# Introduction to Machine Learning: Exploratory Data Analysis With Generative Models

Farzad Kamalabadi

University of Illinois Urbana-Champaign

Heliophysics Summer School, 2026

## Outline of this lecture

- PCA
- k-means
- Gaussian Mixture Models
- Expectation Maximization

## Reading Material

- K. Murphy; Machine Learning: A Probabilistic Perspective; Chapter 12

**Recap:** Supervised learning

Given a dataset $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}$ of data-label pairs, construct a mapping $F(w, x, y)$.

Examples:

- Linear regression
- Logistic regression
- Support vector machine
- Deep net

What if we don't have labels?

Without labels, we can still find structure in unlabeled data

$$\mathcal{D} = \{(x^{(i)})\}$$

Goal of unsupervised learning?
Less clear but generally:

- Recover hidden structure
- Data compression or dimensionality reduction
- Explore or explain data (generate data)
- Construct features for supervised learning

Why modeling a distribution $p(x)$?

- Synthesis of objects (images, text)
- Environment simulator (reinforcement learning, planning)
- Leveraging unlabeled data

Methods:

- Variational Auto-encoders
- Generative Adversarial Nets
- Score-based generative models (Diffusion models)
- PCA
- k-means
- Gaussian Mixture Models

**Recap:** Maximum likelihood so far?

Model:

$$p(\boldsymbol{y}|x) = \frac{\exp F(\boldsymbol{y}, x, \boldsymbol{w})/\epsilon}{\sum_{\hat{\boldsymbol{y}}} \exp F(\hat{\boldsymbol{y}}, x, \boldsymbol{w})/\epsilon}$$

- $\boldsymbol{y}$: discrete output space
- $x$: input data

**Now:**

How about modeling a distribution $p(x)$ for the data?

Given data points $x$, how can we model $p(x)$?

- Maximum likelihood approach:

$$\theta^* = \max_{\theta} \sum_i \log p(x^{(i)}; \theta)$$

  - Fit mean and variance (= parameters $\theta$) of a distribution (e.g., Gaussian)
  - Fit parameters $\theta$ of a mixture distribution (e.g., mixture of Gaussian, k-means)
  - Use a variational auto-encoder

Recall: Linear regression (discriminative)

$$p(y^{(i)}|x^{(i)}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(y^{(i)} - \boldsymbol{w}^\top \phi(x^{(i)}))^2\right)$$

Now: (generative)

$$p(x^{(i)}|\underbrace{\mu, \sigma}_{\theta \text{ or } \boldsymbol{w}}) = \mathcal{N}(x^{(i)}|\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x^{(i)} - \mu)^2\right)$$

Important difference: we are now interested in modeling the distribution of the data $x^{(i)}$ and not the class labels $y^{(i)}$. Though it is sometimes ambiguous what you call data or labels.

**More broadly:**

Generative:

$$\ln p_\theta(x^{(i)}) = \ln \sum_{\boldsymbol{z}} p_\theta(x^{(i)}, \boldsymbol{z})$$

Discriminative:

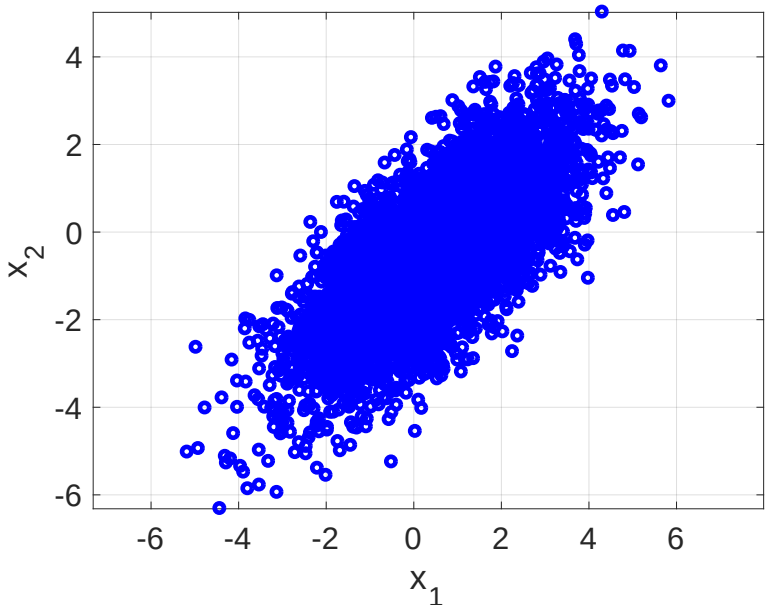
$$\ln p_{\boldsymbol{w}}(\boldsymbol{y}|x^{(i)})$$

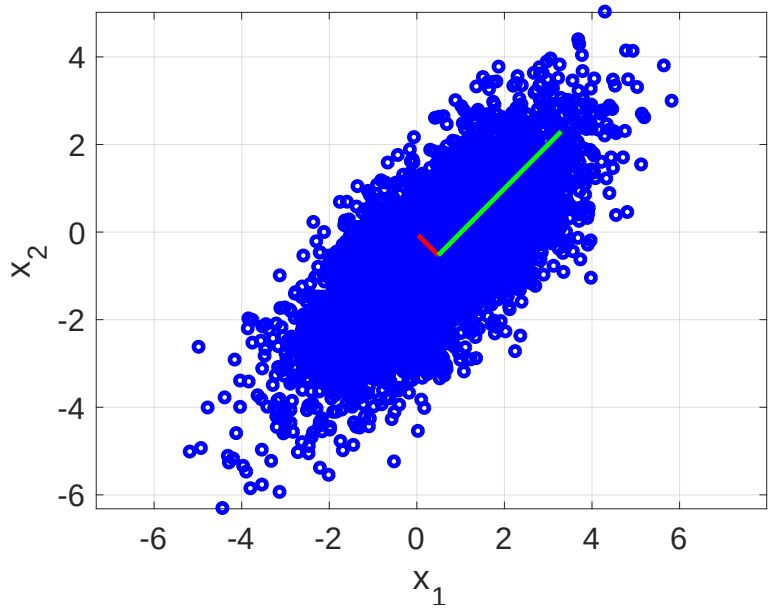
Observations:

- $\boldsymbol{z}$ latent variable; never observed
- $\boldsymbol{y}$ always completely observed

## PCA

Example:



## PCA

Goal: find that lower dimensional **linear** subspace in which the projected data has highest variance

$$\text{(data)} \quad X = \begin{bmatrix} | & & | \\ x^{(1)} & \dots & x^{(|\mathcal{D}|)} \\ | & & | \end{bmatrix}$$

$$\text{(centered data)} \quad \bar{X} = \begin{bmatrix} | & & | \\ x^{(1)} - \mu & \dots & x^{(|\mathcal{D}|)} - \mu \\ | & & | \end{bmatrix} \quad \text{where } \mu = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} x$$

Never forget to center data! Symmetric matrix $\Sigma = \frac{1}{|\mathcal{D}|} \bar{X} \bar{X}^T$

$$\max_{w : \|w\|_2^2 = 1} \text{Var}(w^T \bar{X}) = \max_{w : \|w\|_2^2 = 1} \mathbb{E}[w^T \bar{X} \bar{X}^T w] = \max_{w : \|w\|_2^2 = 1} w^T \Sigma w$$

