Introduction to Machine Learning

Lecture 10　Representation and Clustering - Bayesian Gaussian Mixture Models and Mean Shift

Hongteng Xu

中国人民大學
RENMIN UNIVERSITY OF CHINA

高瓴人工智能学院
Gaoling School of Artificial Intelligence

Outline

Review

- Generative modeling and Gaussian mixture model
- EM algorithm
- Revisit K-means from an EM viewpoint

Outline

Review

- Generative modeling and Gaussian mixture model
- EM algorithm
- Revisit K-means from an EM viewpoint

Today

- A Bayesian viewpoint of Gaussian mixture model and MCMC
- Nonparametric clustering and kernel density estimation
- Mean shift algorithm.

Revisit The Generative Mechanism of GMM

Suppose that there are K Gaussian distributions defined on the sample space $\mathcal{X} \subset \mathbb{R}^D$.

- $\boldsymbol{w} = [w_k] \in \Delta^{K-1}$
- $\{\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\}_{k=1}^K$

Generative process:

1. Determine the cluster: $k \sim \text{Categorical}(\boldsymbol{w})$
2. Determine the sample based on the cluster: $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$.

Revisit The Generative Mechanism of GMM

Suppose that there are K Gaussian distributions defined on the sample space $\mathcal{X} \subset \mathbb{R}^D$.

- $\boldsymbol{w} = [w_k] \in \Delta^{K-1}$
- $\{\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\}_{k=1}^K$

Generative process:

1. Determine the cluster: $k \sim \text{Categorical}(\boldsymbol{w})$
2. Determine the sample based on the cluster: $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$.

How to determine the number of clusters and the corresponding distributions?

Bayesian Inference of GMM

- Recall that a GMM is

$$p(x) = \sum_{k=1}^{K} w_k \underbrace{p_k(x; \boldsymbol{\mu}_k, \Sigma_k)}_{\text{Gaussian}}$$

Bayesian Inference of GMM

- Recall that a GMM is

$$p(x) = \sum_{k=1}^{K} w_k \underbrace{p_k(x; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}_{\text{Gaussian}} = \sum_{k=1}^{K} w_k p_k(x; \boldsymbol{\mu}_k, \underbrace{\boldsymbol{\Phi}_k}_{\boldsymbol{\Sigma}_k^{-1}}).$$

Bayesian Inference of GMM

- Recall that a GMM is

$$p(x) = \sum_{k=1}^{K} w_k \underbrace{p_k(x; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}_{\text{Gaussian}} = \sum_{k=1}^{K} w_k p_k(x; \boldsymbol{\mu}_k, \underbrace{\boldsymbol{\Phi}_k}_{\boldsymbol{\Sigma}_k^{-1}}). \tag{1}$$

- **Bayesian GMM** sets the prior distributions for its parameters $\{w_k, \boldsymbol{\mu}_k, \boldsymbol{\Phi}_k\}_{k=1}^{K}$.

Bayesian Inference of GMM (1D)

Conjugate Priors

- $\boldsymbol{w} \sim \text{Dirichlet}(\delta_1, ..., \delta_K)$
- $\phi_k \sim \text{Gamma}(\frac{a}{2}, \frac{b}{2}), \forall k$
- $\mu_k|\phi \sim \mathcal{N}(m_k, \frac{1}{\alpha_k \phi_k}), \forall k$

Bayesian Inference of GMM (1D)

Conjugate Priors

- $\boldsymbol{w} \sim \text{Dirichlet}(\delta_1, \dots, \delta_K)$
- $\phi_k \sim \text{Gamma}(\frac{a}{2}, \frac{b}{2}), \forall k$
- $\mu_k | \phi \sim \mathcal{N}(m_k, \frac{1}{\alpha_k \phi_k}), \forall k$

Posteriors given data x's and the latent code z's

- $\boldsymbol{w} | x, z \sim \text{Dirichlet}(\delta_1^*, \dots, \delta_K^*)$
- $\phi_k | x, z \sim \text{Gamma}(\frac{a_k^*}{2}, \frac{b_k^*}{2}), \forall k$
- $\mu_k | x, z, \phi \sim \mathcal{N}(m_k^*, \frac{1}{\alpha_k^* \phi_k}), \forall k$

