CSCI-567: Machine Learning (Spring 2019)

Prof. Victor Adamchik

U of Southern California

Mar. 26, 2019

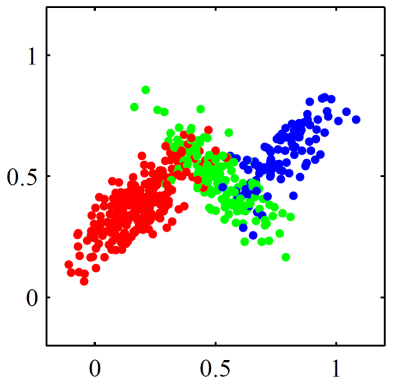
Outline

Outline

Gaussian mixture models

Gaussian mixture models (GMM) is a probabilistic approach for clustering.

We want to come up with a probabilistic model p to **explain how the data is generated**.



We will model each region with a Gaussian distribution.
To generate a point, we

- first randomly pick one of the Gaussian models,
- then draw a point according this Gaussian.

GMM: formal definition

A GMM has the following density function:

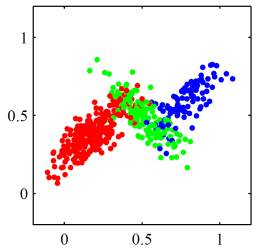
$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \omega_k N(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \sum_{k=1}^{K} \omega_k \frac{1}{\sqrt{(2\pi)^D |\boldsymbol{\Sigma}_k|}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^{\mathrm{T}} \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x}-\boldsymbol{\mu}_k)}$$

where

- K: the number of Gaussian components (same as #clusters we want)
- $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$: mean and covariance matrix of the k-th Gaussian
- $\omega_1, \ldots, \omega_K$: mixture weights, they represent how much each component contributes to the final distribution. It satisfies two properties:

$$\forall\ k,\ \omega_k > 0, \quad \text{and} \quad \sum_k \omega_k = 1$$

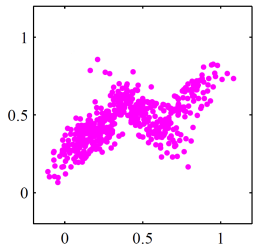
An example



The conditional distributions are

$$p(\boldsymbol{x} \mid z = \text{red}) = N(\boldsymbol{x} \mid \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$$
$$p(\boldsymbol{x} \mid z = \text{blue}) = N(\boldsymbol{x} \mid \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$$
$$p(\boldsymbol{x} \mid z = \text{green}) = N(\boldsymbol{x} \mid \boldsymbol{\mu}_3, \boldsymbol{\Sigma}_3)$$

Here z is the hidden (latent) variable.



The marginal distribution is

$$p(\boldsymbol{x}) = p(\text{red}) N(\boldsymbol{x} \mid \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + p(\text{blue}) N(\boldsymbol{x} \mid \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) + p(\text{green}) N(\boldsymbol{x} \mid \boldsymbol{\mu}_3, \boldsymbol{\Sigma}_3)$$

Learning GMMs

Learning a GMM means finding all the parameters $\boldsymbol{\theta} = \{\omega_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$.

How to learn these parameters?

An obvious attempt is **maximum-likelihood estimation (MLE)**: find

$$\underset{\boldsymbol{\theta}}{\operatorname{argmax}}\ \ln \prod_{n=1}^{N} p(\boldsymbol{x}_n\,; \boldsymbol{\theta}) = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \sum_{n=1}^{N} \ln p(\boldsymbol{x}_n\,; \boldsymbol{\theta}) \triangleq \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, P(\boldsymbol{\theta})$$

The problem is *intractable in general* (non-concave problem, also there is a latent parameter).

One solution is to still apply GD/SGD, but a much more effective approach is the **Expectation–Maximization (EM) algorithm**.

Preview of EM for learning GMMs

Step 0 Initialize $\omega_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ for each $k \in [K]$

Step 1 (E-Step) update the "soft assignment" (fixing parameters)

$$\gamma_{nk} = p(z_n = k \mid \boldsymbol{x}_n) \propto \omega_k N(\boldsymbol{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Step 2 (M-Step) update the model parameter (fixing assignments)

$$\omega_k = \frac{\sum_n \gamma_{nk}}{N} \qquad \boldsymbol{\mu}_k = \frac{\sum_n \gamma_{nk} \boldsymbol{x}_n}{\sum_n \gamma_{nk}}$$

$$\boldsymbol{\Sigma}_k = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^{\mathrm{T}}$$

Step 3 return to Step 1 if not converged

EM algorithm

In general EM is **a heuristic to solve MLE with latent variables** (not just GMM), i.e. find the maximizer of

$$P(\boldsymbol{\theta}) = \sum_{n=1}^{N} \ln p(\boldsymbol{x}_n \,; \boldsymbol{\theta})$$

- $\boldsymbol{\theta}$ is the parameters for a general probabilistic model
- $\boldsymbol{x}_n$'s are observed random variables
- z_n's are latent variables

Again, directly solving the objective is intractable.

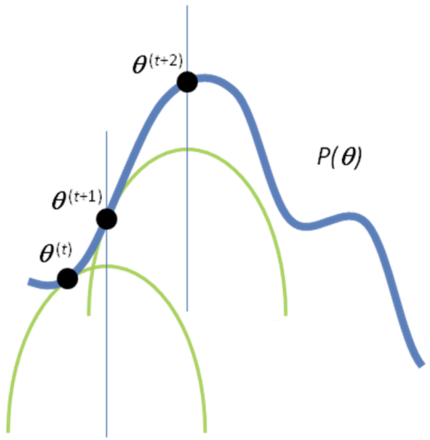
EM algorithm

A general algorithm for dealing with hidden data.

