349:Machine Learning

Fall 2024

Gaussian Mixture Models

Discriminative vs Generative Models

- Discriminative: Learn a decision boundary between two sets

  Neural Networks
  KNN
  K-Means
  Linear Discriminants
  Decision Trees
  Support Vector Machines

- Generative: Learn enough about your sets to be able to make new examples that would be set members

  Gaussian Mixture Models
  Large Language Models
  Vision
  Audio

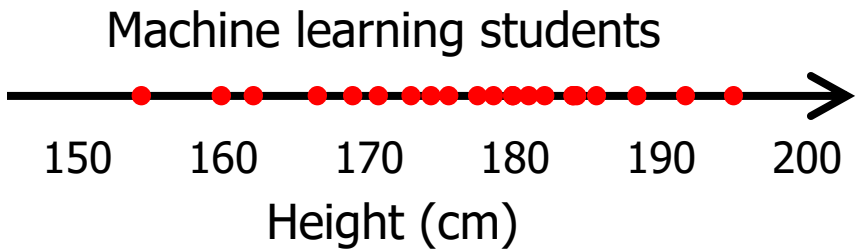
Generative Model Point of View

- Assume the data was generated from a process we can model as a probability distribution
- Learn that probability distribution
- Once learned, use the probability distribution to

  "Make" new examples

  Classify data we haven't seen before.

Non-Parametric Distribution

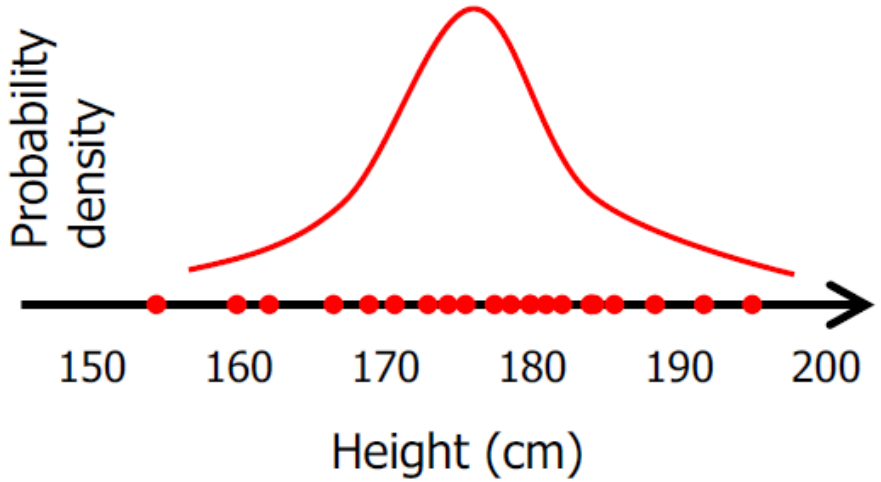


- Let's probabilistically model ML student heights.
- Ruler has 200 marks (100 to 300 cm)
- How many probabilities to learn?
- How many students in the class?
- What if the ruler is continuous?