Bayesian Inference of GMM (1D)

Conjugate Priors

- $\boldsymbol{w} \sim \text{Dirichlet}(\delta_1, ..., \delta_K)$
- $\phi_k \sim \text{Gamma}(\frac{a}{2}, \frac{b}{2}), \forall k$
- $\mu_k|\phi \sim \mathcal{N}(m_k, \frac{1}{\alpha_k \phi_k}), \forall k$

Posteriors given data x's and the latent code z's

- $\boldsymbol{w}|x, z \sim \text{Dirichlet}(\delta_1^*, ..., \delta_K^*)$
- $\phi_k|x, z \sim \text{Gamma}(\frac{a_k^*}{2}, \frac{b_k^*}{2}), \forall k$
- $\mu_k|x, z, \phi \sim \mathcal{N}(m_k^*, \frac{1}{\alpha_k^* \phi_k}), \forall k$

$$
\begin{aligned}
&\delta_k^* = \delta_k + N_k, \quad a_k^* = a + N_k, \quad b_k^* = b + \sum\nolimits_{z_j = k} (x_j - \mu_k)^2 \\
&\alpha_k^* = \alpha_k + N_k, \quad m_k^* = \frac{1}{\alpha_k^*}(\alpha_k m_k + \sum\nolimits_{z_j = k} x_j)
\end{aligned} \tag{2}
$$

MCMC Algorithm for The Bayesian Inference of GMM (1D)

MCMC Algorithm:

- Initialize $\{w_k, \mu_k, \phi_k\}_{k=1}^{K}$ via sampling from priors

MCMC Algorithm for The Bayesian Inference of GMM (1D)

MCMC Algorithm:

- Initialize $\{w_k, \mu_k, \phi_k\}_{k=1}^{K}$ via sampling from priors
- Repeat till converge
  1. Sampling latent codes $z_{nk} \sim z | x_n, w_k, \mu_k, \phi_k$
  2. Sampling $\boldsymbol{w} \sim \boldsymbol{w} | \{x_n\}_n, \{z_{nk}\}_{n,k}$
  3. Sampling $\phi_k \sim \phi_k | \{x_n\}_n, \{z_{nk}\}_{n,k}$
  4. Sampling $\mu_k \sim \mu_k | \{x_n\}_n, \{z_{nk}\}_{n,k}, \phi_k$

MCMC Algorithm for The Bayesian Inference of GMM (1D)

MCMC Algorithm:

- Initialize $\{w_k, \mu_k, \phi_k\}_{k=1}^{K}$ via sampling from priors
- Repeat till converge
  1. Sampling latent codes $z_{nk} \sim z|x_n, w_k, \mu_k, \phi_k$
  2. Sampling $\boldsymbol{w} \sim \boldsymbol{w}|\{x_n\}_n, \{z_{nk}\}_{n,k}$
  3. Sampling $\phi_k \sim \phi_k|\{x_n\}_n, \{z_{nk}\}_{n,k}$
  4. Sampling $\mu_k \sim \mu_k|\{x_n\}_n, \{z_{nk}\}_{n,k}, \phi_k$

Compare to EM:

- EM estimations the responsibility (the probability $p(z|x)$ of latent codes) and **optimizes** parameters in a deterministic way.

MCMC Algorithm for The Bayesian Inference of GMM (1D)

MCMC Algorithm:

- Initialize $\{w_k, \mu_k, \phi_k\}_{k=1}^{K}$ via sampling from priors
- Repeat till converge
  1. Sampling latent codes $z_{nk} \sim z|x_n, w_k, \mu_k, \phi_k$
  2. Sampling $\boldsymbol{w} \sim \boldsymbol{w}|\{x_n\}_n, \{z_{nk}\}_{n,k}$
  3. Sampling $\phi_k \sim \phi_k|\{x_n\}_n, \{z_{nk}\}_{n,k}$
  4. Sampling $\mu_k \sim \mu_k|\{x_n\}_n, \{z_{nk}\}_{n,k}, \phi_k$

Compare to EM:

- EM estimations the responsibility (the probability $p(z|x)$ of latent codes) and **optimizes** parameters in a deterministic way.
- MCMC **samples** latent codes and parameters in a probabilistic way.

