CPSC 540: Machine Learning

Expectation Maximization and Kernel Density Estimation

Mark Schmidt

University of British Columbia

Winter 2016

Admin

- Assignment 2:
  - 2 late day to hand it in now.
  - Thursday is last possible day.
- Assignment 3:
  - Due in 2 weeks, start early.
  - Some additional hints will be added.
- Reading week:
  - No classes next week.
  - I'm talking at Robson Square 6:30pm Wednesday February 17.
- February 25:
  - Default is to not have class this day.
  - Instead go to Rich Sutton's talk in DMP 110:
    - "Reinforcement Learning And The Future of Artificial Intelligence".

Last: Multivariate Gaussian

- The multivariate normal distribution models PDF of vector x as

$$p(x|\mu,\Sigma) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}}\exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$

where $\mu \in \mathbb{R}^d$ and $\Sigma \in \mathbb{R}^{d\times d}$ and $\Sigma \succ 0$.

Last: Multivariate Gaussian

- The multivariate normal distribution models PDF of vector x as

$$p(x|\mu, \Sigma) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$

  where $\mu \in \mathbb{R}^d$ and $\Sigma \in \mathbb{R}^{d\times d}$ and $\Sigma \succ 0$.
- Closed-form MLE:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x^i, \quad \Sigma = \frac{1}{n}\sum_{i=1}^{N}(x^i-\mu)(x^i-\mu)^T.$$

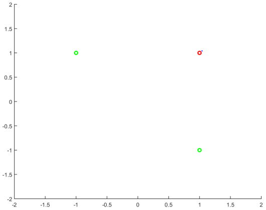
- Closed under several operations: products of PDFs, marginalization, conditioning.
- Uni-modal: probability strictly decreases as you move away from mean.
- Light-tailed': assumes all data is close to mean.
  - Not robust to outliers or data far away from mean.

Last Time: Mixture Models

- To distributions more flexible, we introduced mixture models.
- Example is mixture of Gaussians:
  - We have a set of k Gaussian distributions, each with a mean μ_c and covariance Σ_c.
  - We have a prior probability θ_c of a data point from each distribution c.

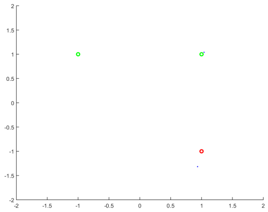
Last Time: Mixture Models

- To distributions more flexible, we introduced mixture models.
- Example is mixture of Gaussians:
  - We have a set of k Gaussian distributions, each with a mean μ_c and covariance Σ_c.
  - We have a prior probability θ_c of a data point from each distribution c.
- How a mixture of Gaussian "generates" data:
  1. Sample cluster z^i based on prior probabilities θ_c (categorical distribution).
  2. Sample example x^i based on mean μ_c and covariance Σ_c.



Last Time: Mixture Models

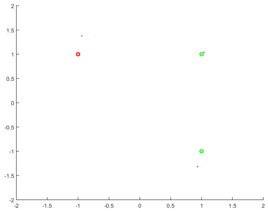
- To distributions more flexible, we introduced mixture models.
- Example is mixture of Gaussians:
  - We have a set of k Gaussian distributions, each with a mean μ_c and covariance Σ_c.
  - We have a prior probability θ_c of a data point from each distribution c.
- How a mixture of Gaussian "generates" data:
  1. Sample cluster z^i based on prior probabilities θ_c (categorical distribution).
  2. Sample example x^i based on mean μ_c and covariance Σ_c.



- Standard appraoch to fitting mixture models: expectation maximization:
  - General method for fitting models with hidden variables.

Last Time: Mixture Models

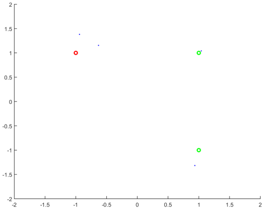
- To distributions more flexible, we introduced mixture models.
- Example is mixture of Gaussians:
  - We have a set of k Gaussian distributions, each with a mean μ_c and covariance Σ_c.
  - We have a prior probability θ_c of a data point from each distribution c.
- How a mixture of Gaussian "generates" data:
  1. Sample cluster z^i based on prior probabilities θ_c (categorical distribution).
  2. Sample example x^i based on mean μ_c and covariance Σ_c.



- Standard appraoch to fitting mixture models: expectation maximization:
  - General method for fitting models with hidden variables.

Last Time: Mixture Models

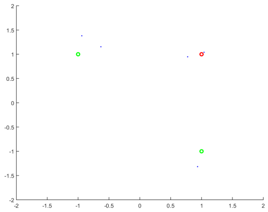
- To distributions more flexible, we introduced mixture models.
- Example is mixture of Gaussians:
  - We have a set of k Gaussian distributions, each with a mean μ_c and covariance Σ_c.
  - We have a prior probability θ_c of a data point from each distribution c.
- How a mixture of Gaussian "generates" data:
  1. Sample cluster z^i based on prior probabilities θ_c (categorical distribution).
  2. Sample example x^i based on mean μ_c and covariance Σ_c.



- Standard appraoch to fitting mixture models: expectation maximization:
  - General method for fitting models with hidden variables.

Last Time: Mixture Models

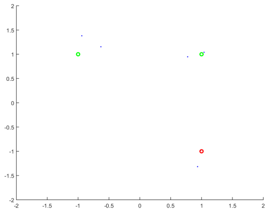
- To distributions more flexible, we introduced mixture models.
- Example is mixture of Gaussians:
  - We have a set of k Gaussian distributions, each with a mean μ_c and covariance Σ_c.
  - We have a prior probability θ_c of a data point from each distribution c.
- How a mixture of Gaussian "generates" data:
  1. Sample cluster z^i based on prior probabilities θ_c (categorical distribution).
  2. Sample example x^i based on mean μ_c and covariance Σ_c.



- Standard appraoch to fitting mixture models: expectation maximization:
  - General method for fitting models with hidden variables.

Last Time: Mixture Models

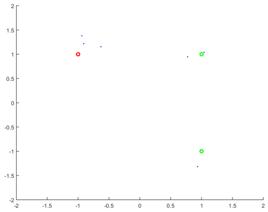
- To distributions more flexible, we introduced mixture models.
- Example is mixture of Gaussians:
  - We have a set of k Gaussian distributions, each with a mean μ_c and covariance Σ_c.
  - We have a prior probability θ_c of a data point from each distribution c.
- How a mixture of Gaussian "generates" data:
  1. Sample cluster z^i based on prior probabilities θ_c (categorical distribution).
  2. Sample example x^i based on mean μ_c and covariance Σ_c.



- Standard appraoch to fitting mixture models: expectation maximization:
  - General method for fitting models with hidden variables.

Last Time: Mixture Models

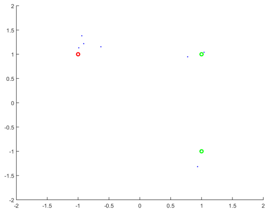
- To distributions more flexible, we introduced mixture models.
- Example is mixture of Gaussians:
  - We have a set of k Gaussian distributions, each with a mean μ_c and covariance Σ_c.
  - We have a prior probability θ_c of a data point from each distribution c.
- How a mixture of Gaussian "generates" data:
  1. Sample cluster z^i based on prior probabilities θ_c (categorical distribution).
  2. Sample example x^i based on mean μ_c and covariance Σ_c.



- Standard appraoch to fitting mixture models: expectation maximization:
  - General method for fitting models with hidden variables.