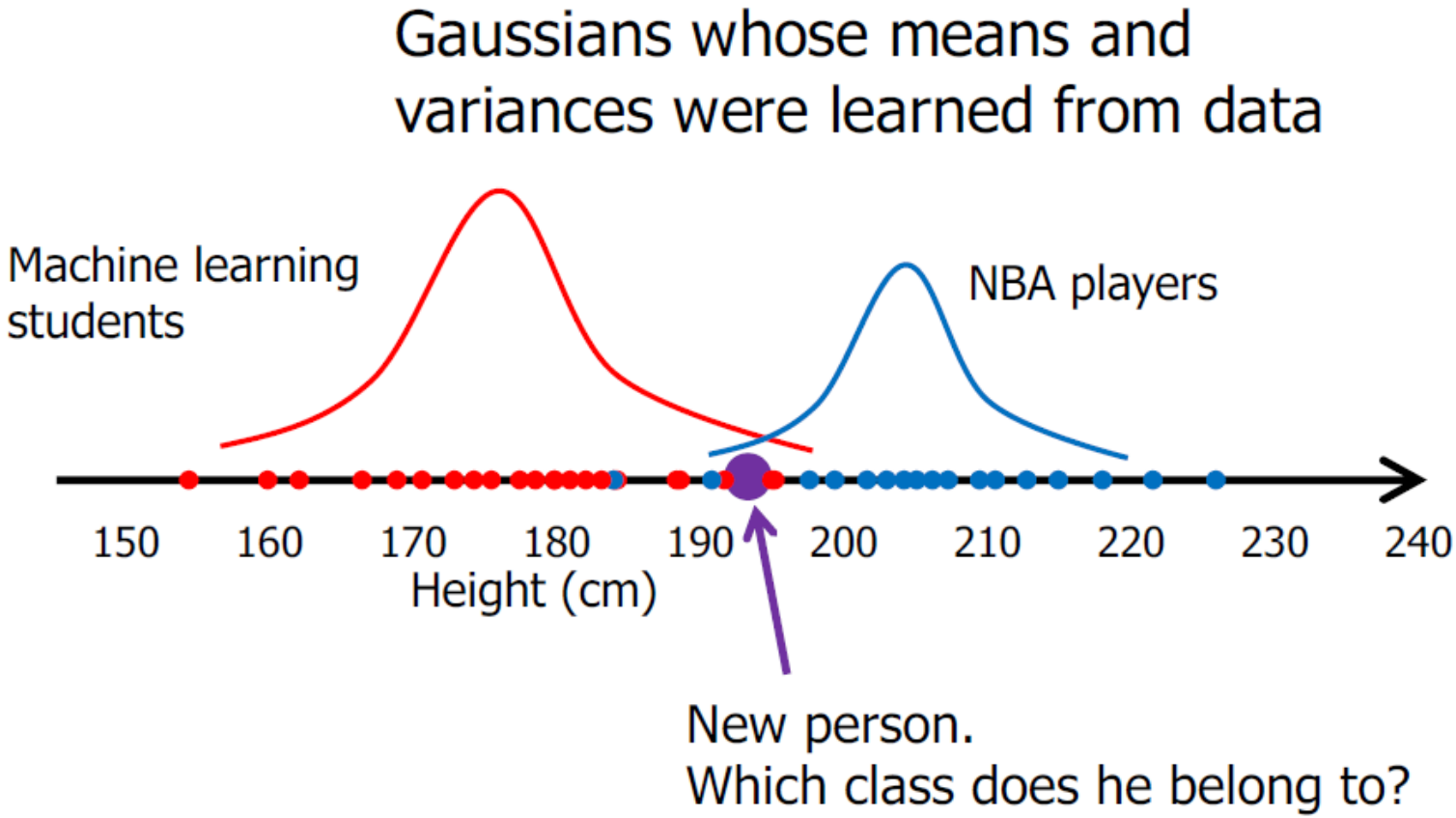
Learning a Parametric Distribution

- Pick a parametric model (e.g. Gaussian)
- Learn just a few parameter values

$$p(x \mid \Theta) \equiv \text{ prob. of x, given parameters } \Theta \text{ of a model, M}$$



Using Generative Models for Classification



Answer: the class that calls him most probable.