Bayesian GMM

- Modeling the uncertainty of model

Bayesian GMM

- Modeling the uncertainty of model
- Extend to infinite mixture model (learn the number of clusters)

Bayesian GMM

- Modeling the uncertainty of model
- Extend to infinite mixture model (learn the number of clusters)
- Generally, the complexity is high (due to the efficiency of sampling)

Nonparametric clustering

Parametric clustering models

- Kmeans, GMM, ... they are parametric clustering models.

Nonparametric clustering

Parametric clustering models

- Kmeans, GMM, ... they are parametric clustering models.
- The distribution of data is parametric and its inference is inductive.

Nonparametric clustering

Parametric clustering models

- Kmeans, GMM, ... they are parametric clustering models.
- The distribution of data is parametric and its inference is inductive.

Nonparametric clustering models

- The distribution of data is constructed by the data itself.

Nonparametric clustering

Parametric clustering models

- Kmeans, GMM, ... they are parametric clustering models.
- The distribution of data is parametric and its inference is inductive.

Nonparametric clustering models

- The distribution of data is constructed by the data itself.
- The inference is transductive.

Kernel Density Estimation (KDE)

Motivation:

- Given i.i.d. samples, how to estimate their probability density function (PDF) in a nonparametric way.

Kernel Density Estimation (KDE)

Motivation:

- Given i.i.d. samples, how to estimate their probability density function (PDF) in a nonparametric way.

Principle:

- Given i.i.d. samples $\{x_n\}_{n=1}^N$, estimate the unknown PDF $p(x)$ by the data themselves as

$$\hat{p}_h(x) = \frac{1}{n} \sum_{n=1}^{N} K_h(x, x_n)$$

Kernel Density Estimation (KDE)

Motivation:

- Given i.i.d. samples, how to estimate their probability density function (PDF) in a nonparametric way.

Principle:

- Given i.i.d. samples $\{x_n\}_{n=1}^{N}$, estimate the unknown PDF $p(x)$ by the data themselves as

$$\hat{p}_h(x) = \frac{1}{n}\sum_{n=1}^{N} K_h(x, x_n) := \frac{1}{n}\sum_{n=1}^{N} K_h(x - x_n) \quad \text{(Recall nonparametric kernel)} \tag{3}$$

h is the bandwidth.

Kernel Density Estimation (KDE)

Motivation:

- Given i.i.d. samples, how to estimate their probability density function (PDF) in a nonparametric way.

Principle:

- Given i.i.d. samples $\{x_n\}_{n=1}^{N}$, estimate the unknown PDF $p(x)$ by the data themselves as

$$\hat{p}_h(x) = \frac{1}{n}\sum_{n=1}^{N} K_h(x, x_n) := \frac{1}{n}\sum_{n=1}^{N} K_h(x - x_n) \quad \text{(Recall nonparametric kernel)} \quad (3)$$

  h is the bandwidth.

- Recall the constraints for a valid nonparametric kernel.

Kernel Density Estimation (KDE)

Motivation:

- Given i.i.d. samples, how to estimate their probability density function (PDF) in a nonparametric way.

Principle:

- Given i.i.d. samples $\{x_n\}_{n=1}^{N}$, estimate the unknown PDF $p(x)$ by the data themselves as

$$\hat{p}_h(x) = \frac{1}{n}\sum_{n=1}^{N} K_h(x, x_n) := \frac{1}{n}\sum_{n=1}^{N} K_h(x - x_n) \quad \text{(Recall nonparametric kernel)} \tag{3}$$

  h is the bandwidth.

- Recall the constraints for a valid nonparametric kernel.
- For 1D data, Gaussian (RBF) kernel, Gaussian-like density, $h = (\frac{4\hat{\sigma}^5}{3N})^{0.2} \approx 1.06\hat{\sigma}N^{-0.2}$.

Kernel Density Estimation (KDE)

Motivation:

- Given i.i.d. samples, how to estimate their probability density function (PDF) in a nonparametric way.