How to solve

$$\max_{w:\|w\|_2^2=1} \quad w^T \Sigma w$$

Lagrangian:

$$L() = w^T \Sigma w - \lambda(w^T w - 1)$$

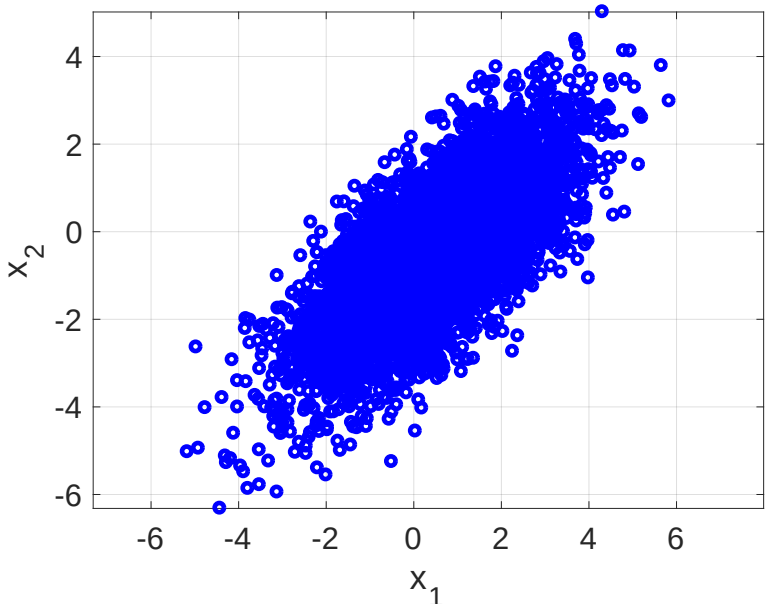
Derivative of $L$ w.r.t. $w$ set to zero:

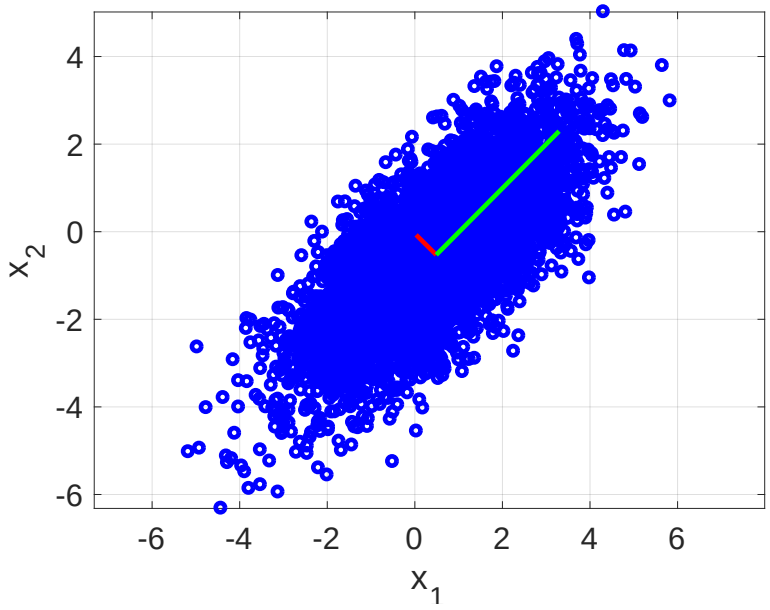
$$\Sigma w = \lambda w \qquad \text{(eigenvalue problem)}$$

Which eigenvector/eigenvalue should we take?

We want to maximize $w^T \Sigma w = \lambda w^T w = \lambda$. Hence, $w$ is the eigenvector corresponding to the largest eigenvalue.

Example:




What if we want to find the direction with second, third largest variance that's orthogonal to the first, first and second?

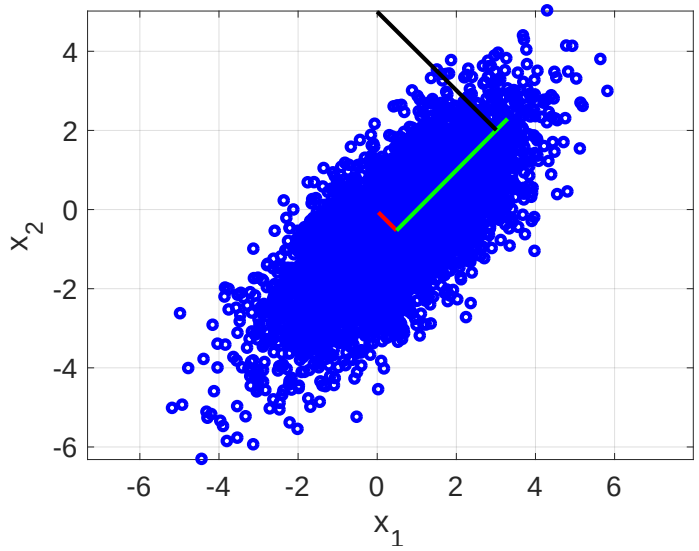
Finding that $d$-dimensional subspace that captures the largest variance?

$$\max_{w_1,\ldots,w_d : w_i^T w_j = \delta_{ij}} \sum_{i=1}^{d} w_i^T \Sigma w_i$$

Algorithm:

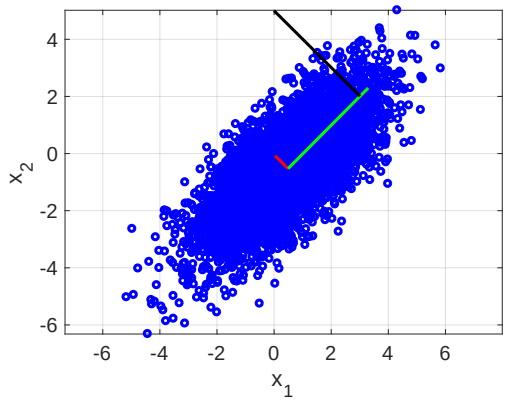
- Work sequentially one vector at a time
- Compute a matrix eigenvalue decomposition

# How to project data into low-dimensional space?



1. Collect all subspace directions:

$$U = \begin{bmatrix} | & & | \\ w_1 & \cdots & w_d \\ | & & | \end{bmatrix}$$




2. Project points into subspace (compressed space)

$$\hat{x} = U^T(x - \mu)$$

3. Approximately reconstructed data

$$\tilde{x} = U\hat{x} + \mu$$

**Alternative view of PCA:**
PCA finds the axis which minimizes the sum of squared distances from points to their orthogonal projections on that axis (we assume $\mu = 0$ for notational convenience):

$$\min_{w:\|w\|_2^2=1} \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \|x^{(i)} - ww^T x^{(i)}\|_2^2 \qquad \text{(see previous slide \& lin reg)}$$