Learning a Gaussian Distribution

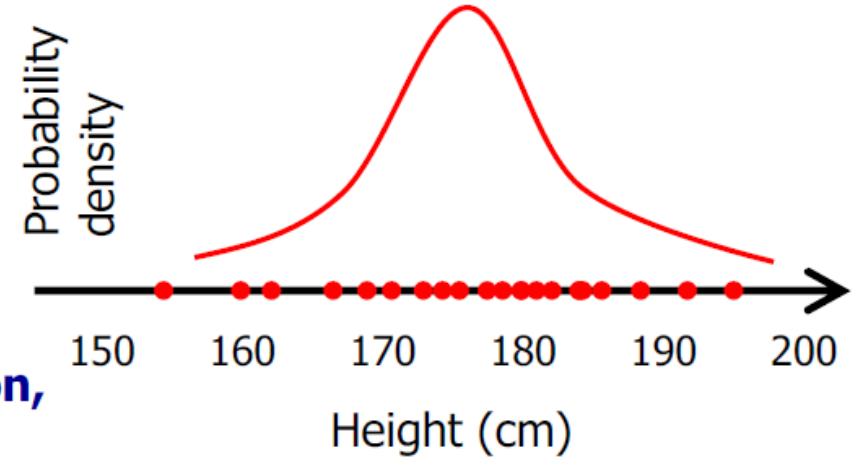
$$p(x \mid \Theta) \equiv \text{ prob. of x, given parameters } \Theta \text{ of a model, M}$$

$$\Theta \equiv \{\mu, \sigma\}$$

The parameters we must learn

$$M \equiv \frac{1}{(2\pi)^{1/2}\sigma} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

The "normal" Gaussian distribution, often denoted N, for "normal"



Goal: Find the Best Gaussian

- Hypothesis space is Gaussian distributions.
- Find parameters Θ^* that maximize the prob. of observing data $X \equiv \{x_1, \ldots x_n\}$

$$\Theta^* = \underset{\Theta}{\operatorname{argmax}}\, p(X \mid \Theta)$$

where each $\Theta \equiv \{\mu, \sigma\}$

Some Math

$$\Theta^* = \underset{\text{argmax}\,\Theta}{p(X \mid \Theta)}, \text{where each } \Theta \equiv \{\mu, \sigma\}$$

$$p(X \mid \Theta) = \prod_{i=1}^{n} p(x_i \mid \Theta)$$

...if can we assume all x_i are i.i.d.

Numbers Getting Smaller

$$p(X \mid \Theta) = \prod_{i=1}^{n} p(x_i \mid \Theta)$$

What happens as n grows? Problem?

We get underflow if n is, say, 500

$$p(X \mid \Theta) \propto \sum_{i=1}^{n} \log(p(x_i \mid \Theta)) \quad \text{solves underflow.}$$

Remember What We Are Maximizing

$$\Theta^* \equiv \underset{\text{argmax}\,\Theta}{p(X \mid \Theta)} = \sum_{i=1}^{n} \log(p(x_i \mid \Theta)) \underset{\text{argmax}\,\Theta}{)}$$

fitting the Gaussian into this...

$$\log(p(x \mid \Theta)) = \log\left(\frac{e^{\frac{-(x-\mu)^2}{2\sigma^2}}}{(2\pi)^{1/2}\sigma}\right)$$

Some Math Get You…

$$\log\left(\frac{e^{\frac{-(x-\mu)^2}{2\sigma^2}}}{(2\pi)^{1/2}\sigma}\right) = \log\left(e^{\frac{-(x-\mu)^2}{2\sigma^2}}\right) - \log\left((2\pi)^{1/2}\sigma\right)$$

$$= \boxed{\frac{-(x-\mu)^2}{2\sigma^2} - \log\sigma} - \log(2\pi)^{1/2}$$

Plug back into equation from slide 11

… Which Gives Us

$$\Theta^* \equiv \underset{\text{argmax}\,\Theta}{p(X \mid \Theta)}$$

$$= \underset{\text{argmax}\,\Theta}{\sum_{i=1}^{n} \log(p(x_i \mid \Theta))}$$

$$= \underset{\text{argmax}\,\Theta}{\sum_{i=1}^{n} \left(\frac{-(x_i - \mu)^2}{2\sigma^2} - \log\sigma \right)}$$

Maximizing Log-Likelihood

- To find best parameters, take the partial derivative with respect to parameters $\{\sigma, \mu\}$ and set to 0.