- EM is an optimization strategy for objective functions that can be interpreted as likelihoods in the presence of missing data.
- EM is much simpler than gradient methods: no need to choose step size.
- EM is an iterative algorithm with two steps:
  - E-step: fill-in hidden values using inference
  - M-step: apply standard MLE method to completed data
- We will prove that EM always converges to a local optimum of the likelihood.

High level idea

Keep maximizing **a lower bound of** P that is more manageable



Derivation of EM

Finding the lower bound of P:

$$\begin{aligned} \ln p(\boldsymbol{x} \,; \boldsymbol{\theta}) &= \ln \frac{p(\boldsymbol{x}, z \,; \boldsymbol{\theta})}{p(z|\boldsymbol{x} \,; \boldsymbol{\theta})} && \text{(true for any } z\text{)} \\ &= \mathbb{E}_{z\sim q}\left[\ln \frac{p(\boldsymbol{x}, z \,; \boldsymbol{\theta})}{p(z|\boldsymbol{x} \,; \boldsymbol{\theta})}\right] && \text{(true for any dist. } q\text{)} \end{aligned}$$

Let us recall the definition of expectation

$$\mathbb{E}_{z\sim q}\left[f(z)\right] = \sum_{z} q(z) f(z)$$

and entropy

$$H(z) = -\mathbb{E}_{z\sim q}\left[\ln q(z)\right] = -\sum_{z} q(z) \ln q(z)$$

Derivation of EM

Finding the lower bound of P:

$$\ln p(\boldsymbol{x}\,;\boldsymbol{\theta}) = \ln \frac{p(\boldsymbol{x}, z\,;\boldsymbol{\theta})}{p(z|\boldsymbol{x}\,;\boldsymbol{\theta})} \qquad \text{(true for any } z\text{)}$$

$$= \mathbb{E}_{z\sim q}\left[\ln \frac{p(\boldsymbol{x}, z\,;\boldsymbol{\theta})}{p(z|\boldsymbol{x}\,;\boldsymbol{\theta})}\right] \qquad \text{(true for any dist. } q\text{)}$$

$$= \mathbb{E}_{z\sim q}\left[\ln p(\boldsymbol{x}, z\,;\boldsymbol{\theta})\right] - \mathbb{E}_{z\sim q}\left[\ln q(z)\right] - \mathbb{E}_{z\sim q}\left[\ln \frac{p(z|\boldsymbol{x}\,;\boldsymbol{\theta})}{q(z)}\right]$$

$$= \mathbb{E}_{z\sim q}\left[\ln p(\boldsymbol{x}, z\,;\boldsymbol{\theta})\right] + H(q) - \mathbb{E}_{z\sim q}\left[\ln \frac{p(z|\boldsymbol{x}\,;\boldsymbol{\theta})}{q(z)}\right] \qquad (H \text{ is entropy})$$

$$\geq \mathbb{E}_{z\sim q}\left[\ln p(\boldsymbol{x}, z\,;\boldsymbol{\theta})\right] + H(q) - \ln \mathbb{E}_{z\sim q}\left[\frac{p(z|\boldsymbol{x}\,;\boldsymbol{\theta})}{q(z)}\right]$$

(Jensen's inequality)

Jensen's inequality

Claim: $\mathbb{E}\left[\ln X\right] \leq \ln\left(\mathbb{E}[X]\right)$

Proof. By the definition of $\mathbb{E}[X] = \frac{1}{N}\left(x_1 + x_2 + \ldots + x_n\right)$, then

$$\mathbb{E}\left[\ln X\right] = \frac{1}{N}\left(\ln x_1 + \ln x_2 + \ldots + \ln x_n\right) = \frac{1}{N}\ln \prod_{n=1}^{N} x_n$$

It follows,

$$\frac{1}{N}\ln \prod_{n=1}^{N} x_n \leq \ln \frac{1}{N}\sum_{n=1}^{N} x_n$$

$$\sqrt[N]{\prod_{n=1}^{N} x_n} \leq \frac{1}{N}\sum_{n=1}^{N} x_n$$

This is the AGM inequality. For $N = 2$, it is just $(x_1 - x_2)^2 \geq 0$.

Derivation of EM

After applying Jensen's inequality, we obtain

$$\ln p(\boldsymbol{x}\,;\boldsymbol{\theta}) \geq \mathbb{E}_{z\sim q}\left[\ln p(\boldsymbol{x}, z\,;\boldsymbol{\theta})\right] + H(q) - \ln \mathbb{E}_{z\sim q}\left[\frac{p(z|\boldsymbol{x}\,;\boldsymbol{\theta})}{q(z)}\right]$$

Next, we observe that

$$\mathbb{E}_{z\sim q}\left[\frac{p(z|\boldsymbol{x}\,;\boldsymbol{\theta})}{q(z)}\right] = \sum_z q(z)\left(\frac{p(z|\boldsymbol{x}\,;\boldsymbol{\theta})}{q(z)}\right) = \sum_z p(z|\boldsymbol{x}\,;\boldsymbol{\theta}) = 1$$

It follows,

$$\ln p(\boldsymbol{x}\,;\boldsymbol{\theta}) \geq \mathbb{E}_{z\sim q}\left[\ln p(\boldsymbol{x}, z\,;\boldsymbol{\theta})\right] + H(q)$$

Alternatively maximize the lower bound

We have found a lower bound for the log-likelihood function

$$P(\boldsymbol{\theta}) = \sum_{n=1}^{N} \ln p(\boldsymbol{x}_n\,;\boldsymbol{\theta})$$

$$\geq \sum_{n=1}^{N}\left(\mathbb{E}_{z_n\sim q_n}\left[\ln p(\boldsymbol{x}_n, z_n\,;\boldsymbol{\theta})\right] + H(q_n)\right) = F(\boldsymbol{\theta}, \{q_n\})$$