Frobenius norm:

$$\|A\|_F^2 = \sum_{i,j} a_{i,j}^2 = \text{Tr}(A^T A)$$

Rewriting the objective:

$$\frac{1}{|\mathcal{D}|}\sum_{i=1}^{|\mathcal{D}|}\|x^{(i)}-ww^Tx^{(i)}\|_2^2=\frac{1}{|\mathcal{D}|}\|\bar{X}-ww^T\bar{X}\|_F^2$$

$$=\frac{1}{|\mathcal{D}|}\mathrm{Tr}((P\bar{X})^T(P\bar{X})) \quad \text{where} \quad P=I-ww^T$$

$$=\frac{1}{|\mathcal{D}|}\mathrm{Tr}(\bar{X}\bar{X}^TP^TP)$$

$$= \quad \mathrm{Tr}(\Sigma P) \qquad \text{since for projection } \; P^TP=P$$

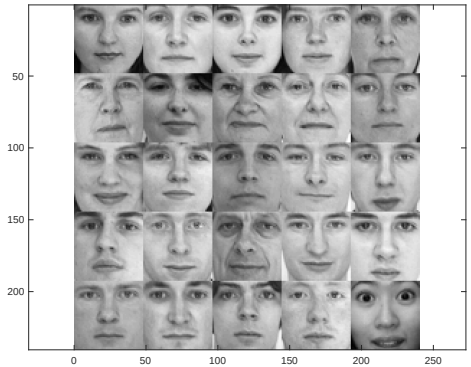
Hence:

$$\arg\min_{w:\|w\|_2^2=1}\frac{1}{|\mathcal{D}|}\sum_{i=1}^{|\mathcal{D}|}\|x^{(i)}-ww^Tx^{(i)}\|_2^2 := \mathrm{Tr}(\Sigma)-\mathrm{Tr}(\Sigma ww^T)$$

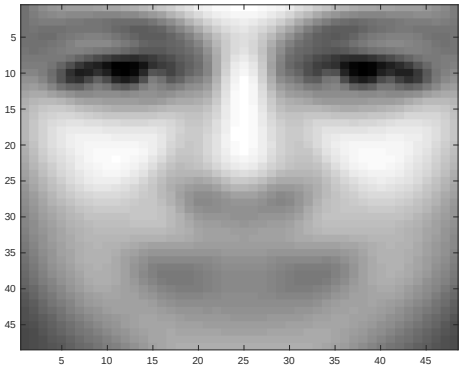
$$= \quad \arg\max_{w:\|w\|_2^2=1} w^T\Sigma w$$

# Applications

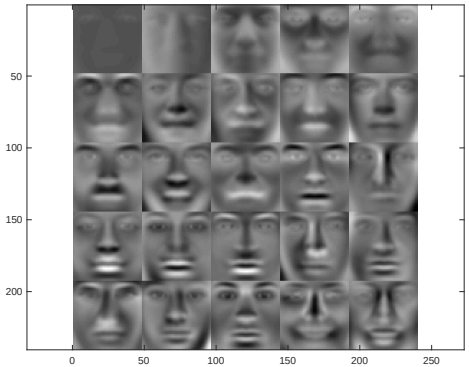
Compressing high-dimensional data



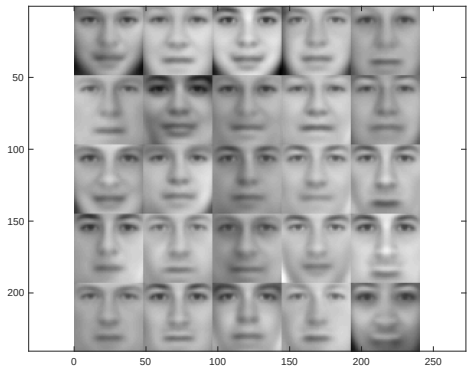

Original $x$




Mean $\mu$




25 eigenvectors $U$




Reconstruction $\tilde{x}$

## Singular Value Decomposition to compute PCA

Currently we compute the eigenvalues of $\Sigma = \frac{1}{|\mathcal{D}|}\bar{X}\bar{X}^T$
Instead of first computing the outer product and then computing its eigenvalues we can use the singular value decomposition.

How?

Given the singular value decomposition

$$\frac{1}{\sqrt{|\mathcal{D}|}}\bar{X} = USV^T$$

We obtain

$$\Sigma = USV^T VSU^T$$

The left singular vectors $U$ of $\frac{1}{\sqrt{|\mathcal{D}|}}\bar{X}$ are needed

# k-Means Clustering

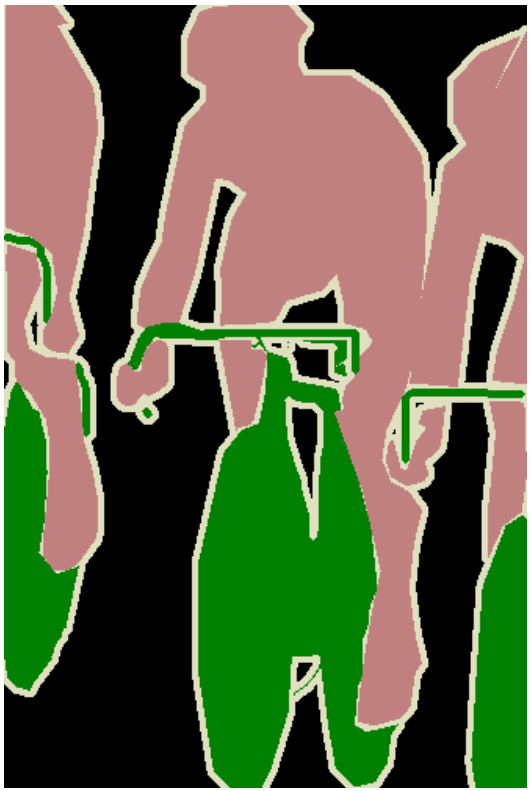
**Goals of this lecture**

- Understanding clustering concept
- Getting to know k-Means

**Reading material:**

- K. Murphy; Machine Learning: A Probabilistic Perspective; Chapter 11

# Recap: Semantic Image Segmentation

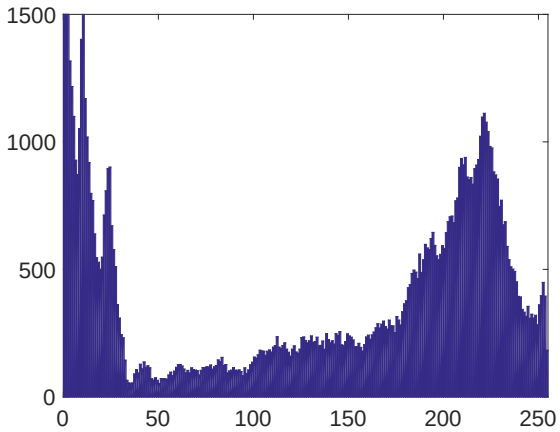



How did we do it?

What if we don't have labels?