Last Time: Mixture Models

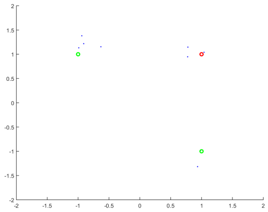
- To distributions more flexible, we introduced mixture models.
- Example is mixture of Gaussians:
  - We have a set of k Gaussian distributions, each with a mean μ_c and covariance Σ_c.
  - We have a prior probability θ_c of a data point from each distribution c.
- How a mixture of Gaussian "generates" data:
  1. Sample cluster z^i based on prior probabilities θ_c (categorical distribution).
  2. Sample example x^i based on mean μ_c and covariance Σ_c.



- Standard appraoch to fitting mixture models: expectation maximization:
  - General method for fitting models with hidden variables.

Last Time: Mixture Models

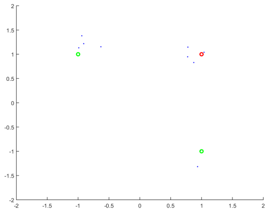
- To distributions more flexible, we introduced mixture models.
- Example is mixture of Gaussians:
  - We have a set of k Gaussian distributions, each with a mean μ_c and covariance Σ_c.
  - We have a prior probability θ_c of a data point from each distribution c.
- How a mixture of Gaussian "generates" data:
  1. Sample cluster z^i based on prior probabilities θ_c (categorical distribution).
  2. Sample example x^i based on mean μ_c and covariance Σ_c.



- Standard appraoch to fitting mixture models: expectation maximization:
  - General method for fitting models with hidden variables.

Last Time: Mixture Models

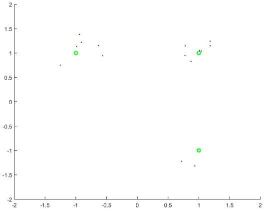
- To distributions more flexible, we introduced mixture models.
- Example is mixture of Gaussians:
  - We have a set of k Gaussian distributions, each with a mean μ_c and covariance Σ_c.
  - We have a prior probability θ_c of a data point from each distribution c.
- How a mixture of Gaussian "generates" data:
  1. Sample cluster z^i based on prior probabilities θ_c (categorical distribution).
  2. Sample example x^i based on mean μ_c and covariance Σ_c.



- Standard appraoch to fitting mixture models: expectation maximization:
  - General method for fitting models with hidden variables.

Last Time: Mixture Models

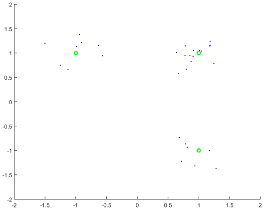
- To distributions more flexible, we introduced mixture models.
- Example is mixture of Gaussians:
  - We have a set of k Gaussian distributions, each with a mean μ_c and covariance Σ_c.
  - We have a prior probability θ_c of a data point from each distribution c.
- How a mixture of Gaussian "generates" data:
  1. Sample cluster z^i based on prior probabilities θ_c (categorical distribution).
  2. Sample example x^i based on mean μ_c and covariance Σ_c.



- Standard appraoch to fitting mixture models: expectation maximization:
  - General method for fitting models with hidden variables.

Last Time: Mixture Models

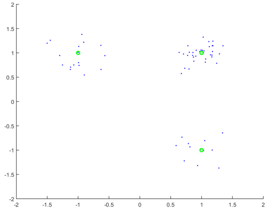
- To distributions more flexible, we introduced mixture models.
- Example is mixture of Gaussians:
  - We have a set of k Gaussian distributions, each with a mean μ_c and covariance Σ_c.
  - We have a prior probability θ_c of a data point from each distribution c.
- How a mixture of Gaussian "generates" data:
  1. Sample cluster z^i based on prior probabilities θ_c (categorical distribution).
  2. Sample example x^i based on mean μ_c and covariance Σ_c.



- Standard appraoch to fitting mixture models: expectation maximization:
  - General method for fitting models with hidden variables.

Last Time: Mixture Models

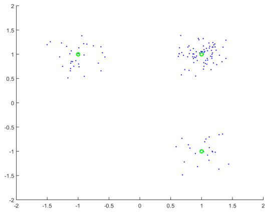
- To distributions more flexible, we introduced mixture models.
- Example is mixture of Gaussians:
  - We have a set of k Gaussian distributions, each with a mean μ_c and covariance Σ_c.
  - We have a prior probability θ_c of a data point from each distribution c.
- How a mixture of Gaussian "generates" data:
  1. Sample cluster z^i based on prior probabilities θ_c (categorical distribution).
  2. Sample example x^i based on mean μ_c and covariance Σ_c.



- Standard appraoch to fitting mixture models: expectation maximization:
  - General method for fitting models with hidden variables.

Last Time: Mixture Models

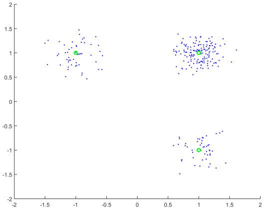
- To distributions more flexible, we introduced mixture models.
- Example is mixture of Gaussians:
  - We have a set of k Gaussian distributions, each with a mean μ_c and covariance Σ_c.
  - We have a prior probability θ_c of a data point from each distribution c.
- How a mixture of Gaussian "generates" data:
  1. Sample cluster z^i based on prior probabilities θ_c (categorical distribution).
  2. Sample example x^i based on mean μ_c and covariance Σ_c.



- Standard appraoch to fitting mixture models: expectation maximization:
  - General method for fitting models with hidden variables.

Last Time: Mixture Models

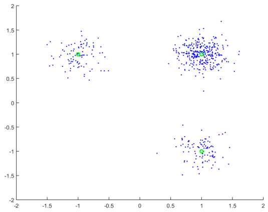
- To distributions more flexible, we introduced mixture models.
- Example is mixture of Gaussians:
  - We have a set of k Gaussian distributions, each with a mean μ_c and covariance Σ_c.
  - We have a prior probability θ_c of a data point from each distribution c.
- How a mixture of Gaussian "generates" data:
  1. Sample cluster z^i based on prior probabilities θ_c (categorical distribution).
  2. Sample example x^i based on mean μ_c and covariance Σ_c.



- Standard appraoch to fitting mixture models: expectation maximization:
  - General method for fitting models with hidden variables.

Last Time: Mixture Models

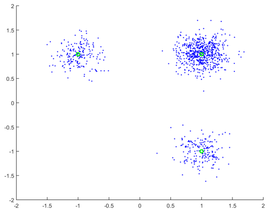
- To distributions more flexible, we introduced mixture models.
- Example is mixture of Gaussians:
  - We have a set of k Gaussian distributions, each with a mean μ_c and covariance Σ_c.
  - We have a prior probability θ_c of a data point from each distribution c.
- How a mixture of Gaussian "generates" data:
  1. Sample cluster z^i based on prior probabilities θ_c (categorical distribution).
  2. Sample example x^i based on mean μ_c and covariance Σ_c.



- Standard appraoch to fitting mixture models: expectation maximization:
  - General method for fitting models with hidden variables.

Last Time: Mixture Models

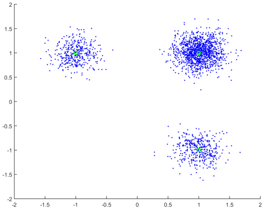
- To distributions more flexible, we introduced mixture models.
- Example is mixture of Gaussians:
  - We have a set of k Gaussian distributions, each with a mean μ_c and covariance Σ_c.
  - We have a prior probability θ_c of a data point from each distribution c.
- How a mixture of Gaussian "generates" data:
  1. Sample cluster z^i based on prior probabilities θ_c (categorical distribution).
  2. Sample example x^i based on mean μ_c and covariance Σ_c.