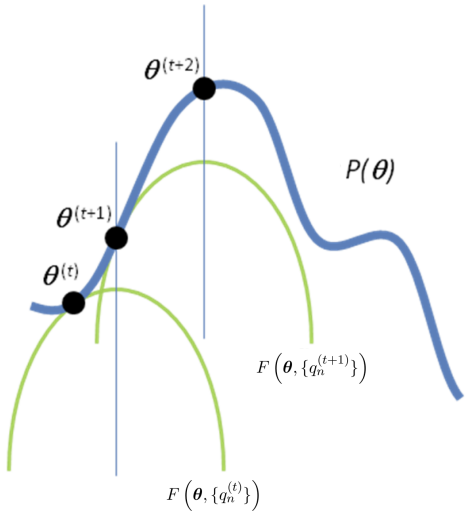
This holds for *any* $\{q_n\}$, so how do we choose?

Naturally, *the one that maximizes the lower bound* (i.e. the tightest lower bound)!

This is similar to K-means: we will alternatively maximizing F over $\{q_n\}$ and $\boldsymbol{\theta}$.

Pictorial explanation

$P(\boldsymbol{\theta})$ is non-concave, but $F\left(\boldsymbol{\theta}, \{q_n^{(t)}\}\right)$ often is concave and easy to maximize.



Maximizing over $\{q_n\}$

Fix $\boldsymbol{\theta}^{(t)}$, and maximize F over $\{q_n\}$

$$\begin{aligned}&\operatorname*{argmax}_{q_n} F(\boldsymbol{\theta}, \{q_n\}) = \operatorname*{argmax}_{q_n} \left(\mathbb{E}_{z_n \sim q_n}\left[\ln p(\boldsymbol{x}_n, z_n \,;\boldsymbol{\theta}^{(t)})\right] + H(q_n)\right)\\ &= \operatorname*{argmax}_{q_n} \sum_{k=1}^{K} \left(q_n(k) \ln p(\boldsymbol{x}_n, z_n = k \,;\boldsymbol{\theta}^{(t)}) - q_n(k) \ln q_n(k)\right)\end{aligned}$$

subject to conditions:

$$q_n(k) \geq 0 \quad \text{and} \quad \sum_k q_n(k) = 1$$

Next, write down the Lagrangian and then apply KKT conditions.

Maximizing over $\{q_n\}$

The solution to

$$\operatorname*{argmax}_{q_n} F(\boldsymbol{\theta}, \{q_n\}) = \operatorname*{argmax}_{q_n} \mathbb{E}_{z_n \sim q_n}\left[\ln p(\boldsymbol{x}_n, z_n \,;\boldsymbol{\theta}^{(t)})\right] + H(q_n)$$

is (you have to verify it by yourself)

$$q_n^{(t)}(z_n) = p(z_n = k \mid \boldsymbol{x}_n \,;\boldsymbol{\theta}^{(t)})$$

i.e., the *posterior distribution of* z_n given $\boldsymbol{x}_n$ and $\boldsymbol{\theta}^{(t)}$.

So at $\boldsymbol{\theta}^{(t)}$, we found the tightest lower bound $F\left(\boldsymbol{\theta}, \{q_n^{(t)}\}\right)$:

- $F\left(\boldsymbol{\theta}, \{q_n^{(t)}\}\right) \leq P(\boldsymbol{\theta})$ for all $\boldsymbol{\theta}$.
- $F\left(\boldsymbol{\theta}^{(t)}, \{q_n^{(t)}\}\right) = P(\boldsymbol{\theta}^{(t)})$

Maximizing over $\boldsymbol{\theta}$

Fix $\{q_n^{(t)}\}$, maximize over $\boldsymbol{\theta}$ (note, $H(q_n^{(t)})$ is independent of $\boldsymbol{\theta}$):

$$\begin{aligned}&\operatorname*{argmax}_{\boldsymbol{\theta}} F\left(\boldsymbol{\theta}, \{q_n^{(t)}\}\right)\\ &= \operatorname*{argmax}_{\boldsymbol{\theta}} \sum_{n=1}^{N} \mathbb{E}_{z_n \sim q_n^{(t)}}\left[\ln p(\boldsymbol{x}_n, z_n \,;\boldsymbol{\theta})\right]\\ &\triangleq \operatorname*{argmax}_{\boldsymbol{\theta}} Q(\boldsymbol{\theta} \,;\boldsymbol{\theta}^{(t)}) \qquad (\{q_n^{(t)}\} \text{ are computed via } \boldsymbol{\theta}^{(t)})\end{aligned}$$

Q is called a **complete likelihood** and is usually more tractable, since z_n are not latent variables anymore.