Image

Intensities




Clustering

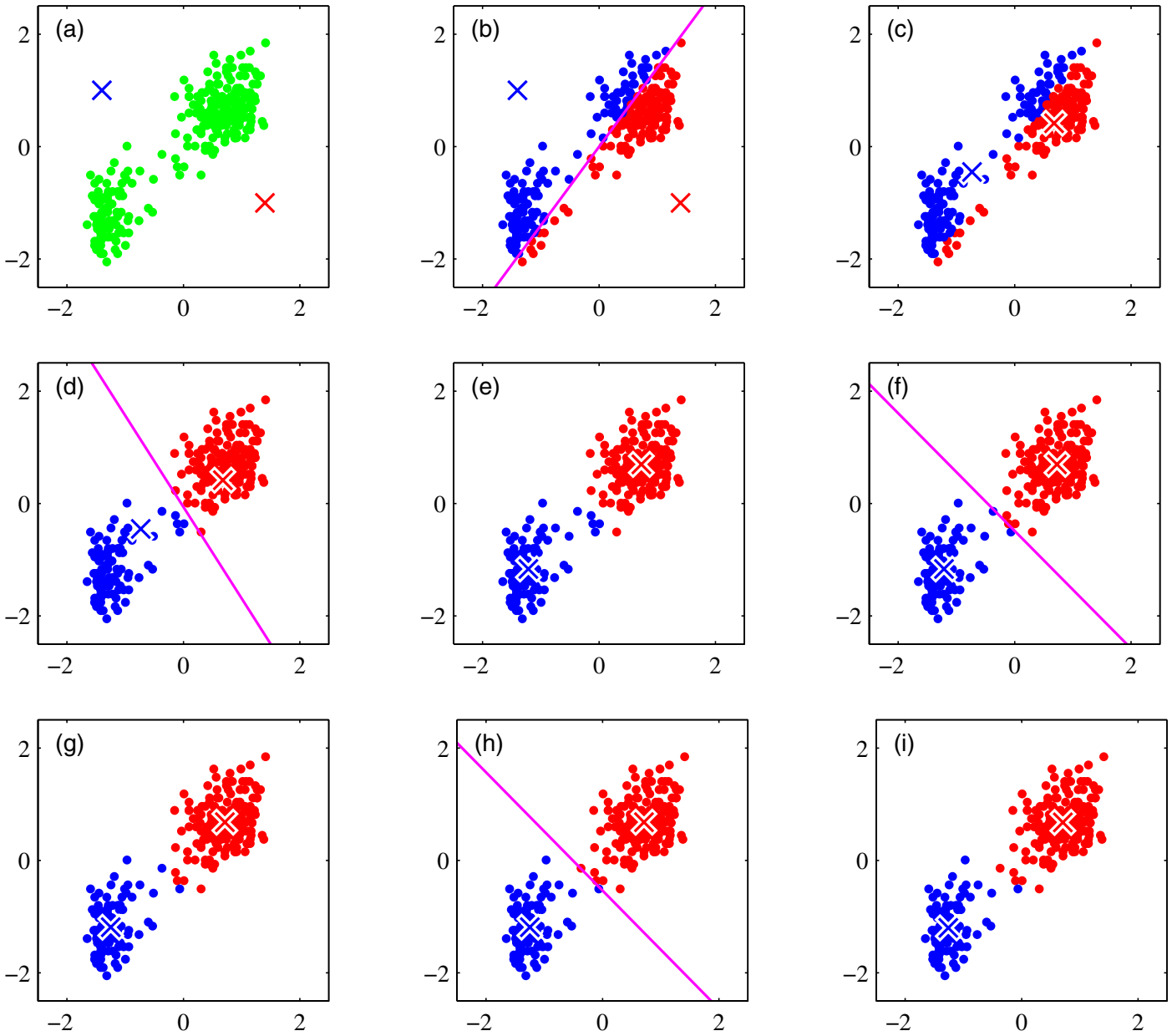
Group perceptually or according to another metric similar regions

How can we do it?

kMeans/Lloyd's Algorithm (informal):

- Initialize: pick K random points as cluster centers $\mu_k$
- Iterate:
  - Assign data points $x^{(i)}$ to closest cluster center according to some metric
  - Update the cluster center to be the average of its assigned points
  - Stopping criterion: when no points' assignments change

# 2D Example:



Formal description:

What cost function does kMeans optimize? (distortion measure)

$$\min_{\mu} \min_{r} \sum_{i \in \mathcal{D}} \sum_{k=1}^{K} \frac{1}{2} r_{ik} \|x^{(i)} - \mu_k\|_2^2 \quad \text{s.t.} \quad \begin{cases} r_{ik} \in \{0, 1\} & \forall i, k \\ \sum_{k=1}^{K} r_{ik} = 1 & \forall i \end{cases}$$

What does the constraint remind us of?

How to optimize this cost function?

Cost function:

$$\min_{\mu} \min_{r} \sum_{i \in \mathcal{D}} \sum_{k=1}^{K} \frac{1}{2} r_{ik} \|x^{(i)} - \mu_k\|_2^2 \quad \text{s.t.} \quad \begin{cases} r_{ik} \in \{0,1\} & \forall i,k \\ \sum_{k=1}^{K} r_{ik} = 1 & \forall i \end{cases}$$

Alternate optimization:

- Optimize for $r$ given $\mu$

$$r_{ik} = \begin{cases} 1 & \text{if } k = \arg\min_{k \in \{1,\ldots,K\}} \|x^{(i)} - \mu_k\|_2^2 \\ 0 & \text{otherwise} \end{cases}$$

- Optimize for $\mu$ given $r$

$$\nabla_{\mu_k}: \qquad \sum_{i \in \mathcal{D}} r_{ik}(x^{(i)} - \mu_k) = 0$$

$$\mu_k = \frac{\sum_{i \in \mathcal{D}} r_{ik} x^{(i)}}{\sum_{i \in \mathcal{D}} r_{ik}}$$

Properties of Algorithm:

- Local optimum is found
- Guaranteed to converge in a finite number of iterations
- Running time per iteration:
  - Assign data points to closest cluster center

$$O(KNd)$$

  - Change the cluster center to the average of its assigned points

$$O(Nd)$$

Extensions:

- kMeans is very sensitive to initialization: kMeans++
- kMeans depends on the feature space: Kernels
- Evaluation

kMeans++: How to initialize kMeans

- Randomly choose first center
- Pick new center with probability proportional to $\|x^{(i)} - \mu_k\|_2^2$ (contribution of $x^{(i)}$ to total error)
- Repeat until $K$ centers are chosen

Try multiple initializations and choose the best

Distance measure:

- Euclidean (most commonly used)
- Cosine
- Non-linear (via Kernels): $\phi(x^{(i)})$

How to evaluate clusters?

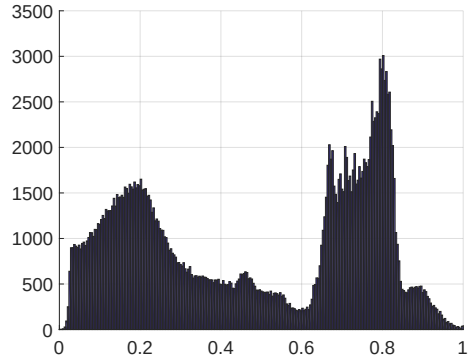
- Generative: How well are points reconstructed from the clusters

