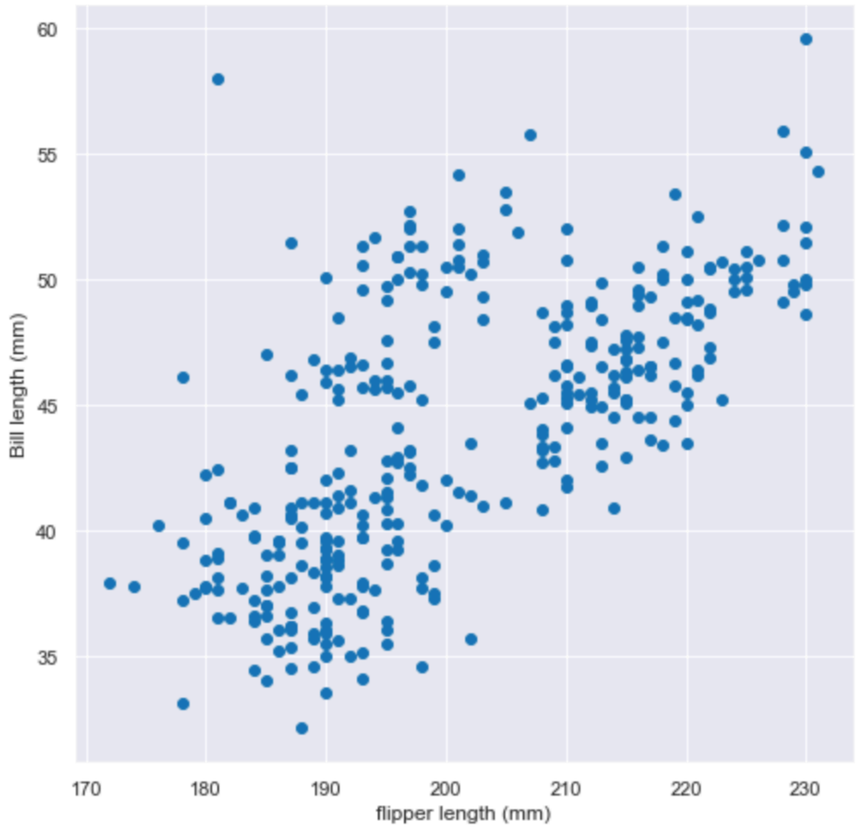
- We have data without labels and want to sort them into groups
  - Cluster news articles or web pages by topic
  - cluster customers based on purchase history
  - cluster galaxies or nearby stars
  - ….



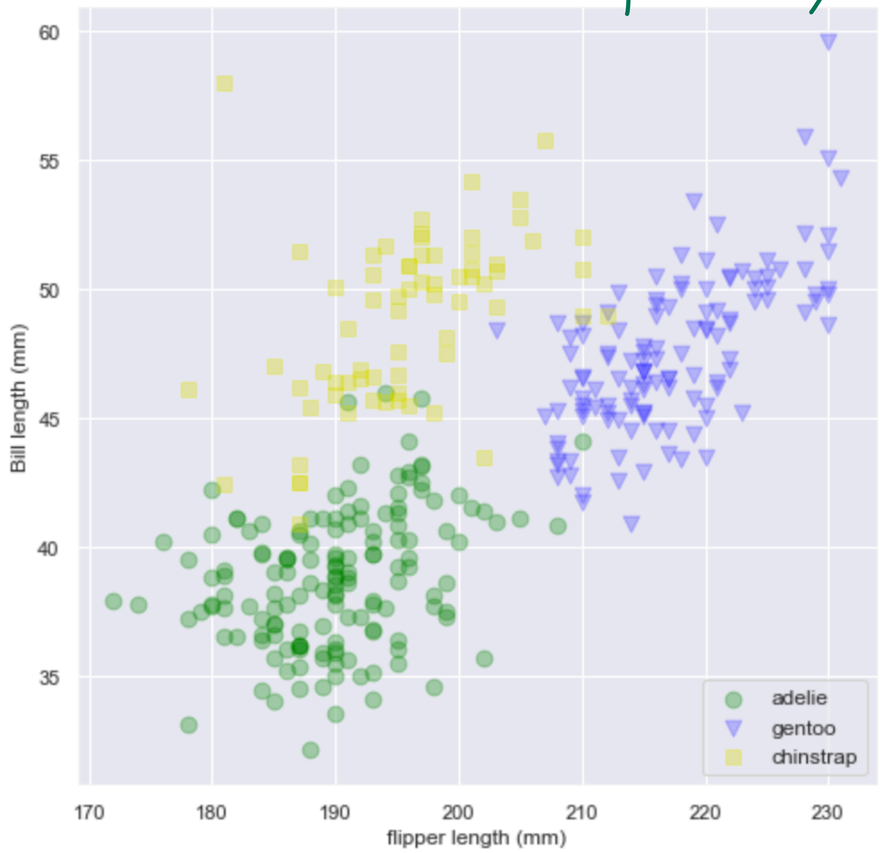
- K-means clustering: an approach to address the above goal
  - K: refers to number of groups we are considering, it is tuned

k-means, example

- Palmer Penguin data set: can we identify potential species from data?