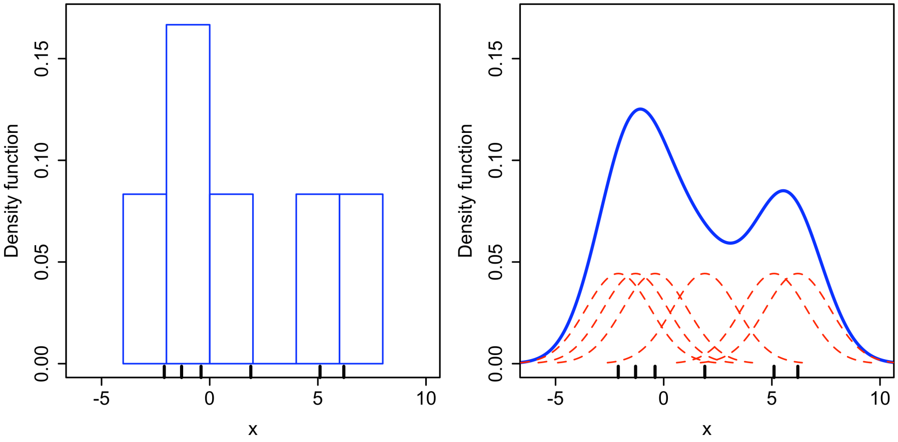
Principle:

- Given i.i.d. samples $\{x_n\}_{n=1}^N$, estimate the unknown PDF $p(x)$ by the data themselves as

$$\hat{p}_h(x) = \frac{1}{n}\sum_{n=1}^{N} K_h(x, x_n) := \frac{1}{n}\sum_{n=1}^{N} K_h(x - x_n) \quad \text{(Recall nonparametric kernel)} \tag{3}$$

  h is the bandwidth.
- Recall the constraints for a valid nonparametric kernel.
- For 1D data, Gaussian (RBF) kernel, Gaussian-like density, $h = (\frac{4\hat{\sigma}^5}{3N})^{0.2} \approx 1.06\hat{\sigma}N^{-0.2}$.
- For 1D data in general, $h = \mathcal{O}(N^{-0.2})$.

Histogram: The Simplest Kernel Density Estimation



Connections to What We Learned/Will Learn

Mixture Model:

- When K_h is a Gaussian kernel: $K_h(x, x') = \frac{1}{\sqrt{2\pi}h} \exp(-\frac{|x-x'|^2}{2h^2})$, KDE actually can be interpreted a GMM model with known parameters (See, the boundary of parametric and nonparametric modeling is not so strict:))

Connections to What We Learned/Will Learn

Mixture Model:

- When K_h is a Gaussian kernel: $K_h(x, x') = \frac{1}{\sqrt{2\pi}h} \exp(-\frac{|x-x'|^2}{2h^2})$, KDE actually can be interpreted a GMM model with known parameters (See, the boundary of parametric and nonparametric modeling is not so strict:))

Naïve Bayes Classifier:

- KDE is often used to estimate the class-conditional marginal densities of data, and thus, improve classification accuracy. (Next lecture)

Mean-shift Algorithm

Motivation:

- As an extension of the kernel density estimation.
- Locate the maxima of the density function (or called **mode**-seeking algorithm).

Mean-shift Algorithm

Motivation:

- As an extension of the kernel density estimation.
- Locate the maxima of the density function (or called **mode**-seeking algorithm).

Find the maxima based on KDE:

- Given $\{x_n\}_{n=1}^N$ and a nonparametric kernel K_h,

$$\hat{p}(x) = \frac{1}{N} \sum_{n=1}^{N} K_h(x, x_n)$$

Mean-shift Algorithm

Motivation:

- As an extension of the kernel density estimation.
- Locate the maxima of the density function (or called **mode**-seeking algorithm).

Find the maxima based on KDE:

- Given $\{x_n\}_{n=1}^N$ and a nonparametric kernel K_h,

$$\hat{p}(x) = \frac{1}{N}\sum_{n=1}^{N} K_h(x, x_n) \tag{4}$$

- "Brute force" approach:

$$\max_{x \in \mathcal{X}} \hat{p}(x). \tag{5}$$

  Gradient ascent, ...

Mean-shift Algorithm

Motivation:

- As an extension of the kernel density estimation.
- Locate the maxima of the density function (or called **mode**-seeking algorithm).