- Standard appraoch to fitting mixture models: expectation maximization:
  - General method for fitting models with hidden variables.

Last Time: Mixture Models

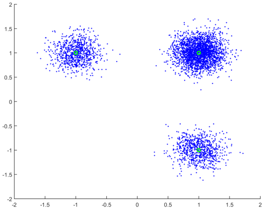
- To distributions more flexible, we introduced mixture models.
- Example is mixture of Gaussians:
  - We have a set of k Gaussian distributions, each with a mean μ_c and covariance Σ_c.
  - We have a prior probability θ_c of a data point from each distribution c.
- How a mixture of Gaussian "generates" data:
  1. Sample cluster z^i based on prior probabilities θ_c (categorical distribution).
  2. Sample example x^i based on mean μ_c and covariance Σ_c.



- Standard appraoch to fitting mixture models: expectation maximization:
  - General method for fitting models with hidden variables.

Last Time: Mixture Models

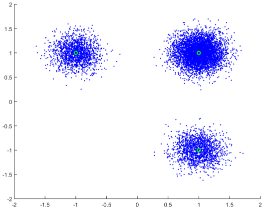
- To distributions more flexible, we introduced mixture models.
- Example is mixture of Gaussians:
  - We have a set of k Gaussian distributions, each with a mean μ_c and covariance Σ_c.
  - We have a prior probability θ_c of a data point from each distribution c.
- How a mixture of Gaussian "generates" data:
  1. Sample cluster z^i based on prior probabilities θ_c (categorical distribution).
  2. Sample example x^i based on mean μ_c and covariance Σ_c.



- Standard appraoch to fitting mixture models: expectation maximization:
  - General method for fitting models with hidden variables.

Last Time: Mixture Models

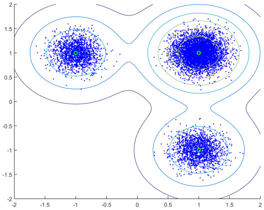
- To distributions more flexible, we introduced mixture models.
- Example is mixture of Gaussians:
  - We have a set of k Gaussian distributions, each with a mean μ_c and covariance Σ_c.
  - We have a prior probability θ_c of a data point from each distribution c.
- How a mixture of Gaussian "generates" data:
  1. Sample cluster z^i based on prior probabilities θ_c (categorical distribution).
  2. Sample example x^i based on mean μ_c and covariance Σ_c.



- Standard appraoch to fitting mixture models: expectation maximization:
  - General method for fitting models with hidden variables.

Last Time: Mixture Models

- To distributions more flexible, we introduced mixture models.
- Example is mixture of Gaussians:
  - We have a set of k Gaussian distributions, each with a mean μ_c and covariance Σ_c.
  - We have a prior probability θ_c of a data point from each distribution c.
- How a mixture of Gaussian "generates" data:
  1. Sample cluster z^i based on prior probabilities θ_c (categorical distribution).
  2. Sample example x^i based on mean μ_c and covariance Σ_c.



- Standard appraoch to fitting mixture models: expectation maximization:
  - General method for fitting models with hidden variables.

Last Time: Learning with Hidden Values

- We often want to learn when some variables unobserved/missing/hidden/latent.
- For example, we could have a dataset

$$X = \begin{bmatrix} N & 33 & 5 \\ F & 10 & 1 \\ F & ? & 2 \\ M & 22 & 0 \end{bmatrix}, y = \begin{bmatrix} -1 \\ +1 \\ -1 \\ ? \end{bmatrix}.$$

- Missing values are very common in real datasets.

Last Time: Learning with Hidden Values

- We often want to learn when some variables unobserved/missing/hidden/latent.
- For example, we could have a dataset

$$X = \begin{bmatrix} N & 33 & 5 \\ F & 10 & 1 \\ F & ? & 2 \\ M & 22 & 0 \end{bmatrix}, y = \begin{bmatrix} -1 \\ +1 \\ -1 \\ ? \end{bmatrix}.$$

- Missing values are very common in real datasets.
- We'll focus on data that is missing at random (MAR):
  - The fact that is ? is missing does not not depend on missing value.
- In the case of mixture models, we'll treat the clusters z^i as missing values.

More Motivation for EM: Semi-Supervised Learning

- Important special case of hidden values is semi-supervised learning.
- Motivation:
  - Getting labels is often expensive.
  - But getting unlabeled examples is usually cheap.

More Motivation for EM: Semi-Supervised Learning

- Important special case of hidden values is semi-supervised learning.
- Motivation:
  - Getting labels is often expensive.
  - But getting unlabeled examples is usually cheap.
- Can we train with labeled data (X, y) and unlabeled data $\tilde{X}$?

$$X = \begin{bmatrix} \quad \end{bmatrix}, y = \begin{bmatrix} \end{bmatrix},$$

$$\tilde{X} = \begin{bmatrix} \quad \end{bmatrix}, \tilde{y} = \begin{bmatrix} ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix},$$

More Motivation for EM: Semi-Supervised Learning

- Important special case of hidden values is semi-supervised learning.
- Motivation:
  - Getting labels is often expensive.
  - But getting unlabeled examples is usually cheap.
- Can we train with labeled data (X, y) and unlabeled data $\tilde{X}$?

$$X = \begin{bmatrix} \quad & \quad & \quad \end{bmatrix}, y = \begin{bmatrix} \; \end{bmatrix},$$

$$\tilde{X} = \begin{bmatrix} \quad & \quad & \quad \\ \\ \\ \\ \end{bmatrix}, \tilde{y} = \begin{bmatrix} ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix},$$

- If these are IID samples, then $\tilde{y}$ values are MAR.
- Classic approach: use generative classifier and apply EM.

Probabilistic Approach to Learing with Hidden Variables

- Let's use O as observed variables and H as our hidden variables.
  - For semi-supervised learning, $O = \{X, y, \tilde{X}\}$ and $H = \{\tilde{y}\}$.
  - We'll use Θ as the parameters we want to optimize.

Probabilistic Approach to Learing with Hidden Variables

- Let's use O as observed variables and H as our hidden variables.
  - For semi-supervised learning, $O = \{X, y, \tilde{X}\}$ and $H = \{\tilde{y}\}$.
  - We'll use Θ as the parameters we want to optimize.
- Since we don't observe H, we'll focus on the probability of observed O,

$$p(O|\Theta) = \sum_H p(O, H|\Theta), \quad \text{(by marginalization rule } p(a) = \sum_b p(a,b)\text{)},$$

  where we sum (or integrate) over all possible hidden values.
- This is nice because we only need likelihood of "complete" data (O, h).

Probabilistic Approach to Learing with Hidden Variables

- Let's use O as observed variables and H as our hidden variables.
  - For semi-supervised learning, $O = \{X, y, \tilde{X}\}$ and $H = \{\tilde{y}\}$.
  - We'll use Θ as the parameters we want to optimize.
- Since we don't observe H, we'll focus on the probability of observed O,

$$p(O|\Theta) = \sum_H p(O, H|\Theta), \quad \text{(by marginalization rule } p(a) = \sum_b p(a, b)\text{)},$$

  where we sum (or integrate) over all possible hidden values.
- This is nice because we only need likelihood of "complete" data (O, h).
- But the log-likelihood has the form

$$-\log p(O|\Theta) = -\log\left(\sum_h p(O, H|\Theta)\right),$$

  which usually is not convex (due to sum to sum inside the log).