  Distortion
- Discriminative: How well do the clusters correspond to labels

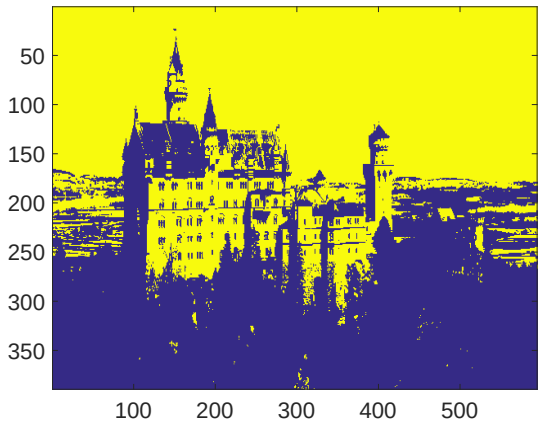
  Purity

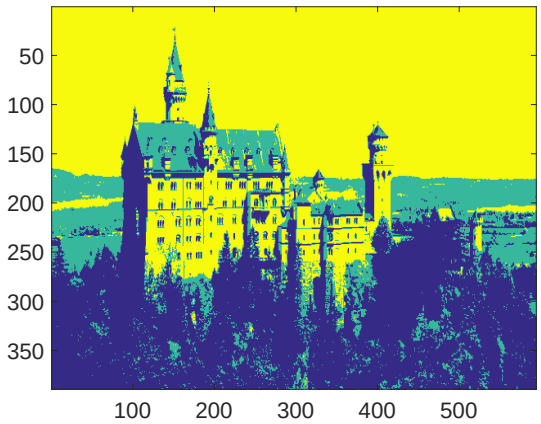
Applications

- Clustering
- Super-pixel segmentation

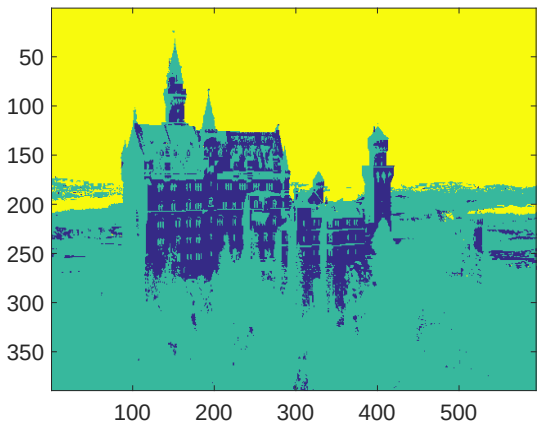



- Grayscale $x^{(i)} \in \mathbb{R}$ 2 clusters 3 clusters




- Color space $x^{(i)} \in \mathbb{R}^3$



How to obtain spatial smoothness?

Augment the feature space $\phi(x^{(i)})$ to contain spatial coordinates in addition to intensities

Superpixel segmentation

- 5D feature space (lab color space, 2d spatial coordinates)
- Distance metric treats color dimensions and spatial dimensions differently
- Initial cluster centers are spaced regularly on the image and slightly perturbed to avoid edges

Summary:

Pros:

- Simple
- Easy to implement

Cons:

- Need to choose $K$
- Sensitive to outliers
- Can get stuck in local minima
- All cluster centers have same parameters (non-adaptive)
- Can be slow $O(KNd)$

# Gaussian Mixture Models

**Goals of this section**

- Understanding Gaussian mixture models
- Getting to know more details about generative modeling
- Learning the relationship between Gaussian mixture models and kMeans

**Reading material:**

- C. Bishop; Pattern Recognition and Machine Learning; Chapter 9.2
- K. Murphy; Machine Learning: A Probabilistic Perspective; Chapter 11

Recall: Linear regression (discriminative)

$$p(y^{(i)}|x^{(i)}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(y^{(i)} - \boldsymbol{w}^\top \phi(x^{(i)}))^2\right)$$

Now: (generative)

$$p(x^{(i)}|\underbrace{\mu, \sigma}_{\theta \text{ or } \boldsymbol{w}}) = \mathcal{N}(x^{(i)}|\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x^{(i)} - \mu)^2\right)$$

Important difference: we are now interested in modeling the distribution of the data $x^{(i)}$ and not the class labels $y^{(i)}$. Though it is sometimes ambiguous what you call data or labels.

Given a dataset $\mathcal{D} = \{(x^{(i)})\}$ how to find $\theta = (\mu, \sigma)$ of

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

Minimize negative log-likelihood

Program:

$$\min_{\mu,\sigma} -\log \prod_{i\in\mathcal{D}} p(x^{(i)}|\mu, \sigma) := \sum_{i\in\mathcal{D}} \frac{1}{2\sigma^2}(x^{(i)} - \mu)^2 + \frac{N}{2}\log(2\pi\sigma^2)$$

Program:

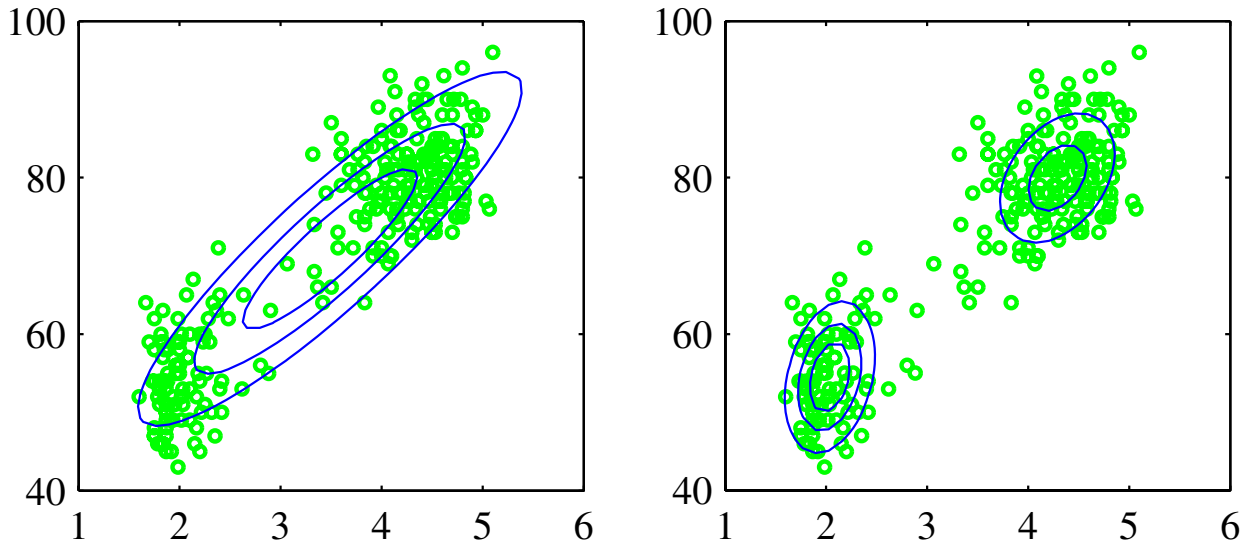
$$\min_{\mu,\sigma} -\log \prod_{i\in\mathcal{D}} p(x^{(i)}|\mu,\sigma) := \sum_{i\in\mathcal{D}} \frac{1}{2\sigma^2}(x^{(i)}-\mu)^2 + \frac{N}{2}\log(2\pi\sigma^2)$$

Optimality condition:

$$\frac{\partial}{\partial\mu}: \qquad \frac{1}{\sigma^2}\sum_{i\in\mathcal{D}}(x^{(i)}-\mu) = 0 \qquad \Longrightarrow \mu = \frac{1}{N}\sum_{i\in\mathcal{D}} x^{(i)}$$

$$\frac{\partial}{\partial\sigma}: \quad \frac{-1}{\sigma^3}\sum_{i\in\mathcal{D}}(x^{(i)}-\mu)^2 + \frac{N}{\sigma} = 0 \qquad \Longrightarrow \sigma^2 = \frac{1}{N}\sum_{i\in\mathcal{D}}(x^{(i)}-\mu)^2$$

Issue: single Gaussian isn't that flexible


100
80
60
40
1
2
3
4
5
6
100
80
60
40
1
2
3
4
5
6

Fix: linear superposition of Gaussians

$$p(x^{(i)}| \underbrace{\pi, \mu, \sigma}_{\text{all components}}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x^{(i)}|\mu_k, \sigma_k)$$

Constraints:

$$\sum_{k=1}^{K} \pi_k = 1 \qquad \pi_k \geq 0$$

Minimize negative log-likelihood:

$$\min_{\pi,\mu,\sigma} -\log \prod_{i \in \mathcal{D}} p(x^{(i)}|\pi, \mu, \sigma) := -\sum_{i \in \mathcal{D}} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(x^{(i)}|\mu_k, \sigma_k)$$

How to optimize:

No closed form solution. Gradient descent is possible.