General EM algorithm

Step 0 Initialize $\boldsymbol{\theta}^{(1)}$, $t = 1$

Step 1 (E-Step) **update the posterior of latent variables**

$$q_n^{(t)}(\cdot) = p(\cdot \mid \boldsymbol{x}_n \,; \boldsymbol{\theta}^{(t)})$$

and obtain **Expectation** of complete likelihood

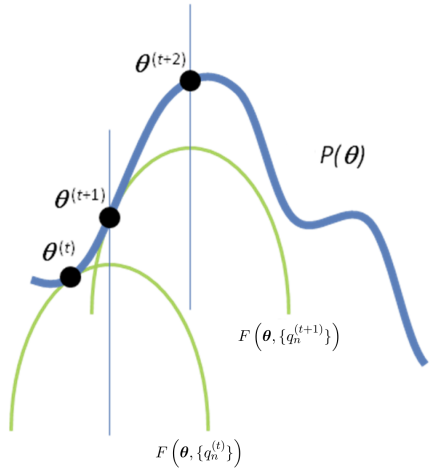
$$Q(\boldsymbol{\theta} \,; \boldsymbol{\theta}^{(t)}) = \sum_{n=1}^{N} \mathbb{E}_{z_n \sim q_n^{(t)}} \left[\ln p(\boldsymbol{x}_n, z_n \,; \boldsymbol{\theta})\right]$$

Step 2 (M-Step) **update the model parameter** via **Maximization**

$$\boldsymbol{\theta}^{(t+1)} \leftarrow \operatorname*{argmax}_{\boldsymbol{\theta}} Q(\boldsymbol{\theta} \,; \boldsymbol{\theta}^{(t)})$$

Step 3 $t \leftarrow t + 1$ and return to Step 1 if not converged

Pictorial explanation



$P(\boldsymbol{\theta})$ is non-concave, but $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)})$ often is concave and easy to maximize.

$$\begin{aligned} P(\boldsymbol{\theta}^{(t+1)}) &\geq F\left(\boldsymbol{\theta}^{(t+1)} \,; \{q_n^{(t)}\}\right) \\ &\geq F\left(\boldsymbol{\theta}^{(t)} \,; \{q_n^{(t)}\}\right) \\ &= P(\boldsymbol{\theta}^{(t)}) \end{aligned}$$

So EM always increases the objective value and will converge to some local maximum (similar to K-means).

Apply EM to learn GMMs

E-Step:

$$\begin{aligned} q_n^{(t)}(z_n = k) &= p\left(z_n = k \mid \boldsymbol{x}_n \,; \boldsymbol{\theta}^{(t)}\right) \\ &= p\left(z_n = k \,; \boldsymbol{\theta}^{(t)}\right) p(\boldsymbol{x}_n \mid z_n = k \,; \boldsymbol{\theta}^{(t)}) \\ &= \omega_k^{(t)} N\left(\boldsymbol{x}_n \mid \boldsymbol{\mu}_k^{(t)}, \boldsymbol{\Sigma}_k^{(t)}\right) \end{aligned}$$

This computes the "soft assignment" $\gamma_{nk} = q_n^{(t)}(z_n = k)$, i.e. conditional probability of $\boldsymbol{x}_n$ belonging to cluster k.

Apply EM to learn GMMs

M-Step:

$$\begin{aligned} \operatorname*{argmax}_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}) &= \operatorname*{argmax}_{\boldsymbol{\theta}} \sum_{n=1}^{N} \mathbb{E}_{z_n \sim q_n^{(t)}} \left[\ln p(\boldsymbol{x}_n, z_n \,; \boldsymbol{\theta})\right] \\ &= \operatorname*{argmax}_{\boldsymbol{\theta}} \sum_{n=1}^{N} \mathbb{E}_{z_n \sim q_n^{(t)}} \left[\ln p(z_n \,; \boldsymbol{\theta}) + \ln p(\boldsymbol{x}_n | z_n \,; \boldsymbol{\theta})\right] \\ &= \operatorname*{argmax}_{\{\omega_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}} \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} \left(\ln \omega_k + \ln N(\boldsymbol{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right) \end{aligned}$$

To find $\omega_1, \ldots, \omega_K$, solve

$$\operatorname*{argmax}_{\boldsymbol{\omega}} \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} \ln \omega_k$$

To find each $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$, solve

$$\operatorname*{argmax}_{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k} \sum_{n=1}^{N} \gamma_{nk} \ln N(\boldsymbol{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

M-Step (continued)

Solutions to previous two problems are very natural (see slide 8), for each k

$$\omega_k = \frac{\sum_n \gamma_{nk}}{N}$$

i.e. (weighted) fraction of examples belonging to cluster k

$$\boldsymbol{\mu}_k = \frac{\sum_n \gamma_{nk} \boldsymbol{x}_n}{\sum_n \gamma_{nk}}$$

i.e. (weighted) average of examples belonging to cluster k

$$\boldsymbol{\Sigma}_k = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^{\mathrm{T}}$$

i.e (weighted) covariance of examples belonging to cluster k

GMM: putting it together

EM for clustering:

Step 0 Initialize $\omega_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ for each $k \in [K]$

Step 1 (E-Step) update the "soft assignment" (fixing parameters)

$$\gamma_{nk} = p(z_n = k \mid \boldsymbol{x}_n) \propto \omega_k N(\boldsymbol{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Step 2 (M-Step) update the model parameter (fixing assignments)

$$\omega_k = \frac{\sum_n \gamma_{nk}}{N} \qquad \boldsymbol{\mu}_k = \frac{\sum_n \gamma_{nk} \boldsymbol{x}_n}{\sum_n \gamma_{nk}}$$

$$\boldsymbol{\Sigma}_k = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^{\mathrm{T}}$$

Step 3 return to Step 1 if not converged

Connection to K-means

K-means is in fact a special case of EM for (a simplified) GMM:

Let $\boldsymbol{\Sigma}_k = \sigma^2 \boldsymbol{I}$ for some fixed σ, so only ω_k and $\boldsymbol{\mu}_k$ are parameters.