Probabilistic Approach to Learing with Hidden Variables

- Let's use O as observed variables and H as our hidden variables.
  - For semi-supervised learning, $O = \{X, y, \tilde{X}\}$ and $H = \{\tilde{y}\}$.
  - We'll use Θ as the parameters we want to optimize.
- Since we don't observe H, we'll focus on the probability of observed O,

$$p(O|\Theta) = \sum_H p(O, H|\Theta), \quad \text{(by marginalization rule } p(a) = \sum_b p(a, b)\text{)},$$

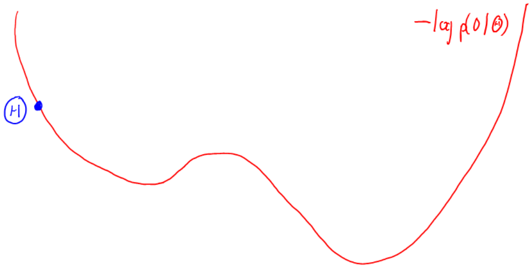
  where we sum (or integrate) over all possible hidden values.
- This is nice because we only need likelihood of "complete" data (O, h).
- But the log-likelihood has the form

$$-\log p(O|\Theta) = -\log\left(\sum_h p(O, H|\Theta)\right),$$

  which usually is not convex (due to sum to sum inside the log).
- Even if $-\log p(O, H|\Theta)$ is "nice" (closed-form, convex, etc.),
  maximizing the likelihood is typically hard

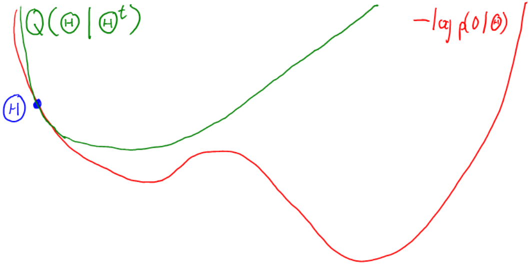
Expectation Maximization as Bound Optimization

- Expectation maximization is a bound-optimization method:
  - At each iteration we optimize a bound on the function.



Expectation Maximization as Bound Optimization

- Expectation maximization is a bound-optimization method:
  - At each iteration we optimize a bound on the function.



- In gradient descent, our bound came from Lipschitz-continuity of the gradient.

Expectation Maximization as Bound Optimization

- Expectation maximization is a bound-optimization method:
  - At each iteration we optimize a bound on the function.

- In gradient descent, our bound came from Lipschitz-continuity of the gradient.
- In EM, our bound comes from expectation over hidden variables.
  - Bound will typically not be a quadratic.

Probabilistic Approach to Learing with Hidden Variables

- For example, in semi-supervised learning our complete-data NLL is

$$-\log p(\underbrace{X, y, \tilde{X}, \tilde{y}}_{O,H}|\Theta) = -\log\left(\left(\prod_{i=1}^{n} p(x^i, y^i|\Theta)\right)\left(\prod_{i=1}^{t} p(\tilde{x}^i, \tilde{y}^i|\Theta)\right)\right)$$

$$= \underbrace{-\sum_{i=1}^{n}\log p(x^i, y^i|\Theta)}_{\text{labeled}} - \underbrace{\sum_{i=1}^{t}\log p(\tilde{x}^i, \tilde{y}^i|\Theta)}_{\text{unlabeled with guesses } \tilde{y}^i},$$

which is convex if $-\log p(x, y|\Theta)$ is convex.

Probabilistic Approach to Learing with Hidden Variables

- For example, in semi-supervised learning our complete-data NLL is

$$-\log p(\underbrace{X, y, \tilde{X}, \tilde{y}}_{O,H}|\Theta) = -\log\left(\left(\prod_{i=1}^{n} p(x^i, y^i|\Theta)\right)\left(\prod_{i=1}^{t} p(\tilde{x}^i, \tilde{y}^i|\Theta)\right)\right)$$

$$= \underbrace{-\sum_{i=1}^{n}\log p(x^i, y^i|\Theta)}_{\text{labeled}} \underbrace{- \sum_{i=1}^{t}\log p(\tilde{x}^i, \tilde{y}^i|\Theta)}_{\text{unlabeled with guesses } \tilde{y}^i},$$

  which is convex if $-\log p(x, y|\Theta)$ is convex.
- But since we don't observe $\tilde{y}^i$, our observed-data NLL is

$$-\log(p(X, y, \tilde{X})|\Theta) = -\log\left(\sum_{\tilde{y}^1}\sum_{\tilde{y}^2}\cdots\sum_{\tilde{y}^t}\prod_{i=1}^{n} p(x^i, y^i|\Theta)\prod_{i=1}^{t} p(\tilde{x}^i, \tilde{y}^i|\Theta)\right)$$

$$= -\sum_{i=1}^{n}\log p(x^i, y^i|\Theta) - \sum_{i=1}^{t}\log\left(\sum_{\tilde{y}^i} p(\tilde{x}^i, \tilde{y}^i|\Theta)\right),$$

"Hard" Expectation Maximization and K-Means

- A heuristic approach to repeat "hard" EM:
  1. Imputation: replace ? with their most likely values.
  2. Optimization: fit model with these values.
- Nice because the second step uses "nice" complete-data NLL.

"Hard" Expectation Maximization and K-Means

- A heuristic approach to repeat "hard" EM:
  1. Imputation: replace ? with their most likely values.
  2. Optimization: fit model with these values.
- Nice because the second step uses "nice" complete-data NLL.
- With mixture of Gaussians this would give:
  1. Imputation: for each x^i, find most likely cluster z^i.
  2. Optimization: update cluster means μ_c and covariances Σ_c.
- This is k-means clustering when covariances are shared across clusters.

"Hard" Expectation Maximization and K-Means

- A heuristic approach to repeat "hard" EM:
  1. Imputation: replace ? with their most likely values.
  2. Optimization: fit model with these values.
- Nice because the second step uses "nice" complete-data NLL.
- With mixture of Gaussians this would give:
  1. Imputation: for each x^i, find most likely cluster z^i.
  2. Optimization: update cluster means μ_c and covariances Σ_c.
- This is k-means clustering when covariances are shared across clusters.
- Instead of single assignment, EM takes combination of all possible hidden values,

$$-\log p(O|\Theta) = -\log\left(\sum_H p(O,H|\Theta)\right) \approx -\sum_H \alpha_H \log p(O,H|\Theta).$$

- The weights α_h are set so that minimizing approximation decreases $-\log p(O)$.

Expectation Maximization (EM)

- EM is local optimizer for cases where minimizing $-\log p(O, H)$ is easy.
- Key idea: start with some Θ^0 and set Θ^{t+1} to minimize upper bound

$$-\log p(O|\Theta) \leq -\sum_H \alpha_H^t \log p(O, H|\Theta) + \text{const.},$$

where using $\alpha_H^t = p(H|O, \Theta^t)$ guarantees that Θ^{t+1} decrease $-\log p(O|\Theta)$.

Expectation Maximization (EM)

- EM is local optimizer for cases where minimizing $-\log p(O, H)$ is easy.
- Key idea: start with some Θ^0 and set Θ^{t+1} to minimize upper bound

$$-\log p(O|\Theta) \leq -\sum_H \alpha_H^t \log p(O, H|\Theta) + \text{const.},$$

  where using $\alpha_H^t = p(H|O, \Theta^t)$ guarantees that Θ^{t+1} decrease $-\log p(O|\Theta)$.
- This is typically written as two steps:
  1. E-step: Define expectation of complete-data log-likelihood given Θ^t,

$$\begin{aligned} Q(\Theta|\Theta^t) &= \mathbb{E}_{H|O,\Theta^t}[\log p(O, H|\Theta)] \\ &= \sum_H \underbrace{p(H|O, \Theta^t)}_{\text{fixed weight}} \underbrace{\log p(O, H|\Theta)}_{\text{nice term}}, \end{aligned}$$

     which is a weighted version of the "nice" $\log p(O, H)$ values.