$$\Theta^* = \sum_{i=1}^{n} \left(\frac{-(x_i - \mu)^2}{2\sigma^2} - \log \sigma \right)$$

$$\operatorname{argmax} \Theta$$

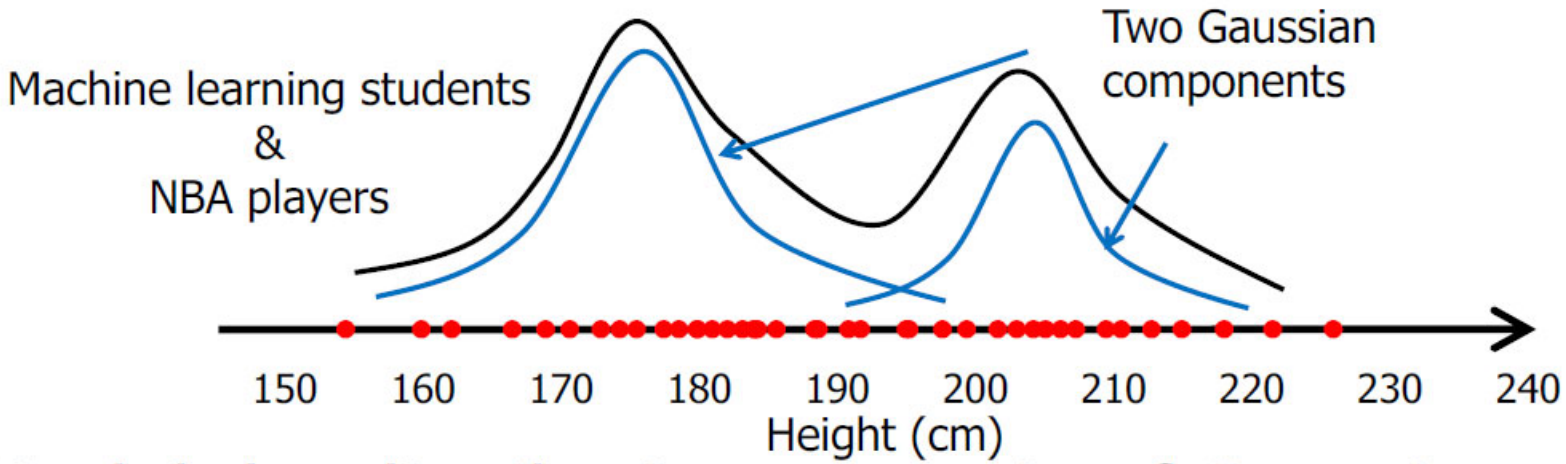
- The result is a closed-form solution

$$\mu = \frac{1}{n} \sum_{i=1}^{n} x_i \qquad \sigma^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu)^2$$

What if…

- …the data distribution can't be well represented by a single Gaussian?
- Can we model more complex distributions using multiple Gaussians?

Gaussian Mixture Model (GMM)



Model the distribution as a mix of Gaussians

$$P(x) = \sum_{j=1}^{K} P(z_j) P(x \mid z_j)$$

x is the observed value

z_j is a Boolean saying whether Gaussian j "made" x

What Are We Optimizing?

$$P(x) = \sum_{j=1}^{K} P(z_j) P(x \mid z_j)$$

Notating $P(z_j)$ as weight w_j and using the Normal (a.k.a. Gaussian) distribution $N(\mu_j, \sigma_j^2)$ gives us...

$$= \sum_{j=1}^{K} w_j N(x \mid \mu_j, \sigma_j^2) \quad \text{such that } 1 = \sum_{j=1}^{K} w_j$$

This gives 3 variables per Gaussian to optimize:

w_j, μ_j, σ_j

Bad News: No Closed Form Solution

$$\Theta^* \equiv \underset{\text{argmax}\,\Theta}{p(X \mid \Theta)} = \sum_{i=1}^{n} \underset{\text{argmax}\,\Theta}{\log(p(x_i \mid \Theta))}$$

$$= \sum_{i=1}^{n} \log \underset{\text{argmax}\,\Theta}{\left(\sum_{j=1}^{K} w_j p(x_i \mid N(\mu_j, \sigma_j^2)) \right)}$$

Expectation Maximization

- Solution: The EM algorithm
- EM updates model parameters iteratively.
- After each iteration, the likelihood the model would generate the observed data increases (or at least it doesn't decrease).
- EM algorithm always converges to a local optimum.