- Can we identify potential species from data (we don't have labels a priori)



K-means approach to clustering

How to cluster data without labels?

Try to find similarity between groups of points

k-means: Group points based on their proximity
(in terms of distance in the feature space)

- Given a set of unlabelled input samples, group the samples into K clusters ($K \in \mathbb{N}$)
- **K-Means idea:**
  - Identify K cluster of data points given N samples. $\{x^i\}_{i=1}^{N}$
  - Find prototype points $\mu_1, \mu_2, \ldots, \mu_K$ representing the center of each cluster and add the other data points to the nearest cluster center.

EPFL

k-means

$\{x^i\}_{i=1}^N$, $x^i \in \mathbb{R}^d$

- A single representative point for data: $\{\mu^\ell\}_{\ell=1}^K \in \mathbb{R}^d$ representative for data

- Example: suppose we want to have one representative point, $\mu \in \mathbb{R}^d$,
  - Use mean-squared loss

error in representing $x^i \in \mathbb{R}^d$, by $\mu \in \mathbb{R}^d$: $L(\mu) = \frac{1}{N} \sum_{i=1}^N \| x^i - \mu \|_2^2$

$\Rightarrow$ $\operatorname{argmin} L(\mu) = \frac{1}{N} \sum_{i=1}^N x^i$ mean of the points.

  - Note: if we use absolute value loss, we will get the median point as the representative

K-Means

- Choose K clusters to represent data $\{\mu^1, \mu^2, \dots, \mu^K\} \subset \mathbb{R}^d$

How do we find $\{\mu^l\}_{l=1}^K$?

- Determining the cluster a single point x^i belongs to $\arg\min_{l=1,2,\dots,K} \|x^i - \mu^l\|_2^2$

$c^i = \arg\min_{l=1,2,\dots,K} \|x^i - \mu^l\|_2^2$: cluster point x^i belongs to

- Determining the cluster centres to minimize the distance of each point to its representative

$$L(\mu^1, \dots, \mu^K) = \frac{1}{N} \sum_{i=1}^{N} \|x^i - \mu^{c^i}\|_2^2 = \frac{1}{N} \sum_{i=1}^{N} \min_{l=1,\dots,K} \|x^i - \mu^l\|_2^2$$

k-means algorithm

- Input: data $\{x^i\}_{i=1}^N \subset \mathbb{R}^d$, Output: cluster means $\{\mu^l\}_{l=1}^K \subset \mathbb{R}^d$
- 1. Initialize cluster means $\mu^1, \ldots, \mu^K \in \mathbb{R}^d$
  - Usually randomly choosing K points from data
- 2. For $i = 1,2,\ldots,N$, assign each point x^i to the closest cluster, indexed by $c^i \in \{1,2,\ldots,K\}$

$$\min_{l=1,\cdots,K} \| x^i - \mu^l \|_2^2 \Rightarrow \text{argm} \text{ is } c^i$$

- 3. Let $C^k = \{x^i, | \; c^i = k\}$ be data points in cluster k. Re-compute the means $\mu^k = \text{mean}(C^k)$