Expectation Maximization (EM)

- EM is local optimizer for cases where minimizing $-\log p(O, H)$ is easy.
- Key idea: start with some Θ^0 and set Θ^{t+1} to minimize upper bound

$$-\log p(O|\Theta) \leq -\sum_H \alpha_H^t \log p(O, H|\Theta) + \text{const.},$$

  where using $\alpha_H^t = p(H|O, \Theta^t)$ guarantees that Θ^{t+1} decrease $-\log p(O|\Theta)$.
- This is typically written as two steps:
  1. E-step: Define expectation of complete-data log-likelihood given Θ^t,

$$Q(\Theta|\Theta^t) = \mathbb{E}_{H|O,\Theta^t}[\log p(O, H|\Theta)] = \sum_H \underbrace{p(H|O, \Theta^t)}_{\text{fixed weight}} \underbrace{\log p(O, H|\Theta)}_{\text{nice term}},$$

     which is a weighted version of the "nice" $\log p(O, H)$ values.
  2. M-step: Maximize this expectation,

$$\Theta^{t+1} = \underset{\Theta}{\text{argmax}}\, Q(\Theta|\Theta^t).$$

Convergence Properties of Expectation Maximization

- We can show that EM does not decrease likelihood.

Convergence Properties of Expectation Maximization

- We can show that EM does not decrease likelihood.
- In fact we have the slightly stronger

$$\log p(O|\Theta^{t+1}) - \log p(O|\Theta^t) \geq Q(\Theta^{t+1}|\Theta^t) - Q(\Theta^t|\Theta^t),$$

  that increase in likelihood is at least as big as increase in bound.
- Does this imply convergence?

Convergence Properties of Expectation Maximization

- We can show that EM does not decrease likelihood.
- In fact we have the slightly stronger

$$\log p(O|\Theta^{t+1}) - \log p(O|\Theta^t) \geq Q(\Theta^{t+1}|\Theta^t) - Q(\Theta^t|\Theta^t),$$

  that increase in likelihood is at least as big as increase in bound.
- Does this imply convergence?
  - Yes, if likelihood is bounded above.
- Does this imply convergence to a stationary point?

Convergence Properties of Expectation Maximization

- We can show that EM does not decrease likelihood.
- In fact we have the slightly stronger

$$\log p(O|\Theta^{t+1}) - \log p(O|\Theta^{t}) \geq Q(\Theta^{t+1}|\Theta^{t}) - Q(\Theta^{t}|\Theta^{t}),$$

  that increase in likelihood is at least as big as increase in bound.
- Does this imply convergence?
  - Yes, if likelihood is bounded above.
- Does this imply convergence to a stationary point?
  - No, although many papers imply that it does.
    - Could have maximum of 3 and objective values of $1, 1 + 1/2, 1 + 1/2 + 1/4, \ldots$
  - Might just asymptotically make less and less progress.
- Almost nothing is known about rate of convergence.

Bound on Progress of Expectation Maximization

The iterations of the EM algorithm satisfy

$$\log p(O|\Theta^{t+1}) - \log p(O|\Theta^{t}) \geq Q(\Theta^{t+1}|\Theta^{t}) - Q(\Theta^{t}|\Theta^{t}),$$

Bound on Progress of Expectation Maximization

The iterations of the EM algorithm satisfy

$$\log p(O|\Theta^{t+1}) - \log p(O|\Theta^t) \geq Q(\Theta^{t+1}|\Theta^t) - Q(\Theta^t|\Theta^t),$$

- Key idea of proof: $-\log(z)$ a convex function, and for a convex f we have

$$f\left(\sum_i \alpha_i z_i\right) \leq \sum_i \alpha_i f(z_i), \text{ for } \alpha_i \geq 0 \text{ and } \sum_i \alpha_i = 1.$$

  Generalizes $f(\alpha z_1 + (1-\alpha)z_2) \leq \alpha f(z_1) + (1-\alpha)z_2)$ to convex combinations.

Bound on Progress of Expectation Maximization

The iterations of the EM algorithm satisfy

$$\log p(O|\Theta^{t+1}) - \log p(O|\Theta^t) \geq Q(\Theta^{t+1}|\Theta^t) - Q(\Theta^t|\Theta^t),$$

- Key idea of proof: $-\log(z)$ a convex function, and for a convex f we have

$$f\left(\sum_i \alpha_i z_i\right) \leq \sum_i \alpha_i f(z_i), \text{ for } \alpha_i \geq 0 \text{ and } \sum_i \alpha_i = 1.$$

  Generalizes $f(\alpha z_1 + (1-\alpha)z_2) \leq \alpha f(z_1) + (1-\alpha)z_2)$ to convex combinations.
- Proof:

$$\begin{aligned} -\log p(O|\Theta) &= -\log(\sum_h p(O,H|\Theta)) \\ &= -\log\left(\sum_H \alpha_H \frac{p(O,H|\Theta)}{\alpha_H}\right) \\ &\leq -\sum_H \alpha_H \log\left(\frac{p(O,H|\Theta)}{\alpha_H}\right). \end{aligned}$$

Bound on Progress of Expectation Maximization

The iterations of the EM algorithm satisfy

$$\log p(O|\Theta^{t+1}) - \log p(O|\Theta^t) \geq Q(\Theta^{t+1}|\Theta^t) - Q(\Theta^t|\Theta^t),$$

- Continuing and using $\alpha_H = p(H|O,\Theta^t)$ we have

$$\begin{aligned}
-\log p(O|\Theta) &\leq -\sum_H \alpha_H \log\left(\frac{p(O,H|\Theta)}{\alpha_H}\right) \\
&= \underbrace{-\sum_H \alpha_H \log p(O,H|\Theta)}_{Q(\Theta|\Theta^t)} + \underbrace{\sum_H \alpha_H \log \alpha_H}_{\text{negative entropy}} \\
&= -Q(\Theta|\Theta^t) - \text{entropy}(\alpha).
\end{aligned}$$

Bound on Progress of Expectation Maximization

The iterations of the EM algorithm satisfy

$$\log p(O|\Theta^{t+1}) - \log p(O|\Theta^t) \geq Q(\Theta^{t+1}|\Theta^t) - Q(\Theta^t|\Theta^t),$$

- We've bounded $p(O|\Theta^{t+1})$, now we need to bound $p(O|\Theta^t)$,

$$p(O|\Theta^t) = p(O, H|\Theta^t)/p(H|O, \Theta^t) \qquad \text{(using } p(a|b) = p(a,b)/p(b)\text{)}$$

Bound on Progress of Expectation Maximization

The iterations of the EM algorithm satisfy

$$\log p(O|\Theta^{t+1}) - \log p(O|\Theta^t) \geq Q(\Theta^{t+1}|\Theta^t) - Q(\Theta^t|\Theta^t),$$

- We've bounded $p(O|\Theta^{t+1})$, now we need to bound $p(O|\Theta^t)$,

$$p(O|\Theta^t) = p(O, H|\Theta^t)/p(H|O, \Theta^t) \qquad \text{(using } p(a|b) = p(a, b)/p(b)\text{)}$$