EM becomes K-means:

$$\operatorname*{argmax}_{\boldsymbol{\theta}} \prod_{n=1}^{N} p(\boldsymbol{x}_n ; \boldsymbol{\theta}) = \operatorname*{argmax}_{\boldsymbol{\theta}} \prod_{n=1}^{N} \sum_{k=1}^{K} p(z_n = k) N(\boldsymbol{x}_n | \boldsymbol{\mu}_k)$$

If we assume hard assignments $p(z_n = k) = 1$, if $k = C(n)$, then

$$\operatorname*{argmax}_{\boldsymbol{\theta}} \prod_{n=1}^{N} p(\boldsymbol{x}_n ; \boldsymbol{\theta}) = \operatorname*{argmax}_{\boldsymbol{\theta}} \prod_{n=1}^{N} N(\boldsymbol{x}_n | \boldsymbol{\mu}_{C(n)})$$

$$= \operatorname*{argmax}_{\boldsymbol{\theta}} \prod_{n=1}^{N} \exp\left(\frac{-1}{2\sigma^2} \|\boldsymbol{x}_n - \boldsymbol{\mu}_{C(n)}\|_2^2 \right) = \operatorname*{argmax}_{\boldsymbol{\mu}, C} \sum_{n=1}^{N} \|\boldsymbol{x}_n - \boldsymbol{\mu}_{C(n)}\|_2^2$$

GMM is a soft version of K-means and it provides a probabilistic interpretation of the data.

Outline

Density estimation

Observe what we have done indirectly for clustering with GMMs is:

Given a training set $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, **estimate a density function p that could have generated this dataset** (via $\boldsymbol{x}_n \overset{i.i.d.}{\sim} p$).

This is exactly the problem of *density estimation*, another important unsupervised learning problem.

Useful for many downstream applications

- we have seen clustering already, will see more applications today
- these applications also *provide a way to measure quality of the density estimator*

Parametric generative models

Parametric estimation assumes a generative model parametrized by $\boldsymbol{\theta}$:

$$p(\boldsymbol{x}) = p(\boldsymbol{x}\,; \boldsymbol{\theta})$$

Examples:

- GMM: $p(\boldsymbol{x}\,; \boldsymbol{\theta}) = \sum_{k=1}^{K} \omega_k N(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ where $\boldsymbol{\theta} = \{\omega_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$
- Multinomial for 1D examples with K possible values

$$p(x = k\,; \boldsymbol{\theta}) = \theta_k$$

  where $\boldsymbol{\theta}$ is a distribution over K elements.

Size of $\boldsymbol{\theta}$ is independent of the training set size, so it's parametric.

Parametric methods

Again, we apply **MLE** to learn the parameters $\boldsymbol{\theta}$:

$$\operatorname*{argmax}_{\boldsymbol{\theta}} = \sum_{n=1}^{N} \ln p(x_n\,; \boldsymbol{\theta})$$

For some cases this is intractable and we can use EM to approximately solve MLE (e.g. GMMs).

For some other cases this admits a simple closed-form solution (e.g. multinomial).

MLE for multinomial

$$\begin{aligned}\operatorname*{argmax}_{\boldsymbol{\theta}} &= \sum_{n=1}^{N} \ln p(x = x_n\,; \boldsymbol{\theta}) = \sum_{n=1}^{N} \ln \theta_{x_n} \\ &= \sum_{k=1}^{K} \sum_{n: x_n = k} \ln \theta_k = \sum_{k=1}^{K} z_k \ln \theta_k\end{aligned}$$

where $z_k = |\{n : x_n = k\}|$ is **the number of examples with value k**.

The solution (your TA4) is simply

$$\theta_k = \frac{z_k}{N} \propto z_k,$$

i.e. the fraction of examples with value k.

Nonparametric models

Can we estimate *without* assuming a fixed generative model?

Kernel density estimation (KDE) is a common approach for nonparametric density estimation.

Here "kernel" means something different from what we have seen for "kernel function".

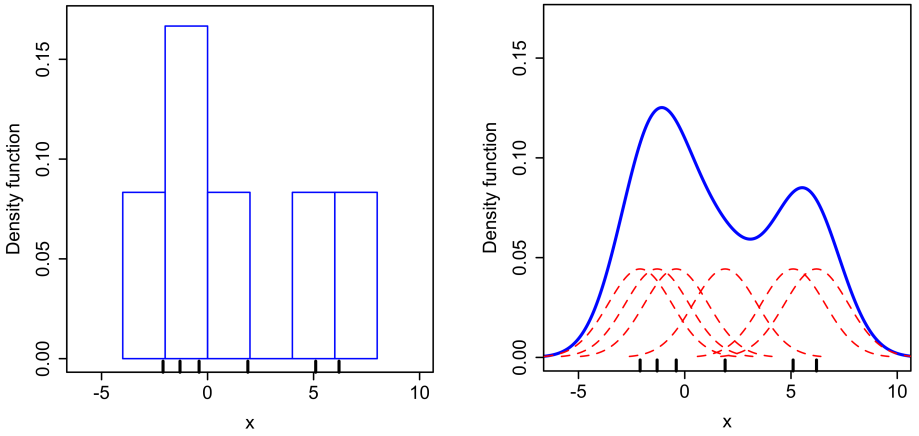
We focus on the 1D (continuous) case.