$$\mu^k = \frac{1}{N} \sum_{i \in C^k} x^i, \quad k = 1, 2, \ldots K.$$

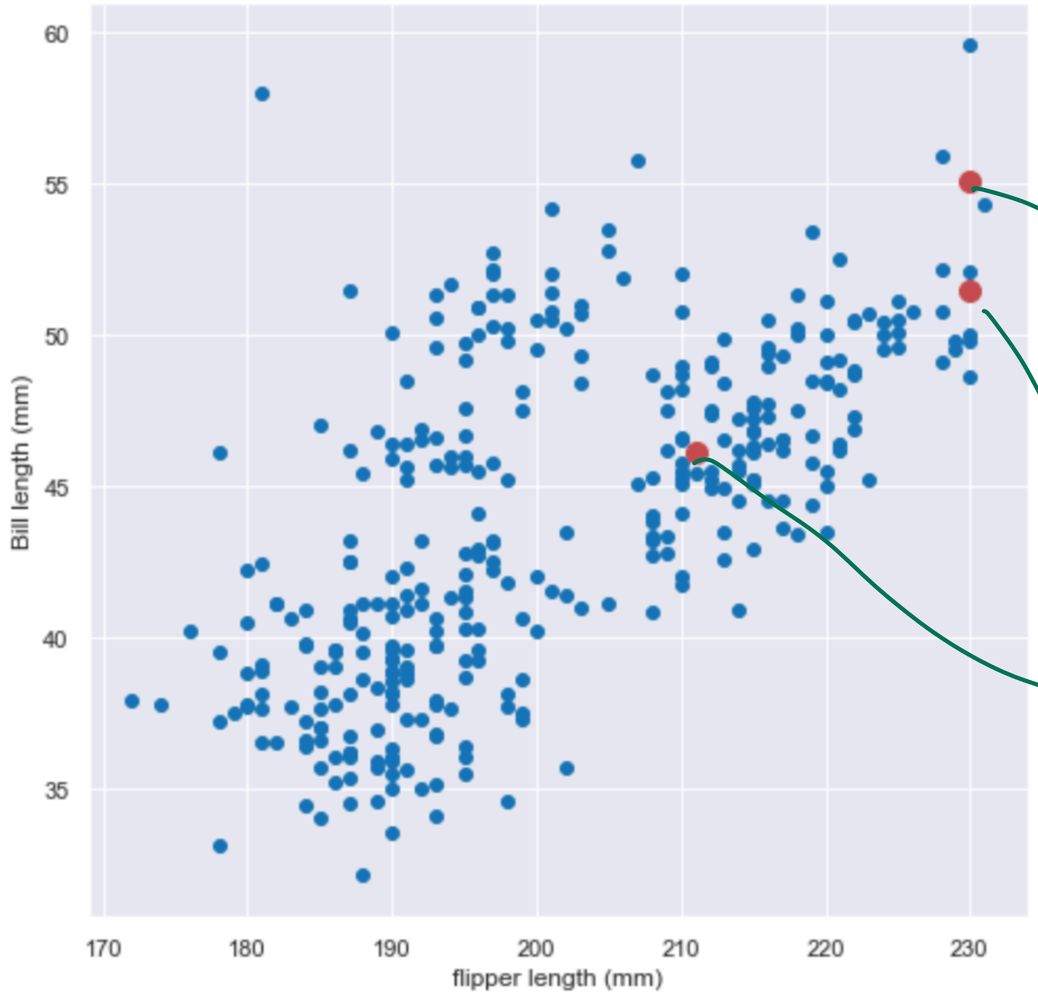
- Go to step 2 if not converged

k-means example

Goal : cluster this dataset into $3 = K$ clusters.