Find the maxima based on KDE:

- Given $\{x_n\}_{n=1}^{N}$ and a nonparametric kernel K_h,

$$\hat{p}(x) = \frac{1}{N} \sum_{n=1}^{N} K_h(x, x_n) \tag{4}$$

- "Brute force" approach:

$$\max_{x \in \mathcal{X}} \hat{p}(x). \tag{5}$$

  Gradient ascent, ...

- Curse of dimensionality

Mean-shift Algorithm

Principle:

- Given a set of samples $\{x_n \in \mathcal{X}\}_{n=1}^{N}$ and a kernel function $K : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}_+$, the mean of the samples in the neighborhood of x is

$$m(x) = \frac{\sum_{x_i \in \mathcal{N}(x)} K_h(x, x_i) x_i}{\sum_{x_i \in \mathcal{N}(x)} K_h(x, x_i)}.$$

Mean-shift Algorithm

Principle:

- Given a set of samples $\{x_n \in \mathcal{X}\}_{n=1}^{N}$ and a kernel function $K : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}_+$, the mean of the samples in the neighborhood of x is

$$m(x) = \frac{\sum_{x_i \in \mathcal{N}(x)} K_h(x, x_i) x_i}{\sum_{x_i \in \mathcal{N}(x)} K_h(x, x_i)}. \tag{6}$$

(A Nadaraya-Watson Estimator imposed on the data itself.)

Mean-shift Algorithm

Principle:

- Given a set of samples $\{x_n \in \mathcal{X}\}_{n=1}^{N}$ and a kernel function $K : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}_+$, the mean of the samples in the neighborhood of x is

$$m(x) = \frac{\sum_{x_i \in \mathcal{N}(x)} K_h(x, x_i) x_i}{\sum_{x_i \in \mathcal{N}(x)} K_h(x, x_i)}. \tag{6}$$

  (A Nadaraya-Watson Estimator imposed on the data itself.)

- $m(x)$ is the **mean** of $\mathcal{N}(x)$ and $m(x) - x$ is the **mean shift vector**.

Mean-shift Algorithm

Principle:

- Given a set of samples $\{x_n \in \mathcal{X}\}_{n=1}^N$ and a kernel function $K : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}_+$, the mean of the samples in the neighborhood of x is

$$m(x) = \frac{\sum_{x_i \in \mathcal{N}(x)} K_h(x, x_i) x_i}{\sum_{x_i \in \mathcal{N}(x)} K_h(x, x_i)}. \tag{6}$$

  (A Nadaraya-Watson Estimator imposed on the data itself.)
- $m(x)$ is the **mean** of $\mathcal{N}(x)$ and $m(x) - x$ is the **mean shift vector**.
  1. For each x_n, compute $m(x_n)$ by (6).
  2. Each $x_n \leftarrow m(x_n)$, and repeat step 1 till $m(x_n)$ converge.

Mean-shift Algorithm

Principle:

- Given a set of samples $\{x_n \in \mathcal{X}\}_{n=1}^{N}$ and a kernel function $K : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}_+$, the mean of the samples in the neighborhood of x is

$$m(x) = \frac{\sum_{x_i \in \mathcal{N}(x)} K_h(x, x_i) x_i}{\sum_{x_i \in \mathcal{N}(x)} K_h(x, x_i)}. \tag{6}$$

  (A Nadaraya-Watson Estimator imposed on the data itself.)
- $m(x)$ is the **mean** of $\mathcal{N}(x)$ and $m(x) - x$ is the **mean shift vector**.
  1. For each x_n, compute $m(x_n)$ by (6).
  2. Each $x_n \leftarrow m(x_n)$, and repeat step 1 till $m(x_n)$ converge.
- The mean shift vector always points toward the direction of the maximum increase in the density.

Mean-shift Algorithm

Principle:

- Given a set of samples $\{x_n \in \mathcal{X}\}_{n=1}^{N}$ and a kernel function $K : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}_+$, the mean of the samples in the neighborhood of x is

$$m(x) = \frac{\sum_{x_i \in \mathcal{N}(x)} K_h(x, x_i) x_i}{\sum_{x_i \in \mathcal{N}(x)} K_h(x, x_i)}. \tag{6}$$

  (A Nadaraya-Watson Estimator imposed on the data itself.)
- $m(x)$ is the **mean** of $\mathcal{N}(x)$ and $m(x) - x$ is the **mean shift vector**.
  1. For each x_n, compute $m(x_n)$ by (6).
  2. Each $x_n \leftarrow m(x_n)$, and repeat step 1 till $m(x_n)$ converge.
- The mean shift vector always points toward the direction of the maximum increase in the density.
- At every iteration the kernel is shifted to the mean of the points within it.