- Taking expectation of logarithm of both sides gives

$$\mathbb{E}_{\alpha_H}[\log p(O|\Theta^t)] = \mathbb{E}_{\alpha_H}[\log p(O, H|\Theta^t) - \log p(H|O, \Theta^t)]$$

$$\sum_H \alpha_H \log p(O|\Theta^t) = \sum_H \alpha_H \log p(O, H|\Theta^t) - \sum_H \alpha_H \log p(H|O, \Theta^t).$$

Bound on Progress of Expectation Maximization

The iterations of the EM algorithm satisfy

$$\log p(O|\Theta^{t+1}) - \log p(O|\Theta^t) \geq Q(\Theta^{t+1}|\Theta^t) - Q(\Theta^t|\Theta^t),$$

- We've bounded $p(O|\Theta^{t+1})$, now we need to bound $p(O|\Theta^t)$,

$$p(O|\Theta^t) = p(O, H|\Theta^t)/p(H|O, \Theta^t) \qquad \text{(using } p(a|b) = p(a,b)/p(b)\text{)}$$

- Taking expectation of logarithm of both sides gives

$$\mathbb{E}_{\alpha_H}[\log p(O|\Theta^t)] = \mathbb{E}_{\alpha_H}[\log p(O, H|\Theta^t) - \log p(H|O, \Theta^t)]$$

$$\sum_H \alpha_H \log p(O|\Theta^t) = \sum_H \alpha_H \log p(O, H|\Theta^t) - \sum_H \alpha_H \log p(H|O, \Theta^t).$$

- And using the definition of α_h we have

$$\log p(O|\Theta^t) \underbrace{\sum_H \alpha_H}_{=1} = Q(\Theta^t|\Theta^t) + \text{entropy}(\alpha).$$

Bound on Progress of Expectation Maximization

The iterations of the EM algorithm satisfy

$$\log p(O|\Theta^{t+1}) - \log p(O|\Theta^t) \geq Q(\Theta^{t+1}|\Theta^t) - Q(\Theta^t|\Theta^t),$$

- Thus we have the two bounds

$$\log p(O|\Theta) \geq Q(\Theta|\Theta^t) + \text{entropy}(\alpha)$$
$$\log p(O|\Theta^t) = Q(\Theta^t|\Theta^t) + \text{entropy}(\alpha).$$

Using $\Theta = \Theta^{t+1}$ and subtracting the second from the first gives the result.

Bound on Progress of Expectation Maximization

The iterations of the EM algorithm satisfy

$$\log p(O|\Theta^{t+1}) - \log p(O|\Theta^t) \geq Q(\Theta^{t+1}|\Theta^t) - Q(\Theta^t|\Theta^t),$$

- Thus we have the two bounds

$$\log p(O|\Theta) \geq Q(\Theta|\Theta^t) + \text{entropy}(\alpha)$$
$$\log p(O|\Theta^t) = Q(\Theta^t|\Theta^t) + \text{entropy}(\alpha).$$

  Using $\Theta = \Theta^{t+1}$ and subtracting the second from the first gives the result.
- Notes:
  - Bound says we can choose any Θ that increases Q over Θ^t.
    - Approximate M-steps are ok.
  - Entropy of hidden variables gives tightness of bound Q:
    - Low entropy (hidden values are predictable): EM bound is tight.
    - High entropy (hidden values are unpredictable): EM bound is loose.

Expectation Maximization for Mixture of Gaussians

- The classic mixture of Gaussians model uses a PDF of the form

$$p(x^i|\Theta) = \sum_{c=1}^{k} p(z^i = c|\Theta)p(x^i|z^i = c, \Theta),$$

Expectation Maximization for Mixture of Gaussians

- The classic mixture of Gaussians model uses a PDF of the form

$$p(x^i|\Theta) = \sum_{c=1}^{k} p(z^i = c|\Theta)p(x^i|z^i = c, \Theta),$$

where each mixture component is a multivariate Gaussian,

$$p(x^i|z^i = c, \Theta) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_c|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x^i - \mu_c)^T\Sigma_c^{-1}(x^i - \mu_c)\right),$$

Expectation Maximization for Mixture of Gaussians

- The classic mixture of Gaussians model uses a PDF of the form

$$p(x^i|\Theta) = \sum_{c=1}^{k} p(z^i = c|\Theta)p(x^i|z^i = c, \Theta),$$

where each mixture component is a multivariate Gaussian,

$$p(x^i|z^i = c, \Theta) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_c|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x^i - \mu_c)^T\Sigma_c^{-1}(x^i - \mu_c)\right),$$

and we model the mixture probabilities as categorical,

$$p(z^i = c|\Theta) = \theta_c.$$

Expectation Maximization for Mixture of Gaussians

- The classic mixture of Gaussians model uses a PDF of the form

$$p(x^i|\Theta) = \sum_{c=1}^{k} p(z^i = c|\Theta)p(x^i|z^i = c, \Theta),$$

where each mixture component is a multivariate Gaussian,

$$p(x^i|z^i = c, \Theta) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_c|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x^i - \mu_c)^T \Sigma_c^{-1}(x^i - \mu_c)\right),$$

and we model the mixture probabilities as categorical,

$$p(z^i = c|\Theta) = \theta_c.$$

- Finding the optimal parameter $\Theta = \{\theta_c, \mu_c, \Sigma_c\}_{c=1}^{k}$ is NP-hard.
  - But EM updates for improving parameters use analytic form of Gaussian MLE.

Expectation Maximization for Mixture of Gaussians

- The weights from the E-step are the responsibilitites,

$$r_c^i \triangleq p(z^i = c | x^i, \Theta^t) = \frac{p(x^i | z^i = c, \Theta^t) p(z^i = c, \Theta^t)}{\sum_{c'=1}^{k} p(x^i | z^i = c', \Theta^t) p(z^i = c', \Theta^t)}$$

Expectation Maximization for Mixture of Gaussians

- The weights from the E-step are the responsibilitites,

$$r_c^i \triangleq p(z^i = c|x^i, \Theta^t) = \frac{p(x^i|z^i = c, \Theta^t)p(z^i = c, \Theta^t)}{\sum_{c'=1}^k p(x^i|z^i = c', \Theta^t)p(z^i = c', \Theta^t)}$$

- The weighted MLE in the M-step is given by

$$\theta_c^{t+1} = \frac{1}{n}\sum_{i=1}^n r_c^i \quad \text{(proportion of examples soft-assigned to cluster } c\text{)}$$

$$\mu_c^{t+1} = \frac{\sum_{i=1}^n r_c^i x^i}{n\theta_c^{t+1}} \quad \text{(mean of examples soft-assigned to } c\text{)}$$

$$\Sigma_c^{t+1} = \frac{\sum_{i=1}^n r_c^i (x^i - \mu_c^{t+1})(x^i - \mu_c^{t+1})^T}{n\theta_c^{t+1}} \quad \text{(covariance of examples soft-assigned to } c\text{).}$$

Expectation Maximization for Mixture of Gaussians

- The weights from the E-step are the responsibilitites,

$$r_c^i \triangleq p(z^i = c | x^i, \Theta^t) = \frac{p(x^i | z^i = c, \Theta^t) p(z^i = c, \Theta^t)}{\sum_{c'=1}^k p(x^i | z^i = c', \Theta^t) p(z^i = c', \Theta^t)}$$

- The weighted MLE in the M-step is given by

$$\theta_c^{t+1} = \frac{1}{n} \sum_{i=1}^n r_c^i \quad \text{(proportion of examples soft-assigned to cluster } c\text{)}$$

$$\mu_c^{t+1} = \frac{\sum_{i=1}^n r_c^i x^i}{n \theta_c^{t+1}} \quad \text{(mean of examples soft-assigned to } c\text{)}$$

$$\Sigma_c^{t+1} = \frac{\sum_{i=1}^n r_c^i (x^i - \mu_c^{t+1})(x^i - \mu_c^{t+1})^T}{n \theta_c^{t+1}} \quad \text{(covariance of examples soft-assigned to } c\text{).}$$

- Derivation is tedious, I'll put a note on the webpage.
  - Uses distributive law, probabilities sum to one, Lagrangian, weighted Gaussian MLE.
- This is k-means if covariances are constant, and $r_c^i = 1$ for most likely cluster.