- Mean initialization μ^1, μ^2, μ^3 chosen randomly from $\{x^i\}_{i=1}^N$

$x^i \in \mathbb{R}^2$



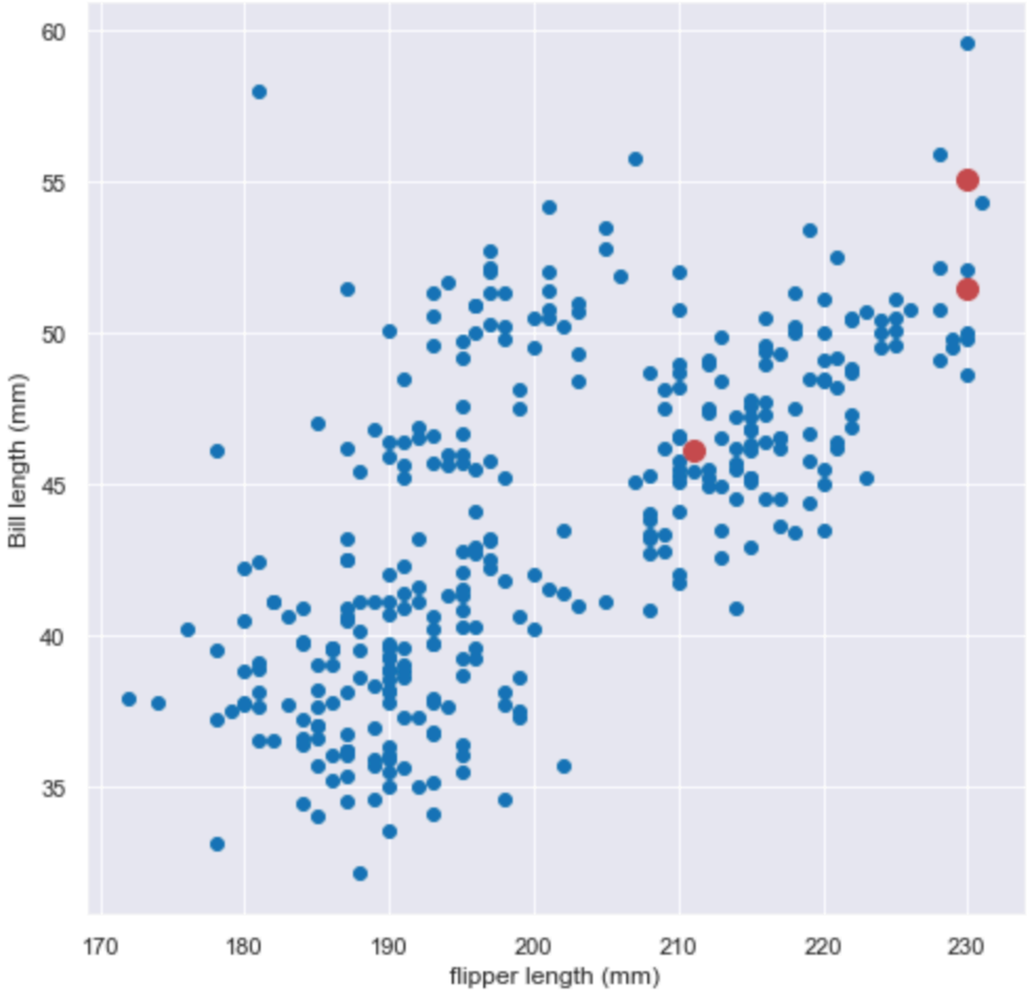
$\mu^1 \in \mathbb{R}^2$

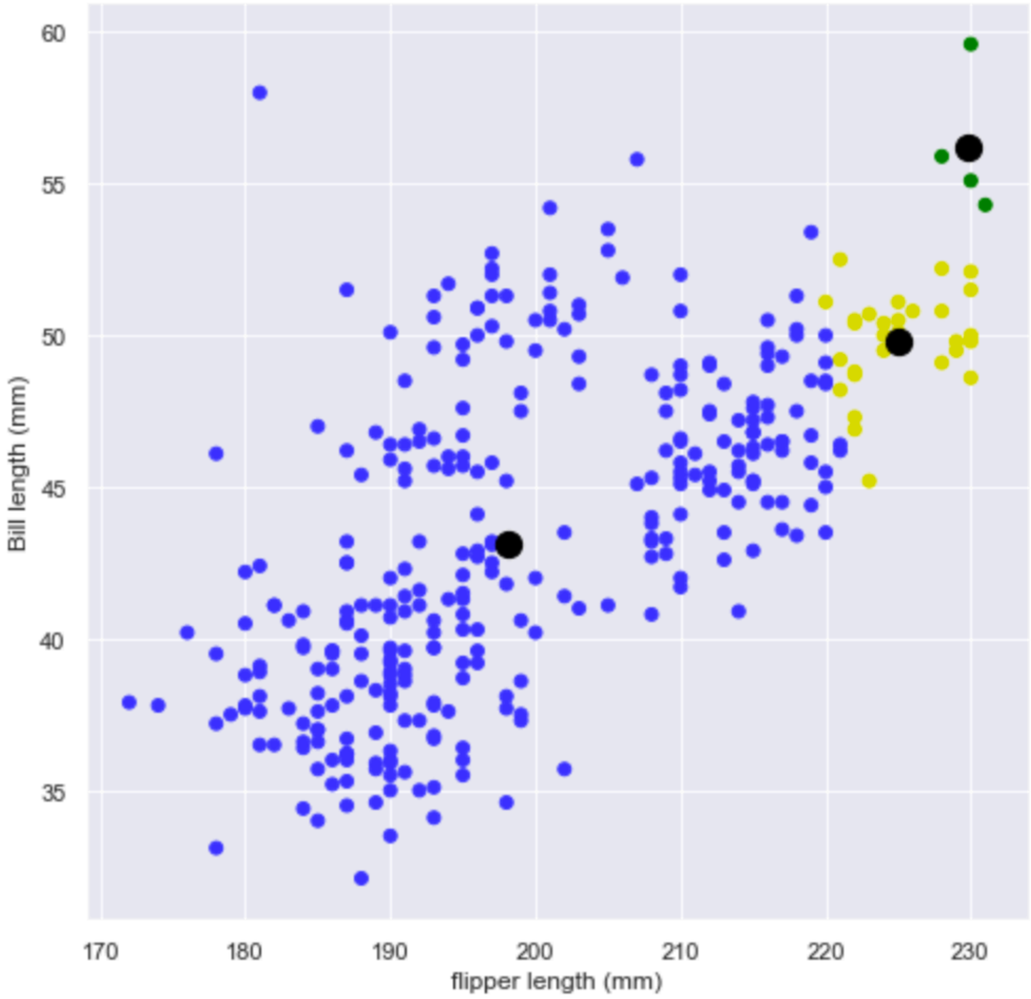