Mean-shift Algorithm

Principle:

- Given a set of samples $\{x_n \in \mathcal{X}\}_{n=1}^{N}$ and a kernel function $K : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}_+$, the mean of the samples in the neighborhood of x is

$$m(x) = \frac{\sum_{x_i \in \mathcal{N}(x)} K_h(x, x_i) x_i}{\sum_{x_i \in \mathcal{N}(x)} K_h(x, x_i)}. \tag{6}$$

  (A Nadaraya-Watson Estimator imposed on the data itself.)
- $m(x)$ is the **mean** of $\mathcal{N}(x)$ and $m(x) - x$ is the **mean shift vector**.
  1. For each x_n, compute $m(x_n)$ by (6).
  2. Each $x_n \leftarrow m(x_n)$, and repeat step 1 till $m(x_n)$ converge.
- The mean shift vector always points toward the direction of the maximum increase in the density.
- At every iteration the kernel is shifted to the mean of the points within it.
- **Why?**

The Rationality of (Gaussian) Mean-shift

- Suppose that $K_h(x, x') = \frac{1}{\sqrt{2\pi}h} \exp(-\frac{\|x-x'\|_2^2}{2h^2})$, derive $\frac{\partial \hat{p}(x)}{\partial x}$.

The Rationality of (Gaussian) Mean-shift

- Suppose that $K_h(x, x') = \frac{1}{\sqrt{2\pi}h} \exp(-\frac{\|x-x'\|_2^2}{2h^2})$, derive $\frac{\partial \hat{p}(x)}{\partial x}$.

$$\frac{\partial \hat{p}(x)}{\partial x} = \frac{1}{N} \sum_{n=1}^{N} \frac{\partial K_h(x, x_n)}{\partial x}$$

The Rationality of (Gaussian) Mean-shift

- Suppose that $K_h(x, x') = \frac{1}{\sqrt{2\pi}h} \exp(-\frac{\|x-x'\|_2^2}{2h^2})$, derive $\frac{\partial \hat{p}(x)}{\partial x}$.

$$
\begin{aligned}
\frac{\partial \hat{p}(x)}{\partial x} &= \frac{1}{N} \sum_{n=1}^{N} \frac{\partial K_h(x, x_n)}{\partial x} \\
&= -\frac{1}{Nh^2} \sum_{n=1}^{N} K_h(x, x_n)(x - x_n)
\end{aligned}
$$

The Rationality of (Gaussian) Mean-shift

- Suppose that $K_h(x, x') = \frac{1}{\sqrt{2\pi}h} \exp(-\frac{\|x-x'\|_2^2}{2h^2})$, derive $\frac{\partial \hat{p}(x)}{\partial x}$.

$$
\begin{aligned}
\frac{\partial \hat{p}(x)}{\partial x} &= \frac{1}{N} \sum_{n=1}^{N} \frac{\partial K_h(x, x_n)}{\partial x} \\
&= -\frac{1}{Nh^2} \sum_{n=1}^{N} K_h(x, x_n)(x - x_n) \\
&\propto \sum_{n=1}^{N} K_h(x, x_n)(x_n - x)
\end{aligned}
$$

The Rationality of (Gaussian) Mean-shift

- Suppose that $K_h(x,x') = \frac{1}{\sqrt{2\pi}h}\exp(-\frac{\|x-x'\|_2^2}{2h^2})$, derive $\frac{\partial \hat{p}(x)}{\partial x}$.

$$
\begin{aligned}
\frac{\partial \hat{p}(x)}{\partial x} &= \frac{1}{N}\sum_{n=1}^{N}\frac{\partial K_h(x,x_n)}{\partial x} \\
&= -\frac{1}{Nh^2}\sum_{n=1}^{N}K_h(x,x_n)(x-x_n) \\
&\propto \sum_{n=1}^{N}K_h(x,x_n)(x_n-x) \propto \underbrace{\frac{\sum_{n=1}^{N}K_h(x,x_n)x_n}{\sum_{n=1}^{N}K_h(x,x_n)}}_{m(x)} - x.
\end{aligned}
$$