High level idea

picture from Wikipedia

Construct something similar to a **histogram**:

- for each data point, create a "hump" (via a kernel)
- sum up all the humps; more data - a higher hump



Kernel

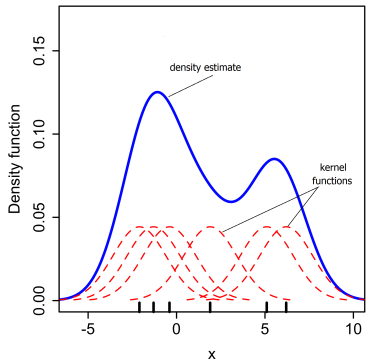
KDE with **a kernel** $K(x)$: $\mathbb{R} \to \mathbb{R}$ centered at x_n:

$$p(x) = \frac{1}{N} \sum_{n=1}^{N} K(x - x_n)$$

Many choices for K, for example, $K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$, the standard Gaussian density

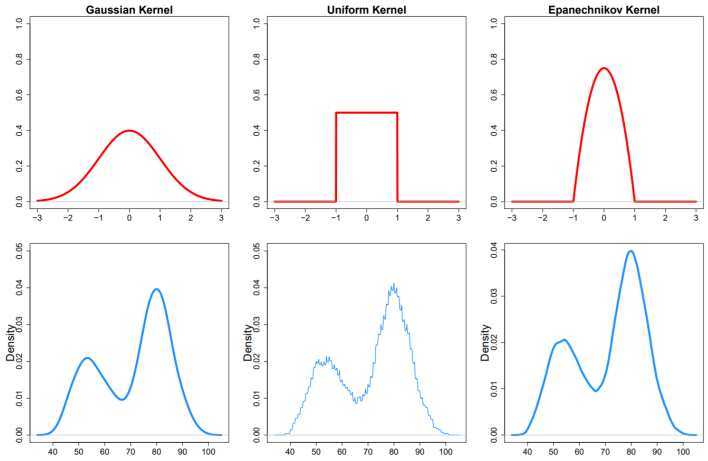
Properties of a kernel:

- symmetry: $K(x) = K(-x)$
- $\int_{-\infty}^{\infty} K(x)dx = 1$, this insures p is a density function.



Different kernels $K(x)$

$$\frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \qquad \frac{1}{2}\mathbb{I}[|x| \leq 1] \qquad \frac{3}{4}\max\{1 - x^2, 0\}$$



Bandwidth

If $K(x)$ is a kernel, then for any $h > 0$

$$K_h(u) \triangleq \frac{1}{h} K\left(\frac{x}{h}\right) \qquad \text{(stretching the kernel)}$$

can be used as a kernel too (verify the two properties yourself)

So, general KDE is determined by both the kernel K and the bandwidth h

$$p(x) = \frac{1}{N}\sum_{n=1}^{N} K_h(x - x_n) = \frac{1}{Nh}\sum_{n=1}^{N} K\left(\frac{x - x_n}{h}\right)$$

- x_n controls the center of each hump
- h controls the width/variance of the humps

Effect of bandwidth

picture from Wikipedia

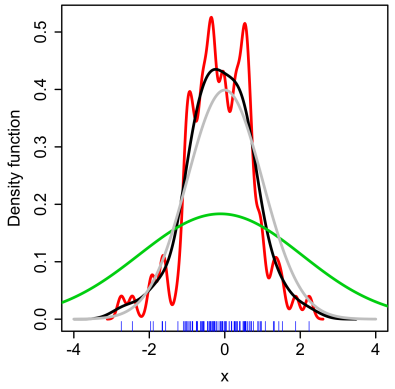
A larger h will smooth a density.

A small h will yield a density that is spiky and very hard to interpret.

Assume Gaussian kernel.

Gray curve is ground-truth

- Red: $h = 0.05$
- Black: $h = 0.337$
- Green: $h = 2$



Bandwidth selection

Selecting h is a deep topic

- one can also do cross-validation based on downstream applications
- there are theoretically-motivated approaches

  Find a value of h that minimizes the error between the estimated density and the true density:

$$\mathbb{E}\left[(p_{KDE}(x) - p(x))^2\right] = \mathbb{E}\left[p_{KDE}(x) - p(x)\right]^2 + Var\left[p_{KDE}(x)\right]$$

  This expression is an example of the bias-variance tradeoff, which we saw in the earlier lecture.

Outline

Bayes optimal classifier

Suppose the data $(\boldsymbol{x}_n, y_n)$ is drawn from a joint distribution $p(x, y)$, the **Bayes optimal classifier** is

$$f^*(\boldsymbol{x}) = \operatorname*{argmax}_{c \in [\mathsf{C}]} p(c \mid \boldsymbol{x})$$

i.e. predict the class with the largest conditional probability.

$p(x, y)$ is of course unknown, but we can estimate it, which is *exactly a density estimation problem!*

Observe that

$$p(\boldsymbol{x}, y) = p(y)p(\boldsymbol{x} \mid y)$$

To estimate $p(\boldsymbol{x} \mid y = c)$ for some $c \in [\mathsf{C}]$, we are doing density estimation using data with label $y = c$.

Discrete features

For a label $c \in [\mathsf{C}]$,

$$p(y = c) = \frac{|\{n : y_n = c\}|}{N}$$

For each possible value k of a discrete feature d,

$$p(x_d = k \mid y = c) = \frac{|\{n : x_{nd} = k, y_n = c\}|}{|\{n : y_n = c\}|}$$

Continuous features

If the feature is continuous, we can do

- parametric estimation, e.g. via a Gaussian

$$p(x_d = x \mid y = c) = \frac{1}{\sqrt{2\pi}\sigma_{cd}} \exp\left(-\frac{(x - \mu_{cd})^2}{2\sigma_{cd}^2}\right)$$

  where μ_{cd} and σ_{cd}^2 are the empirical mean and variance of feature d among all examples with label c.

- or nonparametric estimation, e.g. via a kernel K and bandwidth h:

$$p(x_d = x \mid y = c) = \frac{1}{|\{n : y_n = c\}|} \sum_{n : y_n = c} K_h(x - x_{nd})$$

How to predict?