$\mu^2 \in \mathbb{R}^2$

$\mu^3 \in \mathbb{R}^2$

k-Means

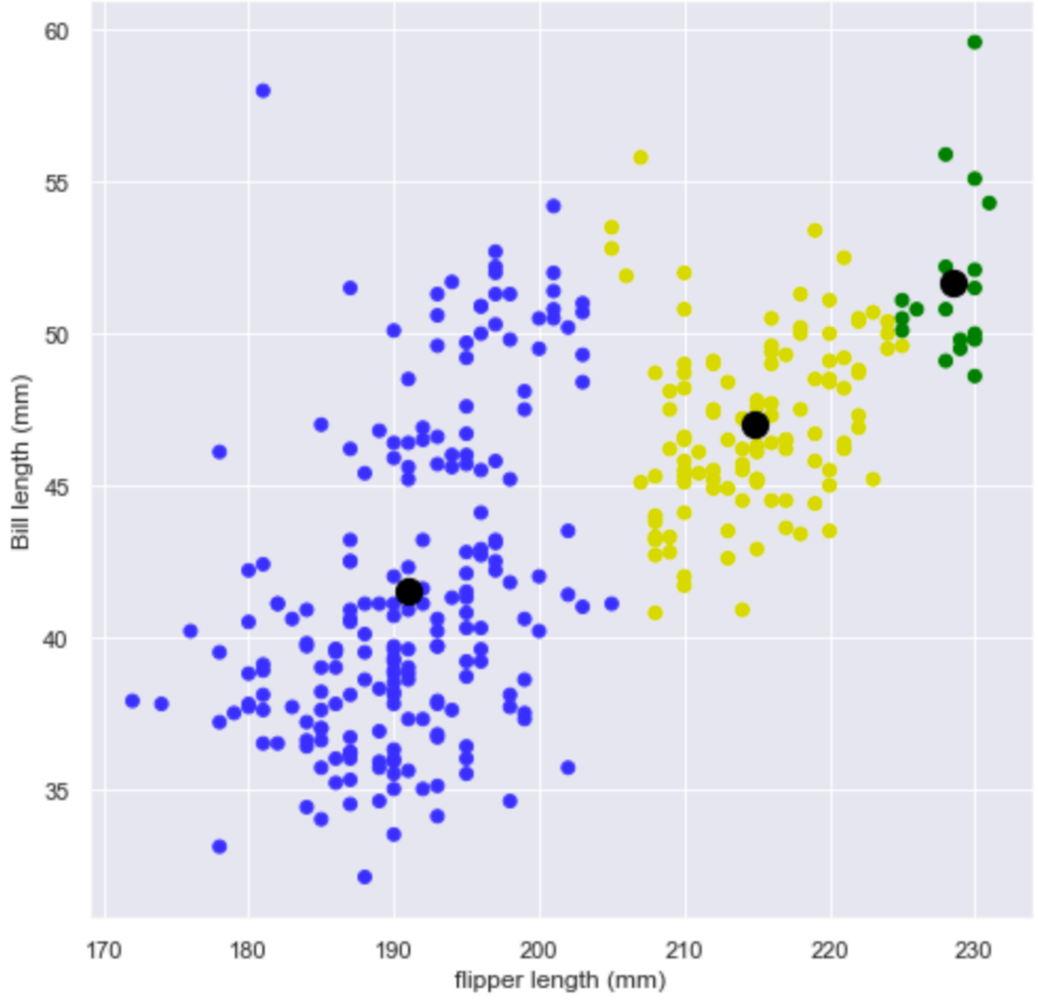
- Iteration 1, First assignment (step 1) and updating the means (step 2)