Alternative: auxiliary/latent variable $z_{ik} \in \{0, 1\}$ with $\sum_{k=1}^{K} z_{ik} = 1 \; \forall i$

Marginal for $z_{ik}$

$$p(z_{ik} = 1) = \pi_k \qquad p(\boldsymbol{z}_i) = \prod_{k=1}^{K} \pi_k^{z_{ik}} \text{ where } \boldsymbol{z}_i = [z_{i1}, \ldots, z_{iK}]^\top$$

Conditional

$$p(x^{(i)}|z_{ik} = 1) = \mathcal{N}(x^{(i)}|\mu_k, \sigma_k)$$

Marginal for $x^{(i)}$

$$\begin{aligned} p(x^{(i)}|\pi, \mu, \sigma) &= \sum_{\boldsymbol{z}_i} p(x^{(i)}|\boldsymbol{z}_i) p(\boldsymbol{z}_i) = \sum_{\boldsymbol{z}_i} \prod_{k=1}^{K} \pi_k^{z_{ik}} \mathcal{N}(x^{(i)}|\mu_k, \sigma_k)^{z_{ik}} \\ &= \sum_{k=1}^{K} \pi_k \mathcal{N}(x^{(i)}|\mu_k, \sigma_k) \end{aligned}$$

Posterior:

$$r_{ik} = p(z_{ik} = 1|x^{(i)}) = \frac{p(z_{ik} = 1) p(x^{(i)}|z_{ik} = 1)}{\sum_{\hat{k}=1}^{K} p(z_{i\hat{k}} = 1) p(x^{(i)}|z_{i\hat{k}} = 1)} = \frac{\pi_k \mathcal{N}(x^{(i)}|\mu_k, \sigma_k)}{\sum_{\hat{k}=1}^{K} \pi_{\hat{k}} \mathcal{N}(x^{(i)}|\mu_{\hat{k}}, \sigma_{\hat{k}})}$$

With all those definitions at hand, minimize negative log-likelihood:

$$\min_{\pi,\mu,\sigma} -\log \prod_{i\in\mathcal{D}} p(x^{(i)}|\pi,\mu,\sigma) := -\sum_{i\in\mathcal{D}} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(x^{(i)}|\mu_k,\sigma_k) \quad \text{s.t.} \quad \sum_{k=1}^{K} \pi_k = 1$$

Stationary point: (per cluster weight $N_k = \sum_{i\in\mathcal{D}} r_{ik}$)

$$\frac{\partial}{\partial \mu_k}: \quad -\sum_{i\in\mathcal{D}} r_{ik} \left(-\frac{1}{2\sigma^2}(x^{(i)}-\mu_k)\right) = 0 \quad \Longrightarrow \quad \mu_k = \frac{1}{N_k}\sum_{i\in\mathcal{D}} r_{ik} x^{(i)}$$

$$\frac{\partial}{\partial \sigma_k}: \sum_{i\in\mathcal{D}} r_{ik} \left(\frac{1}{\sigma} - \frac{1}{\sigma^3}(x^{(i)}-\mu_k)^2\right) = 0 \implies \sigma_k^2 = \frac{1}{N_k}\sum_{i\in\mathcal{D}} r_{ik}(x^{(i)}-\mu_k)^2$$

$$\frac{\partial}{\partial \pi_k}: \quad \text{with Lagrange multiplier: } \sum_{i\in\mathcal{D}} \frac{\mathcal{N}(x^{(i)}|\mu_k,\sigma_k)}{\sum_{\hat{k}=1}^{K} \pi_{\hat{k}} \mathcal{N}(x^{(i)}|\mu_{\hat{k}},\sigma_{\hat{k}})} + \lambda = 0$$

multiplication with $\pi_k$ and summation over $k$: $\lambda = -N$

multiplication with $\pi_k$ and rearranging: $\pi_k = \frac{N_k}{N}$

Not a closed form solution

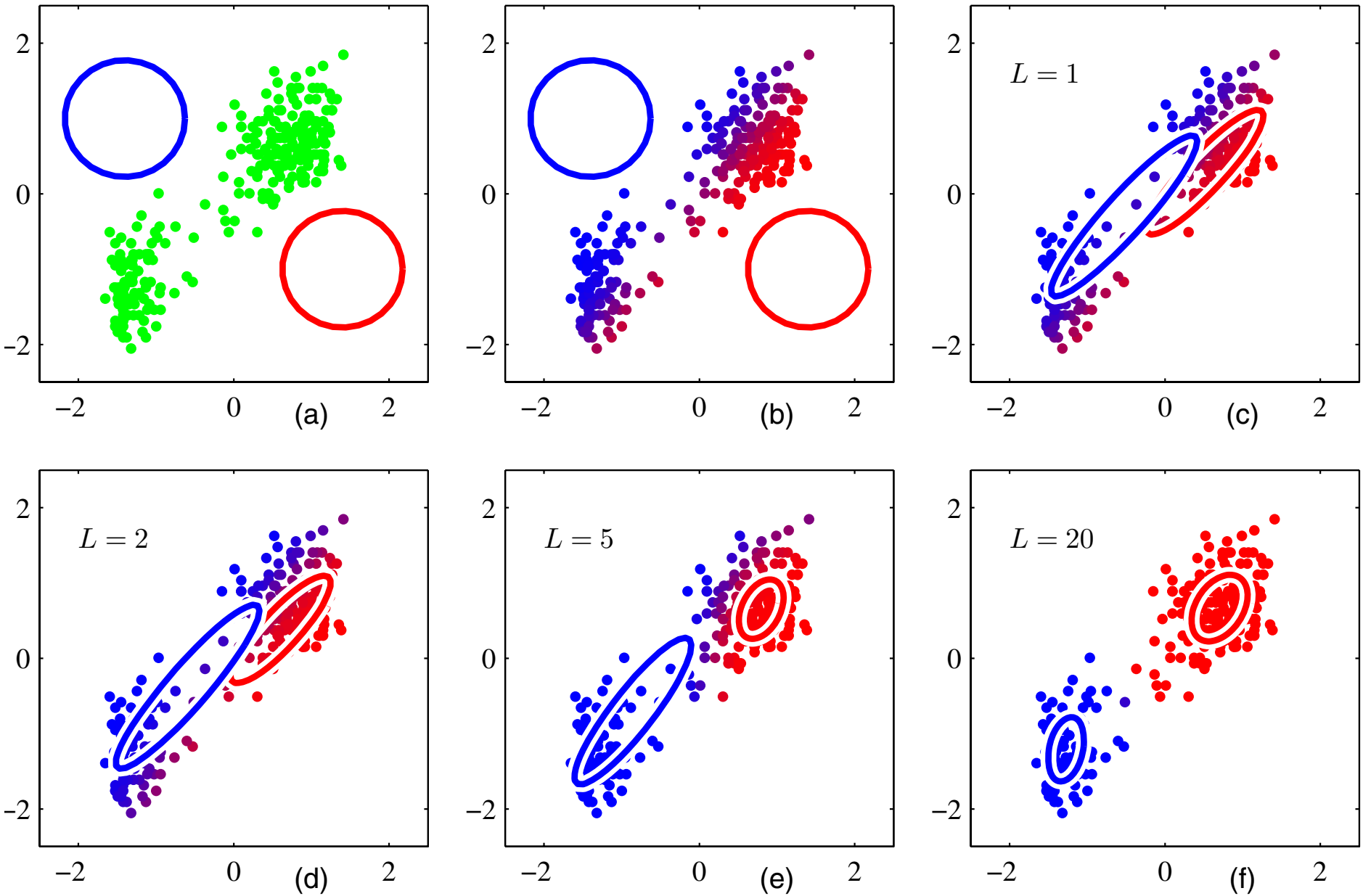
Gaussian Mixture Model Algorithm:

- Initialize $\mu, \sigma, \pi$
- Iterate:
  - E-Step: Update

$$r_{ik} = \frac{\pi_k \mathcal{N}(x^{(i)}|\mu_k, \sigma_k)}{\sum_{\hat{k}=1}^{K} \pi_{\hat{k}} \mathcal{N}(x^{(i)}|\mu_{\hat{k}}, \sigma_{\hat{k}})}$$

  - M-Step: Update

$$\begin{aligned}
\mu_k &= \frac{1}{N_k} \sum_{i \in \mathcal{D}} r_{ik} x^{(i)} \\
\sigma_k^2 &= \frac{1}{N_k} \sum_{i \in \mathcal{D}} r_{ik} (x^{(i)} - \mu_k)^2 \\
\pi_k &= \frac{N_k}{N}
\end{aligned}$$


2
0
−2
−2
0
2
(a)
(b)
$L = 1$
(c)
$L = 2$
(d)
$L = 5$
(e)
$L = 20$
(f)

Similarity to kMeans:

- $r_{ik}$ is an assignment of sample $i$ to cluster $k$, albeit a soft assignment
- $\mu_k$ are the cluster centers

Can we make this similarity formal?

Fix $\sigma_k^2 = \epsilon \; \forall k$

Responsibilities:

$$r_{ik} = \frac{\pi_k \exp(-\frac{1}{2\epsilon}(x^{(i)} - \mu_k)^2)}{\sum_{\hat{k}=1}^{K} \pi_{\hat{k}} \exp(-\frac{1}{2\epsilon}(x^{(i)} - \mu_{\hat{k}})^2)}$$

What happens for $\epsilon \to 0$?

- In the denominator the term for which $(x^{(i)} - \mu_{\hat{k}})^2$ is smallest goes to zero slowest
- All responsibilities will go to zero except the one for which $(x^{(i)} - \mu_{\hat{k}})^2$ is smallest, which will go to unity
- Responsibilities are hard assignments
- Cost function can be shown to be identical in the limit

# Expectation Maximization/Majorize-Minimize/Concave-convex procedure

**Goals of this lecture**

- Generalizing the kMeans/Gaussian mixture model algorithm
- Getting to know the Concave-convex procedure (CCCP)

**Reading material:**

- C. Bishop; Pattern Recognition and Machine Learning; Chapter 9.3, 9.4
- Yuille and Rangarajan; Concave Convex Procedure (CCCP); NIPS 2001
- K. Murphy; Machine Learning: A Probabilistic Perspective; Chapter 11

Recap:

$$\min_{\pi,\mu,\sigma} -\sum_{i\in\mathcal{D}} \ln \underbrace{\sum_{k=1}^{K} \pi_k \mathcal{N}(x^{(i)}|\mu_k,\sigma_k)}_{\sum_{z_i} \underbrace{p(x^{(i)}|z_i)p(z_i)}_{p(x^{(i)},z_i)}} \quad \text{s.t.} \quad \sum_{k=1}^{K} \pi_k = 1, \quad \pi_k \geq 0$$

More generally: (ignoring $\sum_i$)

$$\ln p_\theta(x^{(i)}) = \ln \sum_{\boldsymbol{z}_i} p_\theta(x^{(i)}, \boldsymbol{z}_i)$$

Two options:

- Evidence Lower Bound (ELBO)
- Concave-Convex Procedure/Majorize-Minimize

End up being identical

**Evidence Lower Bound:**

Goal: maximize likelihood

$$\ln p_\theta(x^{(i)}) = \ln \sum_{\boldsymbol{z}} p_\theta(x^{(i)}, \boldsymbol{z})$$

Let's introduce distribution $q(\boldsymbol{z})$ and rewrite:

$$\ln p_\theta(x^{(i)}) = \mathcal{L}(p_\theta(x^{(i)}, \boldsymbol{z}), q(\boldsymbol{z})) + D_{\mathrm{KL}}(q(\boldsymbol{z}), p_\theta(\boldsymbol{z}|x^{(i)}))$$

where

$$\begin{aligned} \mathcal{L}(p_\theta(x^{(i)}, \boldsymbol{z}), q(\boldsymbol{z})) &= \sum_{\boldsymbol{z}} q(\boldsymbol{z}) \ln \frac{p_\theta(x^{(i)}, \boldsymbol{z})}{q(\boldsymbol{z})} \\ D_{\mathrm{KL}}(q(\boldsymbol{z}), p_\theta(\boldsymbol{z}|x^{(i)})) &= \sum_{\boldsymbol{z}} q(\boldsymbol{z}) \ln \frac{q(\boldsymbol{z})}{p_\theta(\boldsymbol{z}|x^{(i)})} \end{aligned}$$

$D_{\mathrm{KL}}$: Kullback-Leibler divergence

Jensen's inequality:

$$f \text{ convex:} \quad f\left(\sum_{\boldsymbol{z}} q(\boldsymbol{z}) g(\boldsymbol{z})\right) \leq \sum_{\boldsymbol{z}} q(\boldsymbol{z}) f(g(\boldsymbol{z}))$$

$$f \text{ concave:} \quad f\left(\sum_{\boldsymbol{z}} q(\boldsymbol{z}) g(\boldsymbol{z})\right) \geq \sum_{\boldsymbol{z}} q(\boldsymbol{z}) f(g(\boldsymbol{z}))$$

Consequence for $D_{\mathrm{KL}}$:

$$\begin{aligned} -D_{\mathrm{KL}}(q(\boldsymbol{z}), p_\theta(\boldsymbol{z}|x^{(i)})) &= \sum_{\boldsymbol{z}} q(\boldsymbol{z}) \ln \frac{p_\theta(\boldsymbol{z}|x^{(i)})}{q(\boldsymbol{z})} \\ &\leq 0 \qquad \text{(pull out ln)} \end{aligned}$$

Kullback-Leibler divergence is non-negative

Consequence for log-likelihood:

$$\ln p_\theta(x^{(i)}) = \mathcal{L}(p_\theta(x^{(i)}, \boldsymbol{z}), q(\boldsymbol{z})) + D_{\mathrm{KL}}(q(\boldsymbol{z}), p_\theta(\boldsymbol{z}|x^{(i)}))$$