Using Naive Bayes assumption:

$$p(\boldsymbol{x} \mid y = c) = \prod_{d=1}^{\mathsf{D}} p(x_d \mid y = c)$$

the **prediction** for a new example x is

$$\begin{aligned}
\operatorname*{argmax}_{c \in [\mathsf{C}]} p(y = c \mid \boldsymbol{x}) &= \operatorname*{argmax}_{c \in [\mathsf{C}]} \frac{p(\boldsymbol{x} \mid y = c)p(y = c)}{p(\boldsymbol{x})} \\
&= \operatorname*{argmax}_{c \in [\mathsf{C}]} \left(p(y = c) \prod_{d=1}^{\mathsf{D}} p(x_d \mid y = c)\right) \\
&= \operatorname*{argmax}_{c \in [\mathsf{C}]} \left(\ln p(y = c) + \sum_{d=1}^{\mathsf{D}} \ln p(x_d \mid y = c)\right)
\end{aligned}$$

Naive Bayes

For **discrete features**, plugging in previous MLE estimations gives

$$\begin{aligned}
&\operatorname*{argmax}_{c\in[\mathsf{C}]} p(y=c \mid x) \\
&= \operatorname*{argmax}_{c\in[\mathsf{C}]} \left(\ln p(y=c) + \sum_{d=1}^{\mathsf{D}} \ln p(x_d \mid y=c) \right) \\
&= \operatorname*{argmax}_{c\in[\mathsf{C}]} \left(\ln |\{n : y_n = c\}| + \sum_{d=1}^{\mathsf{D}} \ln \frac{|\{n : x_{nd} = x_d, y_n = c\}|}{|\{n : y_n = c\}|} \right)
\end{aligned}$$

Naive Bayes

For **continuous features** with a Gaussian model,

$$\begin{aligned}
&\operatorname*{argmax}_{c\in[\mathsf{C}]} p(y=c \mid \boldsymbol{x}) \\
&= \operatorname*{argmax}_{c\in[\mathsf{C}]} \left(\ln p(y=c) + \sum_{d=1}^{\mathsf{D}} \ln p(x_d \mid y=c) \right) \\
&= \operatorname*{argmax}_{c\in[\mathsf{C}]} \left(\ln |\{n : y_n = c\}| + \sum_{d=1}^{\mathsf{D}} \ln \left(\frac{1}{\sqrt{2\pi}\sigma_{cd}} \exp\left(-\frac{(x_d - \mu_{cd})^2}{2\sigma_{cd}^2} \right) \right) \right) \\
&= \operatorname*{argmax}_{c\in[\mathsf{C}]} \left(\ln |\{n : y_n = c\}| - \sum_{d=1}^{\mathsf{D}} \left(\ln \sigma_{cd} + \frac{(x_d - \mu_{cd})^2}{2\sigma_{cd}^2} \right) \right)
\end{aligned}$$

Connection to logistic regression

Let us fix the variance for each feature to be σ (i.e. not a parameter of the model any more), then the prediction becomes

$$\begin{aligned}
&\operatorname*{argmax}_{c\in[\mathsf{C}]} p(y=c \mid \boldsymbol{x}) \\
&= \operatorname*{argmax}_{c\in[\mathsf{C}]} \left(\ln |\{n : y_n = c\}| - \sum_{d=1}^{\mathsf{D}} \left(\ln \sigma + \frac{(x_d - \mu_{cd})^2}{2\sigma^2} \right) \right) \\
&= \operatorname*{argmax}_{c\in[\mathsf{C}]} \left(\ln |\{n : y_n = c\}| - \frac{\|\boldsymbol{x}\|_2^2}{2\sigma^2} - \sum_{d=1}^{\mathsf{D}} \frac{\mu_{cd}^2}{2\sigma^2} + \sum_{d=1}^{\mathsf{D}} \frac{\mu_{cd}}{\sigma^2} x_d \right) \\
&= \operatorname*{argmax}_{c\in[\mathsf{C}]} \left(w_{c0} + \sum_{d=1}^{\mathsf{D}} w_{cd} x_d \right) = \operatorname*{argmax}_{c\in[\mathsf{C}]} \boldsymbol{w}_c^{\mathrm{T}} \boldsymbol{x} \qquad (\textit{linear classifier!})
\end{aligned}$$

where we denote $w_{c0} = \ln |\{n : y_n = c\}| - \sum_{d=1}^{\mathsf{D}} \frac{\mu_{cd}^2}{2\sigma^2}$ and $w_{cd} = \frac{\mu_{cd}}{\sigma^2}$.

Connection to logistic regression

You can verify

$$p(y=c \mid x) \propto e^{\boldsymbol{w}_c^{\mathrm{T}} \boldsymbol{x}}$$

This is exactly the **softmax** function, the same model we used for a probabilistic interpretation of logistic regression!

So what is different then? They **learn the parameters in different ways**:

- both via MLE, one on $p(y=c \mid x)$, the other on $p(x, y)$
- solutions are different: logistic regression has no closed-form, naive Bayes admits a simple closed-form

Two different modeling paradigms

Suppose the training data is from an *unknown* joint probabilistic model $p(\boldsymbol{x}, y)$. There are two kinds of classification models in machine learning — generative models and discriminative models.