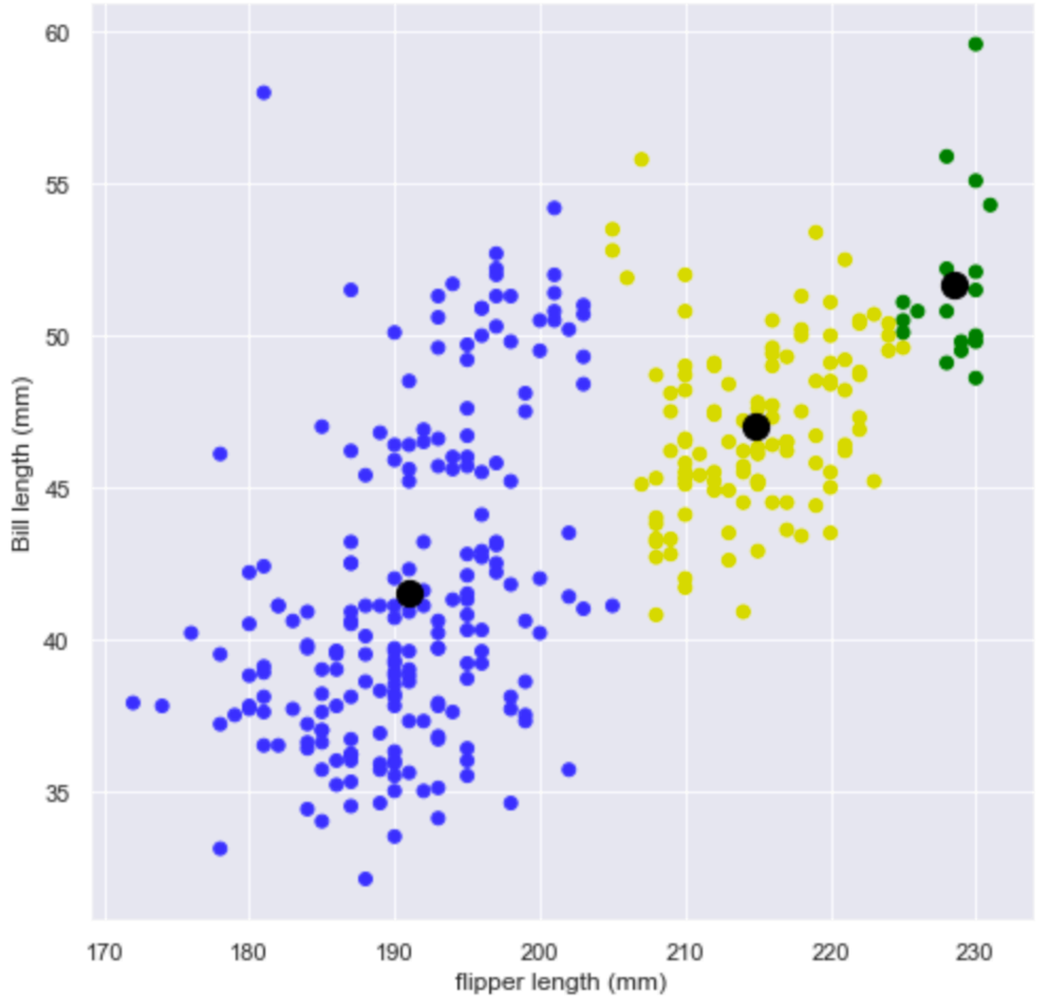
k-Means

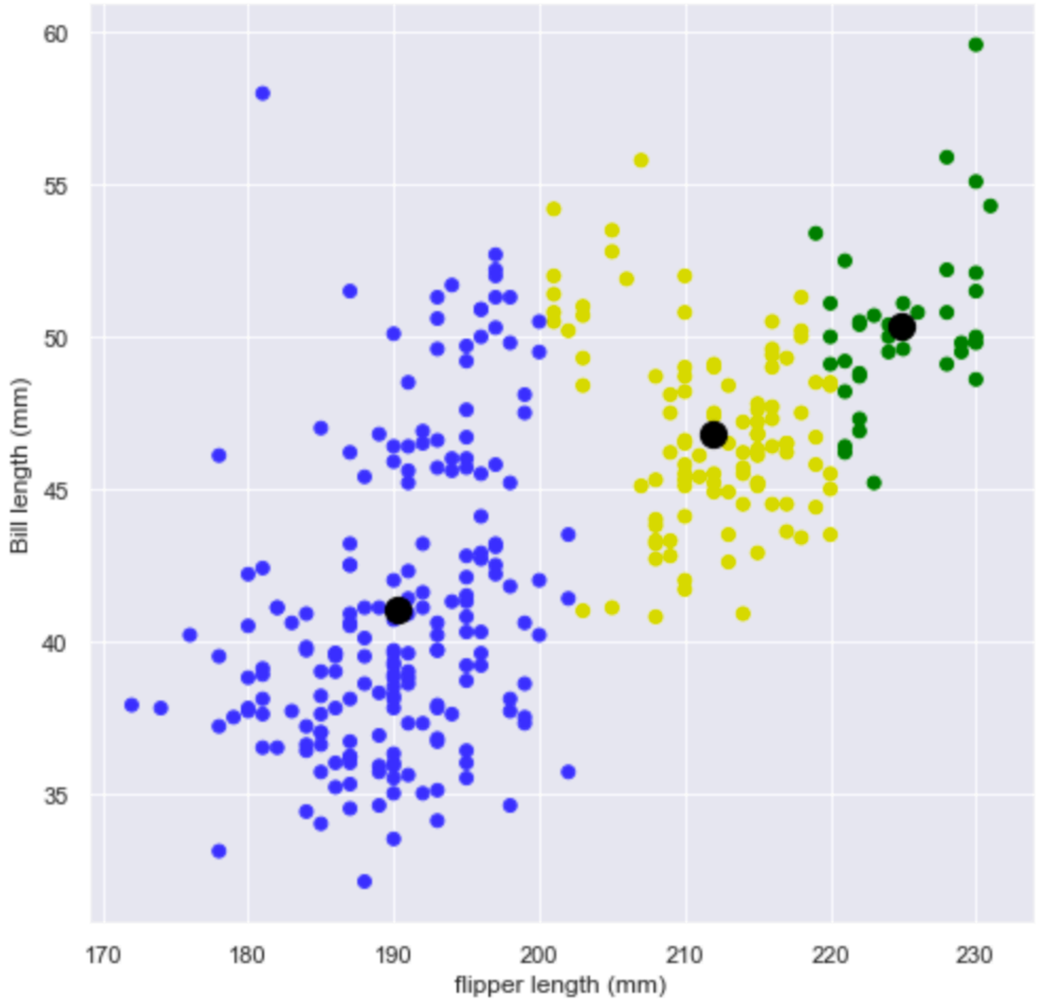
- Iteration 2, Second assignment (step 1) and updating the means (step 2)