The Rationality of (Gaussian) Mean-shift

- Suppose that $K_h(x, x') = \frac{1}{\sqrt{2\pi}h} \exp(-\frac{\|x-x'\|_2^2}{2h^2})$, derive $\frac{\partial \hat{p}(x)}{\partial x}$.

$$\begin{aligned}\frac{\partial \hat{p}(x)}{\partial x} &= \frac{1}{N} \sum\nolimits_{n=1}^{N} \frac{\partial K_h(x, x_n)}{\partial x} \\ &= -\frac{1}{Nh^2} \sum\nolimits_{n=1}^{N} K_h(x, x_n)(x - x_n) \\ &\propto \sum\nolimits_{n=1}^{N} K_h(x, x_n)(x_n - x) \propto \underbrace{\frac{\sum_{n=1}^{N} K_h(x, x_n) x_n}{\sum_{n=1}^{N} K_h(x, x_n)}}_{m(x)} - x.\end{aligned} \tag{7}$$

- **In summary,** $x_n^{(t+1)} = x_n^{(t)} + \tau_n \frac{\partial \hat{p}(x)}{\partial x}$ **is achieved by mean shift.**

The Rationality of (Gaussian) Mean-shift

- Suppose that $K_h(x, x') = \frac{1}{\sqrt{2\pi}h}\exp(-\frac{\|x-x'\|_2^2}{2h^2})$, derive $\frac{\partial \hat{p}(x)}{\partial x}$.

$$
\begin{aligned}
\frac{\partial \hat{p}(x)}{\partial x} &= \frac{1}{N}\sum\nolimits_{n=1}^{N}\frac{\partial K_h(x, x_n)}{\partial x} \\
&= -\frac{1}{Nh^2}\sum\nolimits_{n=1}^{N} K_h(x, x_n)(x - x_n) \\
&\propto \sum\nolimits_{n=1}^{N} K_h(x, x_n)(x_n - x) \propto \underbrace{\frac{\sum_{n=1}^{N} K_h(x, x_n)x_n}{\sum_{n=1}^{N} K_h(x, x_n)}}_{m(x)} - x.
\end{aligned}
\tag{7}
$$

- **In summary, $x_n^{(t+1)} = x_n^{(t)} + \tau_n \frac{\partial \hat{p}(x)}{\partial x}$ is achieved by mean shift.**
- When kernel is band-limited, or we set $n \in \mathcal{N}(x)$ rather than $\{1, ..., N\}$, the gradient ascent is stochastic/adaptive, and we obtain mean-shift.

The Rationality of (Gaussian) Mean-shift

- Suppose that $K_h(x,x') = \frac{1}{\sqrt{2\pi}h}\exp(-\frac{\|x-x'\|_2^2}{2h^2})$, derive $\frac{\partial \hat{p}(x)}{\partial x}$.

$$\begin{aligned}\frac{\partial \hat{p}(x)}{\partial x} &= \frac{1}{N}\sum\nolimits_{n=1}^{N}\frac{\partial K_h(x,x_n)}{\partial x}\\ &= -\frac{1}{Nh^2}\sum\nolimits_{n=1}^{N}K_h(x,x_n)(x-x_n)\\ &\propto \sum\nolimits_{n=1}^{N}K_h(x,x_n)(x_n-x) \propto \underbrace{\frac{\sum_{n=1}^{N}K_h(x,x_n)x_n}{\sum_{n=1}^{N}K_h(x,x_n)}}_{m(x)} - x.\end{aligned} \tag{7}$$

- **In summary,** $x_n^{(t+1)} = x_n^{(t)} + \tau_n\frac{\partial \hat{p}(x)}{\partial x}$ **is achieved by mean shift.**
- When kernel is band-limited, or we set $n \in \mathcal{N}(x)$ rather than $\{1, ..., N\}$, the gradient ascent is stochastic/adaptive, and we obtain mean-shift.
- The curse of dimensionality is still a problem.

(Gaussian) Mean-shift Is An EM Algorithm

(Gaussian) Mean-shift Is An EM Algorithm

E-step:

- Compute $m(x_n)\ \forall n$.