EM Algorithm Summary

- Initialize the parameters
- E step: calculate the likelihood a model with these parameters generated the data
- M step: Update parameters to increase the likelihood from E step
- Repeat E & M steps until convergence to a local optimum.

EM for GMM -- Initialization

- Choose the number of Gaussian components K

  K should be much less than the number of data points to avoid overfitting.

- (Randomly) select parameters for each Gaussian j: w_j, μ_j, σ_j

$$\text{...such that } 1 = \sum_{j=1}^{K} w_j$$

EM for GMM -- Expectation Step

The responsibility $\gamma_{j,n}$ of Gaussian j for observation x_n is defined as...

Data point n has value x

Gaussian j caused data point n

$$\gamma_{j,n} \equiv p(z_j \mid x_n) = \frac{p(x_n \mid z_j)p(z_j)}{p(x_n)}$$

$$= \frac{p(x_n \mid z_j)p(z_j)}{\sum_{k=1}^{K} p(z_k)p(x_n \mid z_k)} = \frac{w_j N(x_n \mid \mu_j, \sigma_j^2)}{\sum_{k=1}^{K} w_k N(x_n \mid \mu_k, \sigma_k^2)}$$

EM for GMM -- Expectation Step

Define the responsibility Γ_j of Gaussian j for all the observed data as...

$$\Gamma_j \equiv \sum_{n=1}^{N} \gamma_{j,n}$$

You can think of this as the proportion of the data explained by Gaussian j.

EM for GMM -- Maximization Step

Update our parameters as follows...

$$\text{new } w_j = \frac{\Gamma_j}{N}$$

$$\text{new } \mu_j = \frac{\sum_{i=1}^{N} \gamma_{j,i} x_i}{\Gamma_j}$$

$$\text{new } \sigma_j^2 = \frac{\sum_{i=1}^{N} \gamma_{j,i} (x_i - \mu_j)^2}{\Gamma_j}$$

What if…

- …our data isn't just scalars, but each data point has multiple dimensions?
- Can we generalize to multiple dimensions?
- We need to define a covariance matrix.

Covariance Matrix

Given d-dimensional random variable vector $\vec{\mathbf{X}} = [X_1, ..., X_d]$

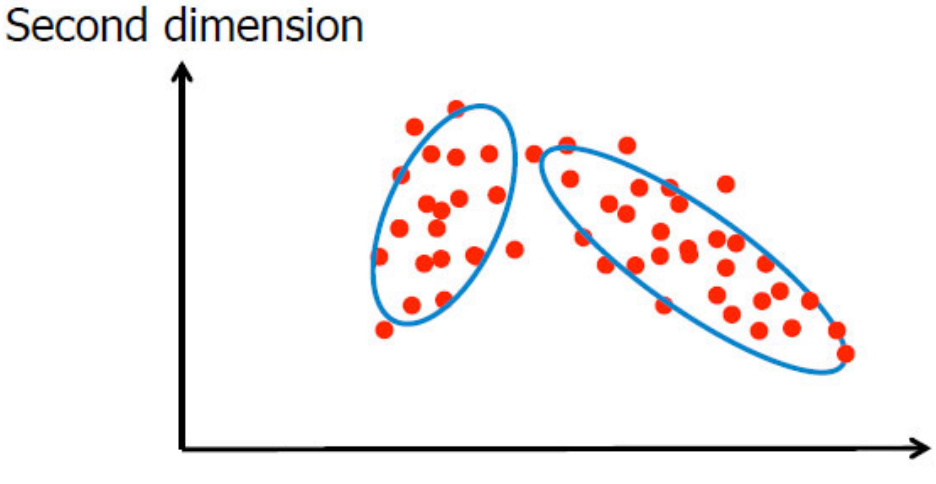
the covariance matrix denoted Σ (confusing, eh?) is defined as...