k-Means

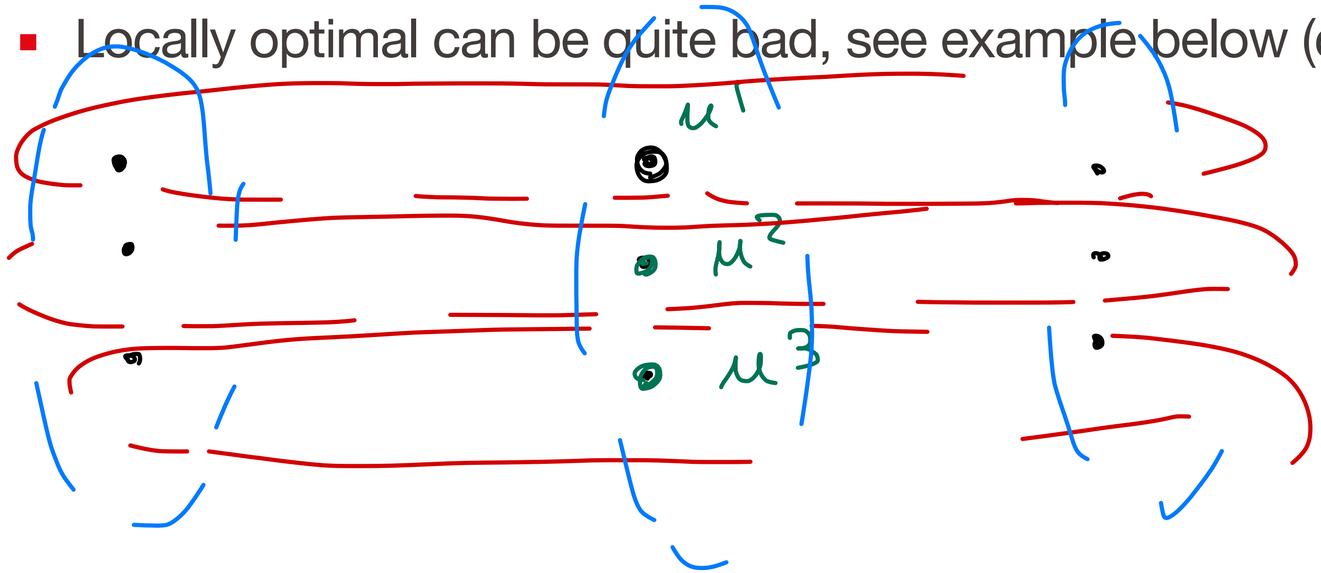
- Iteration 3, third assignment (step 1) and updating the means (step 2)
- Final assignment





k-means algorithm convergence

- k-means objective is non-convex, but it can be shown that the k-means algorithm converges to a locally optimal solution
- The particular local optimum depends on the initial condition
- Locally optimal can be quite bad, see example below (courtesy of Matt Gormley)



k-means initialized on $\mu^1, \mu^2, \mu^3 \in \mathbb{R}^2$ gives horizontal clusters but the optimal cluster is shown in blue

k-means clustering approach limitations

- k-means decision boundaries are linear

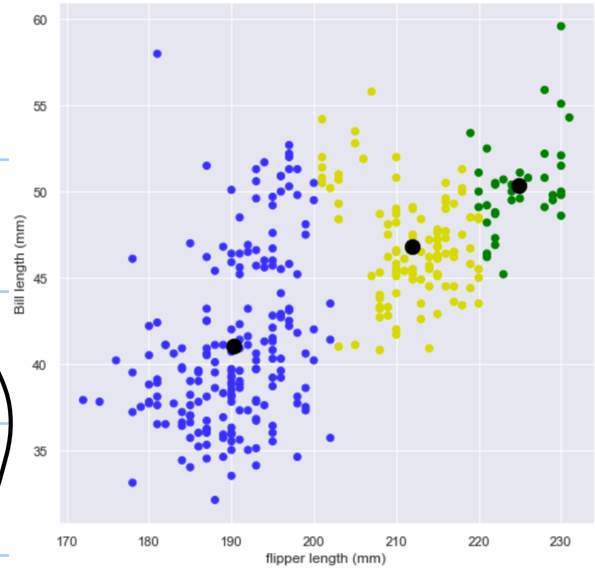
$x^i \in \mathbb{R}^d$ belongs to cluster $l \in \{1, 2, \dots, K\}$

$$\| x^i - \mu^l \|_2^2 \leq \| x^i - \mu^k \|_2^2, \quad \forall k \neq l$$

$$(x^i)^T x^i - 2(\mu^l)^T x^i + (\mu^l)^T \mu^l - \left((x^i)^T x^i - 2(\mu^k)^T x^i + (\mu^k)^T \mu^k \right)$$