Expectation Maximization for Mixture of Gaussians

EM for fitting mixture of Gaussians in action:
`https://www.youtube.com/watch?v=B36fzChfyGU`

Discussing of EM for Mixtures of Gaussians

- EM and Gaussian mixtures are used in a ton of applications.
  - One of the default unsupervised learning methods.

Discussing of EM for Mixtures of Gaussians

- EM and Gaussian mixtures are used in a ton of applications.
    - One of the default unsupervised learning methods.
- EM usually doesn't reach global optimum.
    - Restart the algorithm from different initializations.

Discussing of EM for Mixtures of Gaussians

- EM and Gaussian mixtures are used in a ton of applications.
    - One of the default unsupervised learning methods.
- EM usually doesn't reach global optimum.
    - Restart the algorithm from different initializations.
- MLE for some clusters may not exist (e.g., only responsible for one point).
    - Use MAP estimates or remove these clusters.

Discussing of EM for Mixtures of Gaussians

- EM and Gaussian mixtures are used in a ton of applications.
  - One of the default unsupervised learning methods.
- EM usually doesn't reach global optimum.
  - Restart the algorithm from different initializations.
- MLE for some clusters may not exist (e.g., only responsible for one point).
  - Use MAP estimates or remove these clusters.
- How do you choose number of mixtures k?
  - Use cross-validation or other model selection criteria.

Discussing of EM for Mixtures of Gaussians

- EM and Gaussian mixtures are used in a ton of applications.
  - One of the default unsupervised learning methods.
- EM usually doesn't reach global optimum.
  - Restart the algorithm from different initializations.
- MLE for some clusters may not exist (e.g., only responsible for one point).
  - Use MAP estimates or remove these clusters.
- How do you choose number of mixtures k?
  - Use cross-validation or other model selection criteria.
- Can you make it robust?
  - Use mixture of Laplace of student t distributions.

Discussing of EM for Mixtures of Gaussians

- EM and Gaussian mixtures are used in a ton of applications.
  - One of the default unsupervised learning methods.
- EM usually doesn't reach global optimum.
  - Restart the algorithm from different initializations.
- MLE for some clusters may not exist (e.g., only responsible for one point).
  - Use MAP estimates or remove these clusters.
- How do you choose number of mixtures k?
  - Use cross-validation or other model selection criteria.
- Can you make it robust?
  - Use mixture of Laplace of student t distributions.
- Are there alternatives to EM?
  - Could use gradient descent on NLL.
  - Spectral and other recent methods have some global guarantees.

(pause)

A Non-Parametric Mixture Model

- The classic parametric mixture model has the form

$$p(x) = \sum_{c=1}^{k} p(z = c)p(x|z = c).$$

A Non-Parametric Mixture Model

- The classic parametric mixture model has the form

$$p(x) = \sum_{c=1}^{k} p(z = c)p(x|z = c).$$

- A natural way to define a non-parametric mixture model is

$$p(x) = \sum_{i=1}^{n} p(z = i)p(x|z = i),$$

  where we have one mixture for every training example i.

A Non-Parametric Mixture Model

- The classic parametric mixture model has the form

$$p(x) = \sum_{c=1}^{k} p(z=c)p(x|z=c).$$

- A natural way to define a non-parametric mixture model is

$$p(x) = \sum_{i=1}^{n} p(z=i)p(x|z=i),$$

  where we have one mixture for every training example i.

- Common example: z is uniform and $x|z$ is Gaussian with mean x^i,

$$p(x) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{N}(x|x^i, \sigma^2 I),$$

  and we use a shared covariance $\sigma^2 I$ (and σ estimated by cross-validation).

A Non-Parametric Mixture Model

- The classic parametric mixture model has the form

$$p(x) = \sum_{c=1}^{k} p(z = c)p(x|z = c).$$

- A natural way to define a non-parametric mixture model is

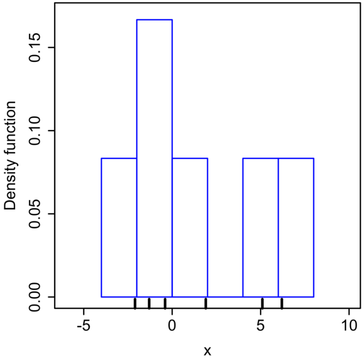
$$p(x) = \sum_{i=1}^{n} p(z = i)p(x|z = i),$$

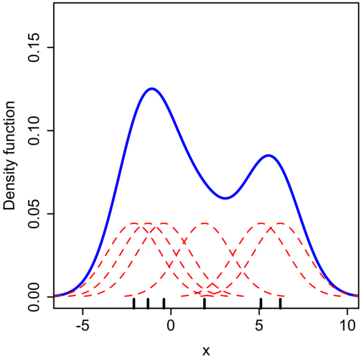
  where we have one mixture for every training example i.
- Common example: z is uniform and $x|z$ is Gaussian with mean x^i,

$$p(x) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{N}(x|x^i, \sigma^2 I),$$

  and we use a shared covariance $\sigma^2 I$ (and σ estimated by cross-validation).
- A special case of kernel density estimation or Parzen window.

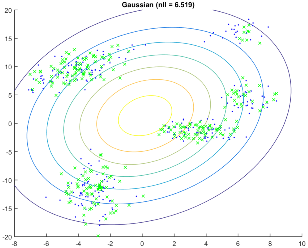
Parzen Window vs. Gaussian and Mixture of Gaussian

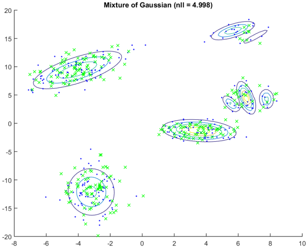




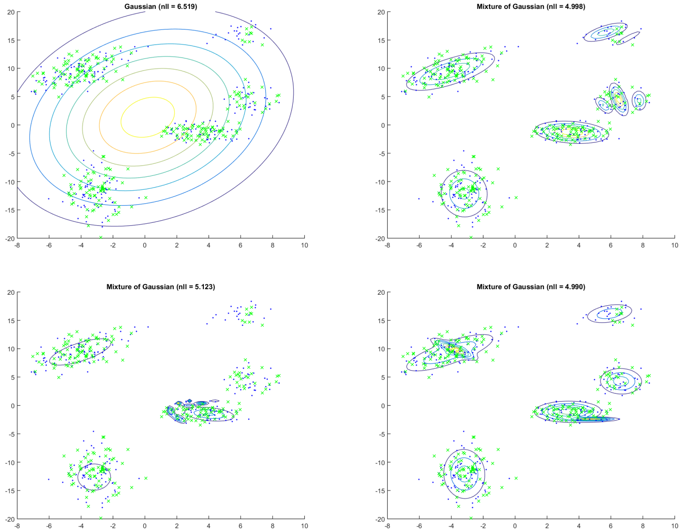
https://en.wikipedia.org/wiki/Kernel_density_estimation