(Gaussian) Mean-shift Is An EM Algorithm

E-step:

- Compute $m(x_n)$ $\forall n$.

M-step:

- Update $x_n \leftarrow m(x_n)$.

(Gaussian) Mean-shift Is An EM Algorithm

E-step:

- Compute $m(x_n)\ \forall n$.

M-step:

- Update $x_n \leftarrow m(x_n)$.

Recall that EM learns the model with **observed data** and **latent variables**.

(Gaussian) Mean-shift Is An EM Algorithm

E-step:

- Compute $m(x_n)$ $\forall n$.

M-step:

- Update $x_n \leftarrow m(x_n)$.

Recall that EM learns the model with **observed data** and **latent variables**.

- Data: x_n's.

(Gaussian) Mean-shift Is An EM Algorithm

E-step:

- Compute $m(x_n)\ \forall n$.

M-step:

- Update $x_n \leftarrow m(x_n)$.

Recall that EM learns the model with **observed data** and **latent variables**.

- Data: x_n's.
- Latent variable: means/centroids m's.

(Gaussian) Mean-shift Is An EM Algorithm

E-step:

- Compute $m(x_n)\ \forall n$.

M-step:

- Update $x_n \leftarrow m(x_n)$.

Recall that EM learns the model with **observed data** and **latent variables**.

- Data: x_n's.
- Latent variable: means/centroids m's.
- E-step: estimate $m(x_n)$ (the mean conditioned on x_n)

(Gaussian) Mean-shift Is An EM Algorithm

E-step:

- Compute $m(x_n)$ $\forall n$.

M-step:

- Update $x_n \leftarrow m(x_n)$.

Recall that EM learns the model with **observed data** and **latent variables**.

- Data: x_n's.
- Latent variable: means/centroids m's.
- E-step: estimate $m(x_n)$ (the mean conditioned on x_n) (What is $p(m|x_n)$?)

(Gaussian) Mean-shift Is An EM Algorithm

E-step:

- Compute $m(x_n)$ $\forall n$.

M-step:

- Update $x_n \leftarrow m(x_n)$.

Recall that EM learns the model with **observed data** and **latent variables**.

- Data: x_n's.
- Latent variable: means/centroids m's.
- E-step: estimate $m(x_n)$ (the mean conditioned on x_n) (What is $p(m|x_n)$?)
- M-step: maximize $p(x) = \int p(x|m)p(m)\mathrm{d}m$.

(Gaussian) Mean-shift Is An EM Algorithm

E-step:

- Compute $m(x_n)$ $\forall n$.

M-step:

- Update $x_n \leftarrow m(x_n)$.

Recall that EM learns the model with **observed data** and **latent variables**.

- Data: x_n's.
- Latent variable: means/centroids m's.
- E-step: estimate $m(x_n)$ (the mean conditioned on x_n) (What is $p(m|x_n)$?)
- M-step: maximize $p(x) = \int p(x|m)p(m)\mathrm{d}m$. (What are $p(m)$ and $p(x|m)$?)

In Summary

- A Bayesian viewpoint of GMMs
- Kernel density estimation
- Mean-shift algorithm

Next...

- Classification problem and its challenges
- Linear classifiers (LDA and Logistic Regression)

HW 4: DDL May 12, 2022

Python Programming

1. Lab # 7 (4 Pts)
2. Lab # 8 (4 Pts)

Questions for Tech Report (6 Pts, ≤ 3 Pages)

1. **Gaussian mixture model with outliers.** Suppose that the observed data contains several outliers. The mixture model can be:

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} w_k p(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) + w_{K+1}, \quad \boldsymbol{w} = [w_1, ..., w_{K+1}] \in \Delta^K \tag{8}$$

and w_{K+1} is probability that the sample is an outlier. Modify the EM algorithm to learn this model. (2 Pts)

2. **Revisit mean-shift.** Derive the mean-shift as a maximum likelihood estimation method (refer to (7)). Derive a modified mean-shift as MAP if x owns a prior $\mathcal{N}(\mu, \sigma^2)$. What if the prior is a GMM $p_{\text{prior}}(x) = \frac{1}{K} \sum_k p(x; \mu_k, \sigma_k^2)$? (4 Pts)