Lower bound:

$$\ln p_\theta(x^{(i)}) \geq \mathcal{L}(p_\theta(x^{(i)}, \boldsymbol{z}), q(\boldsymbol{z}))$$

Idea: instead of maximizing log-likelihood, let's maximize lower bound:

$$\max_{q,\theta} \mathcal{L}(p_\theta(x^{(i)}, \boldsymbol{z}), q(\boldsymbol{z}))$$

How: alternating optimization w.r.t. $q$ and $\theta$

Alternating optimization:

$$\max_{q,\theta} \mathcal{L}(p_\theta(x^{(i)}, \boldsymbol{z}), q(\boldsymbol{z}))$$

- Maximize w.r.t. $q$:

$$\Longrightarrow \qquad q(\boldsymbol{z}) = p_\theta(\boldsymbol{z}|x^{(i)})$$

$\ln p_\theta(x^{(i)})$ is upper bound and $\ln p_\theta(x^{(i)}) = \mathcal{L}(p_\theta(x^{(i)}, \boldsymbol{z}), q(\boldsymbol{z}))$ if $q(\boldsymbol{z}) = p_\theta(\boldsymbol{z}|x^{(i)})$ (KL-divergence is zero)

- Maximize w.r.t. $\theta$

Alternative to show that $q(\boldsymbol{z}) = p_\theta(\boldsymbol{z}|x^{(i)})$:

$$\max_q \mathcal{L}(p_\theta(x^{(i)}, \boldsymbol{z}), q(\boldsymbol{z}))$$

$$\max_q \sum_{\boldsymbol{z}} q(\boldsymbol{z}) \ln p_\theta(x^{(i)}, \boldsymbol{z}) + H(q(\boldsymbol{z})) \quad \text{s.t.} \begin{cases} q(\boldsymbol{z}) \geq 0 \\ \sum_{\boldsymbol{z}} q(\boldsymbol{z}) = 1 \end{cases}$$

How to solve:

Stationarity of Lagrangian

Solution:

$$q(\boldsymbol{z}) = \frac{p_\theta(x^{(i)}, \boldsymbol{z})}{\sum_{\boldsymbol{z}} p_\theta(x^{(i)}, \boldsymbol{z})} = p_\theta(\boldsymbol{z}|x^{(i)}) = r_i$$

In the Gaussian case:

$$
\begin{aligned}
\mathcal{L}(p_\theta(x^{(i)}, \boldsymbol{z}), q(\boldsymbol{z})) &= \sum_{\boldsymbol{z}} q(\boldsymbol{z}) \ln \frac{p_\theta(x^{(i)}, \boldsymbol{z})}{q(\boldsymbol{z})} \\
&= \sum_{\boldsymbol{z}} q(\boldsymbol{z}) \ln \frac{\prod_{k=1}^{K} \pi_k^{z_{ik}} \mathcal{N}(x^{(i)} | \mu_k, \sigma_k)^{z_{ik}}}{q(\boldsymbol{z})} \\
&= \sum_{\boldsymbol{z},k} q(\boldsymbol{z}) \ln \pi_k^{z_{ik}} \mathcal{N}(x^{(i)} | \mu_k, \sigma_k)^{z_{ik}} + H(q(\boldsymbol{z})) \\
&= \sum_{k} r_{ik} \ln \pi_k \mathcal{N}(x^{(i)} | \mu_k, \sigma_k) - \sum_{k} r_{ik} \ln r_{ik}
\end{aligned}
$$

Why is this easier to optimize than the original program?

In the general case:

$$p_\theta(x^{(i)}, \boldsymbol{z}) = \frac{1}{Z(\theta)} \exp F(x^{(i)}, \boldsymbol{z}, \theta) \qquad Z(\theta)\text{: partition function}$$

$$\begin{aligned} -\mathcal{L}(p_\theta(x^{(i)}, \boldsymbol{z}), q(\boldsymbol{z})) &= -\sum_{\boldsymbol{z}} q(\boldsymbol{z}) \ln \frac{p_\theta(x^{(i)}, \boldsymbol{z})}{q(\boldsymbol{z})} \\ &= \ln Z(\theta) - \sum_{\boldsymbol{z}} q(\boldsymbol{z}) F(x^{(i)}, \boldsymbol{z}, \theta) - H(q(\boldsymbol{z})) \end{aligned}$$

Keep that in mind

## Concave-convex procedure (CCCP):

Model:

$$p_\theta(x^{(i)}, \boldsymbol{z}) = \frac{1}{Z(\theta)} \exp F(x^{(i)}, \boldsymbol{z}, \theta)$$

Maximum Likelihood (marginalizing over latent space):

$$\min_\theta - \ln \sum_{\boldsymbol{z}} \frac{\exp F(x^{(i)}, \boldsymbol{z}, \theta)}{Z(\theta)}$$

$$\min_\theta \underbrace{\ln Z(\theta)}_{\text{convex if } F \text{ linear in } \theta} \underbrace{- \ln \sum_{\boldsymbol{z}} \exp F(x^{(i)}, \boldsymbol{z}, \theta)}_{\text{convex if } F \text{ linear in } \theta}$$

## Concave-convex procedure (CCCP):

- Initialize $\theta$
- Repeat:
  - Decompose concave part into 'convex + concave' at current $\theta$
  - Solve convex program

$$\min_{\theta} \underbrace{\ln Z(\theta)}_{\text{convex if } F \text{ linear in } \theta} \underbrace{- \ln \sum_{\boldsymbol{z}} \exp F(x^{(i)}, \boldsymbol{z}, \theta)}_{\text{convex if } F \text{ linear in } \theta}$$

How to decompose: (one possibility)

$$\begin{aligned} \ln \sum_{\boldsymbol{z}} \exp F(x^{(i)}, \boldsymbol{z}, \theta) &= \ln \sum_{\boldsymbol{z}} q(\boldsymbol{z}) \frac{\exp F(x^{(i)}, \boldsymbol{z}, \theta)}{q(\boldsymbol{z})} \qquad \text{(Jensen's)} \\ &= \max_{q(\boldsymbol{z})} \left( \sum_{\boldsymbol{z}} q(\boldsymbol{z}) F(x^{(i)}, \boldsymbol{z}, \theta) + H(q(\boldsymbol{z})) \right) \end{aligned}$$

## **Concave-convex procedure (CCCP):** Summary

$$\min_{\theta} \underbrace{\ln Z(\theta)}_{\text{convex if } F \text{ linear in } \theta} \underbrace{- \ln \sum_{\boldsymbol{z}} \exp F(x^{(i)}, \boldsymbol{z}, \theta)}_{\text{convex if } F \text{ linear in } \theta}$$

Decomposition:

$$\ln \sum_{\boldsymbol{z}} \exp F(x^{(i)}, \boldsymbol{z}, \theta) = \max_{q(\boldsymbol{z})} \left( \sum_{\boldsymbol{z}} q(\boldsymbol{z}) F(x^{(i)}, \boldsymbol{z}, \theta) + H(q(\boldsymbol{z})) \right)$$

Results in:

$$\min_{\theta, q} \ln Z(\theta) - \sum_{\boldsymbol{z}} q(\boldsymbol{z}) F(x^{(i)}, \boldsymbol{z}, \theta) - H(q(\boldsymbol{z}))$$