Parzen Window vs. Gaussian and Mixture of Gaussian

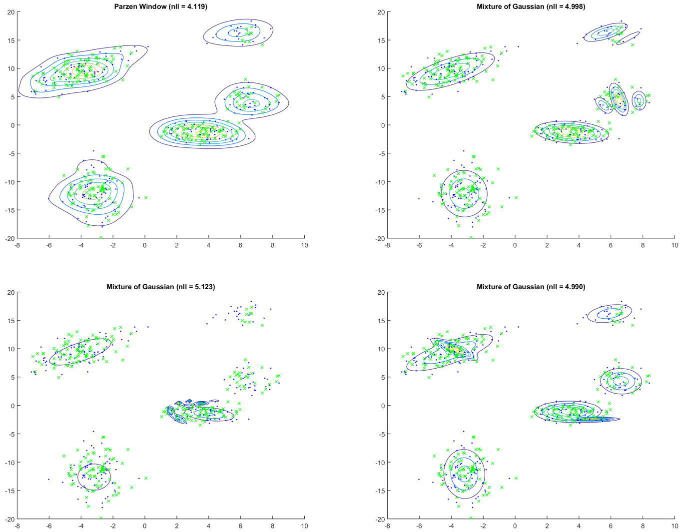




Parzen Window vs. Gaussian and Mixture of Gaussian



Parzen Window vs. Gaussian and Mixture of Gaussian



Kernel Density Estimation

- The 1D kernel density estimation (KDE) model uses

$$p(x) = \frac{1}{n}\sum_{i=1}^{n} k_h(x - x^i),$$

where the PDF k is "kernel" and the parameter h is the "bandwidth".

Kernel Density Estimation

- The 1D kernel density estimation (KDE) model uses
$$p(x) = \frac{1}{n}\sum_{i=1}^{n} k_h(x - x^i),$$
where the PDF k is "kernel" and the parameter h is the "bandwidth".
- In the previous slide we used the (normalized) Gaussian kernel,
$$k_1(x) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{x^2}{2}\right), \quad k_h(x) = \frac{1}{h\sqrt{2\pi}}\exp\left(-\frac{x^2}{2h^2}\right).$$

Kernel Density Estimation

- The 1D kernel density estimation (KDE) model uses

$$p(x) = \frac{1}{n}\sum_{i=1}^{n} k_h(x - x^i),$$

  where the PDF k is "kernel" and the parameter h is the "bandwidth".

- In the previous slide we used the (normalized) Gaussian kernel,

$$k_1(x) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{x^2}{2}\right), \quad k_h(x) = \frac{1}{h\sqrt{2\pi}}\exp\left(-\frac{x^2}{2h^2}\right).$$

- Note that we can add a bandwith h to any PDF k_1, using

$$k_h(x) = \frac{1}{h}k_1\left(\frac{x}{h}\right),$$

  which follows from the change of variables formula for probabilities.

- Under common choices of kernels, KDEs can model any density.

Efficient Kernel Density Estimation

- KDE with the Gaussian kernel is slow at test time:
  - We need to compute distance of test point to every training point.

Efficient Kernel Density Estimation

- KDE with the Gaussian kernel is slow at test time:
  - We need to compute distance of test point to every training point.
- A common alternative is the Epanechnikov kernel,

$$k_1(x) = \frac{3}{4}\left(1 - x^2\right)\mathcal{I}\left[|x| \leq 1\right].$$

- This kernel has two nice properties:
  - Epanechnikov showed that it is asymptotically optimal in terms of squared error.
  - It can be much faster to use since it only depends on nearby points.
    - You can use fast methods for computing nearest neighbours.

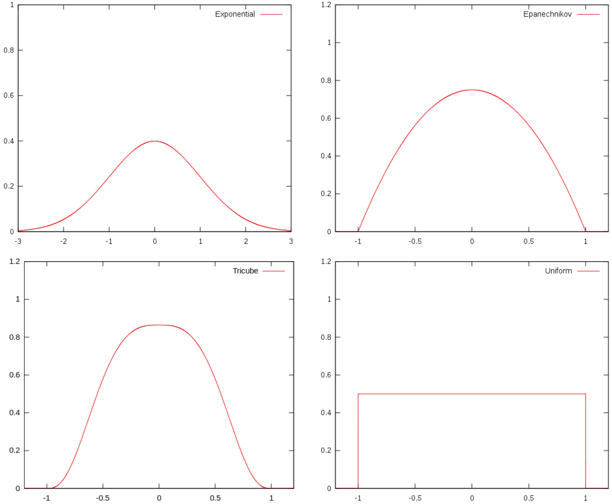
Efficient Kernel Density Estimation

- KDE with the Gaussian kernel is slow at test time:
  - We need to compute distance of test point to every training point.
- A common alternative is the Epanechnikov kernel,

$$k_1(x) = \frac{3}{4}\left(1 - x^2\right)\mathcal{I}\left[|x| \leq 1\right].$$

- This kernel has two nice properties:
  - Epanechnikov showed that it is asymptotically optimal in terms of squared error.
  - It can be much faster to use since it only depends on nearby points.
    - You can use fast methods for computing nearest neighbours.
- The kernel is non-smooth, at the boundaries, but many smooth approximations exist.
  - Quartic, triweight, tricube, cosine, etc.

Efficient Kernel Density Estimation



https://en.wikipedia.org/wiki/Kernel_%28statistics%29

Multivariate Kernel Density Estimation

- The general kernel density estimation (KDE) model uses

$$p(x) = \frac{1}{n}\sum_{i=1}^{n} k_{\Sigma}(x - x^i),$$

- The most common kernel is again the Gaussian,

$$k_I(x) = \frac{1}{\sqrt{2\pi}^{\frac{d}{2}}}\exp\left(-\frac{\|x\|^2}{2}\right).$$

Multivariate Kernel Density Estimation

- The general kernel density estimation (KDE) model uses

$$p(x) = \frac{1}{n}\sum_{i=1}^{n} k_\Sigma(x - x^i),$$

- The most common kernel is again the Gaussian,

$$k_I(x) = \frac{1}{\sqrt{2\pi}^{\frac{d}{2}}}\exp\left(-\frac{\|x\|^2}{2}\right).$$

- By the multivariate change of variables formula we can add bandwith H using

$$k_H(x) = \frac{1}{|H|}k_1(H^{-1}x) \qquad \text{(generalizes } k_h(x) = \frac{1}{h}k_1\left(\frac{x}{h}\right)\text{)}.$$

- We get a multivariate Gaussian corresponds to using $H = \Sigma^{\frac{1}{2}}$.
- To reduce number of paramters, we typically:
  - Use a product of independent distributions and use $H = hI$ for some h.

Summary

- Semi-supervised learning: Learning with labeled and unlabeled data.

Summary

- Semi-supervised learning: Learning with labeled and unlabeled data.
- Expectation maximization: optimization with hidden variables, when knowing hidden variables make problem easy.

Summary

- Semi-supervised learning: Learning with labeled and unlabeled data.
- Expectation maximization: optimization with hidden variables, when knowing hidden variables make problem easy.
- Monotonicity of EM: EM is guaranteed not to decrease likelihood.

Summary

- Semi-supervised learning: Learning with labeled and unlabeled data.
- Expectation maximization: optimization with hidden variables, when knowing hidden variables make problem easy.
- Monotonicity of EM: EM is guaranteed not to decrease likelihood.
- Kernel density estimation: Non-parametric continuous density estimation method.

- Next time: Probabilistic PCA and factor analysis.