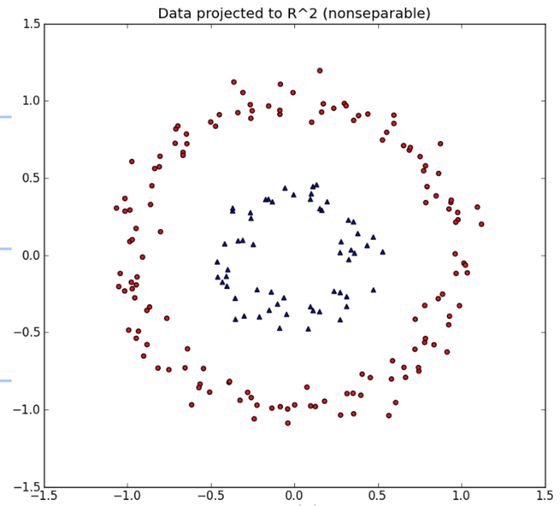
$$= \underbrace{-2(\mu^l - \mu^k)^T}_{w^T} x^i + \underbrace{(\mu^l)^T \mu^l - (\mu^k)^T \mu^k}_{b \in \mathbb{R}} \leq 0$$

$w^T x^i + b$



- What if we have nonlinear boundaries

K-mean clustering with 2 clusters would perform very poorly because the data is not linearly separable



Kernelized k-means algorithm

- Consider a nonlinear transformation of data using feature $\Phi : \mathbb{R}^d \to \mathbb{R}^p$
  - example: $x \in \mathbb{R}^2$, $\phi(x) = (1, x_1, x_2, x_1 x_2, x_1^2, x_2^2)$, $\mathbb{R}^2 \to \mathbb{R}^6$
- K-means algorithm needs only the kernel values $K_{ij} = < \Phi(x_i), \Phi(x_j) >$ for $i, j \in \{1, \ldots, N\}$.

Let $\{\mu^l\}_{l=1}^K \subset \mathbb{R}^p$ be cluster centers in the transformed space. Then,

$$\|\Phi(x^i) - \mu^l\|_2^2 = \underbrace{\phi(x^i)^T \phi(x^i)}_{K_{ii}} - \underbrace{2 (\mu^l)^T \phi(x^i)}_{\frac{1}{n_l} \sum_{j \in C} K_{ij}} + \underbrace{(\mu^l)^T (\mu^l)}_{\left(\frac{1}{n_l}\right)^2 \sum_{j \in C} \sum_{k \in C} K_{jk}}$$

$$\mu^l = \frac{1}{n_l} \sum_{j \in C} \phi(x^j), \quad \text{where } C \text{ is index of points for cluster } l$$

Note that we never need to know the mapping Φ, rather we need to know $K_{ij} = < \Phi(x_i), \Phi(x_j) >$. Thus, we can equivalently choose a so-called kernel K

Kernelized k-means algorithm

- A kernel function defines an inner product similarity between points $K(x^i, x^j) = \Phi(x^i)^T \Phi(x^j), \Phi : \mathbb{R}^d \to \mathbb{R}^p$
- Popular examples of kernels
  - Linear: $K(x^i, x^j) = (x^i)^T x^j$, $\Phi(x) = x$, namely, the transformation is the identity
  - Quadratic: $K(x^i, x^j) = (1 + (x^i)^T x^j)^2$, $\Phi(x) = (1, \sqrt{2}x_1, \ldots, \sqrt{2}x_1 x_2, \ldots, \ldots, x_d^2), p = \begin{pmatrix} d+2 \\ 2 \end{pmatrix}$
  - Polynomial: $k(x^i, x^j) = (1 + (x^i)^T x^j)^m$, $\Phi : \mathbb{R}^d \to \mathbb{R}^p, p = \begin{pmatrix} d+m \\ m \end{pmatrix}$ (number of monomials of degree m in d variables)
  - Radial basis function (Gaussian) $k(x^i, x^j) = \exp(-\gamma \|x^i - x^j\|_2^2), \gamma > 0, p = \infty$
- Many other machine learning approaches we looked at can also be addressed through implicitly working in a higher-dimensional feature space (via a kernel function)

Summary

- Over the past two lectures we discussed unsupervised learning
  1. Dimensionality reduction through PCA and brief introduction to auto encoders
  2. Clustering through k-means, brief peak kernelized k-means

- There are other unsupervised learning problems that we didn't look at
  - density estimation, generative modeling, ….

- Your tasks this week
  - Python code for k-means, it builds upon PCA python
  - Questions on python homework → December 17 due.