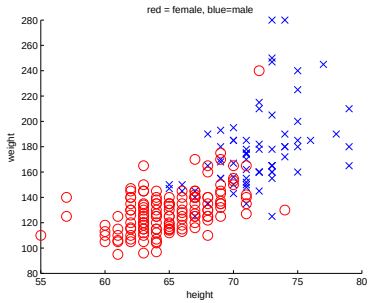
Differences in *assuming* models for the data

- the generative approach requires we specify the model for the joint distribution (such as Naive Bayes), and thus, maximize the *joint* likelihood $\sum_n \log p(\boldsymbol{x}_n, y_n)$
- the discriminative approach (discriminative) requires only specifying a model for the conditional distribution (such as logistic regression), and thus, maximize the *conditional* likelihood $\sum_n \log p(y_n|\boldsymbol{x}_n)$
- Sometimes, modeling by discriminative approach is easier
- Sometimes, parameter estimation by generative approach is easier

Generative model v.s discriminative model

	Discriminative model	Generative model
Example	logistic regression	naive Bayes
Model	conditional $p(y \mid x)$	joint $p(x, y)$ (might have same $p(y \mid x)$)
Learning	MLE	MLE
Accuracy	usually better for large N	usually better for small N
Remark		more flexible, can generate data after learning

Determining sex (man or woman) based on measurements

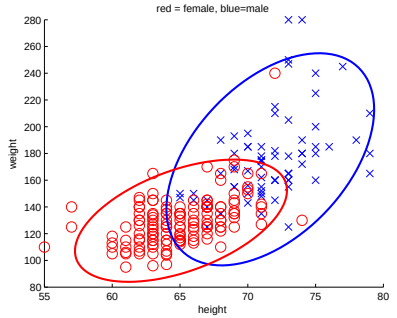


Example: Generative approach

Propose a model of the joint distribution of (x = height, y =sex)

our data

Sex	Height
1	$6'$
2	$5'2"$
1	$5'6"$
1	$6'2"$
2	$5.7"$
...	...

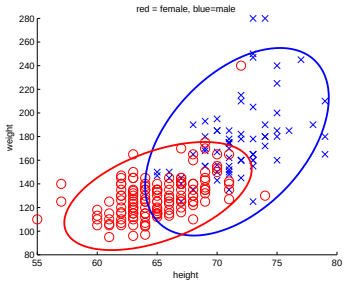


Intuition: we will model how heights vary (according to a Gaussian) in each sub-population (male and female).
Note: This is similar to Naive Bayes for detecting spam emails.

Model of the joint distribution

$$p(x,y) = p(y)p(x|y)$$
$$= \begin{cases} p_1 \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}} & \text{if } y = 1 \\ p_2 \frac{1}{\sqrt{2\pi}\sigma_2} e^{-\frac{(x-\mu_2)^2}{2\sigma_2^2}} & \text{if } y = 2 \end{cases}$$

where $p_1 + p_2 = 1$ represents two *prior* probabilities that x is given the label 1 or 2 respectively. $p(x|y)$ is assumed to be Gaussians.



Parameter estimation

Likelihood of the training data $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^N$ with $y_n \in \{1, 2\}$

$$\log P(\mathcal{D}) = \sum_n \log p(x_n, y_n)$$
$$= \sum_{n:y_n=1} \log \left(p_1 \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(x_n-\mu_1)^2}{2\sigma_1^2}} \right)$$
$$+ \sum_{n:y_n=2} \log \left(p_2 \frac{1}{\sqrt{2\pi}\sigma_2} e^{-\frac{(x_n-\mu_2)^2}{2\sigma_2^2}} \right)$$

Maximize the likelihood function

$$(p_1^*, p_2^*, \mu_1^*, \mu_2^*, \sigma_1^*, \sigma_2^*) = \operatorname{argmax} \log P(\mathcal{D})$$

Decision boundary

The decision boundary between two classes is defined by

$$p(y = 1|x) \geq p(y = 2|x)$$

which is equivalent to

$$p(x|y = 1)p(y = 1) \geq p(x|y = 2)p(y = 2)$$

Namely,

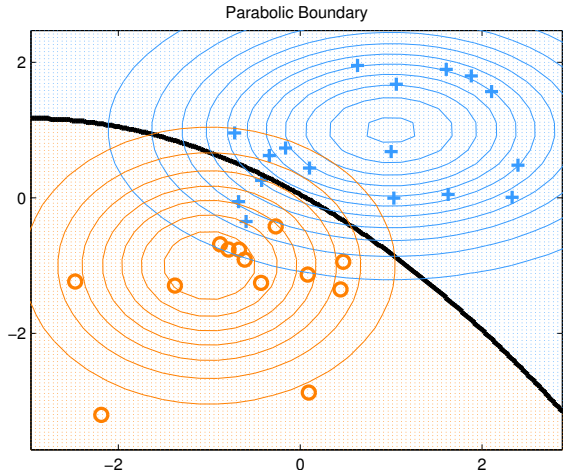
$$-\frac{(x-\mu_1)^2}{2\sigma_1^2} - \log\sqrt{2\pi}\sigma_1 + \log p_1 \geq -\frac{(x-\mu_2)^2}{2\sigma_2^2} - \log\sqrt{2\pi}\sigma_2 + \log p_2$$

It is quadratic in x. It follows (for some a, b and c, that

$$ax^2 + bx + c \geq 0$$

The decision boundary is *not linear*!

Example of nonlinear decision boundary



Note: the boundary is characterized by a quadratic function, giving rise to the shape of parabolic curve.

A special case

What if we assume the two Gaussians have the same variance?

We will get a *linear* decision boundary

From the previous slide:

$$-\frac{(x-\mu_1)^2}{2\sigma_1^2} - \log\sqrt{2\pi}\sigma_1 + \log p_1 \geq -\frac{(x-\mu_2)^2}{2\sigma_2^2} - \log\sqrt{2\pi}\sigma_2 + \log p_2$$

Setting $\sigma_1 = \sigma_2$, we obtain

$$bx + c \geq 0$$

Note: equal variances across two different categories could be a very strong assumption.

For example, the plot suggests that the *male* population has slightly bigger variance (i.e., bigger eclipse) than the *female* population.