$$\Sigma \equiv \begin{bmatrix} \mathbf{E}[(X_1 - \mu_1)(X_1 - \mu_1)] & \mathbf{E}[(X_1 - \mu_1)(X_2 - \mu_2)] & \dots & \mathbf{E}[(X_1 - \mu_1)(X_d - \mu_d)] \\ \mathbf{E}[(X_2 - \mu_2)(X_1 - \mu_1)] & \mathbf{E}[(X_2 - \mu_2)(X_2 - \mu_2)] & \cdots & \mathbf{E}[(X_2 - \mu_2)(X_d - \mu_d)] \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{E}[(X_d - \mu_d)(X_1 - \mu_1)] & \mathbf{E}[(X_d - \mu_d)(X_2 - \mu_2)] & \cdots & \mathbf{E}[(X_d - \mu_d)(X_d - \mu_d)] \end{bmatrix}$$

This is a generalization of one-dimensional variance for a scalar random variable X

$$\sigma^2 = \text{var}(X) = E\left[(X - \mu)^2\right]$$

Multivariate Gaussian Mixture

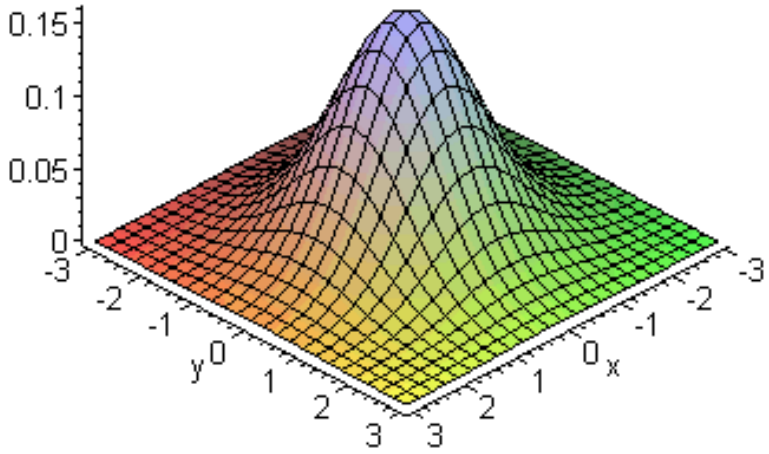


The d by d covariance matrix Σ describes the shape and orientation of an elipse.

$$P(\vec{\mathbf{X}}) = \sum_{j=1}^{K} w_j p(\vec{\mathbf{X}} \mid N(\vec{\mu}, \Sigma_j))$$

Given d dimensions and K Gaussians, how many parameters?

Bivariate Gaussian



$$f(x,y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} e^{-\frac{1}{2(1-\rho^2)}\left[\left(\frac{x-\mu_x}{\sigma_x}\right)^2 - 2\rho\left(\frac{x-\mu_x}{\sigma_x}\right)\left(\frac{y-\mu_y}{\sigma_y}\right)+\left(\frac{y-\mu_y}{\sigma_y}\right)^2\right]}$$

https://www.youtube.com/watch?v=2xHeg6n89tU

GMM Remarks

- GMM is powerful: any density function can be arbitrarily-well approximated by a GMM with enough components.
- If the number of components K is too large, data will be overfitted.
  - Likelihood increases with K.
  - Extreme case: N Gaussians for N data points, with variances $\rightarrow 0$
- How to choose K?
  - Use domain knowledge.
  - Validate through visualization.

GMM is a Soft Version of k-Means

- Similarity
  - K needs to be specified.
  - Converges to some local optima.
  - Initialization matters final results.
  - One would want to try different initializations.

- Differences
  - GMM Assigns "soft" labels to instances.
  - GMM Considers variances in addition to means.

GMM for Classification

- Given training data with multiple classes…
  1) Model the training data for each class with a GMM
  2) Classify a new point by estimating the probability each class generated the point
  3) Pick the class with the highest probability as the label.

(illustration from Leon Bottou's slides on EM)