CPSC 540 Tutorial

Reza Babanezhad
rezababa@cs.ubc.ca
Outline

• EM
• Robust PCA algorithm
• Fun with ML
• Question for you!
EM Algorithm

• $X$ is observed
• $Y$ is unobserved
• $\Theta$ is parameter whose estimation is easier with considering $Y$!

\[
p(x|\theta) = \int p(y, x|\theta) \, dy
\]

• E-Step:

\[
Q(\theta|\hat{\theta}^{(t)}) \equiv E[\log p(y, \theta|x)|x, \hat{\theta}^{(t)}]
\]

\[
\propto \log p(\theta) + E[\log p(y, x|\theta)|x, \hat{\theta}^{(t)}]
\]

\[
= \log p(\theta) + \int p(y|x, \hat{\theta}^{(t)}) \log p(y, x|\theta) \, dy
\]

• M-Step:

\[
\hat{\theta}^{(t+1)} = \arg \max_{\theta} Q(\theta|\hat{\theta}^{(t)})
\]
EM for GMM

• Multivariate Normal Distribution

\[ p(x_i | \mu, \Sigma_k) = (2\pi)^{-\frac{D}{2}} |\Sigma|^{-\frac{1}{2}} \exp \left( -\frac{1}{2} (x_i - \mu)^T \Sigma^{-1} (x_i - \mu) \right) \]

• Mixture distribution

\[ p(x_i) = \sum_{k=1}^{K} p(z_i = k | \pi) p(x_i | \mu_k, \Sigma_k), \]

• Probability of each cluster:

\[ p(z_i = k) = \pi_k \]
GMM

• E-Step:
  • Log likelihood:

\[
L(X, Z) = \log \prod_{k=1}^{K} \prod_{i=1}^{N} \left( \frac{P(z_i = k \mid \pi) P(x_i \mid z_i = k, \mu_k, \Sigma_k)}{\sum_{k=1}^{K} \sum_{i=1}^{N} I[z_i = k]} \right) \]

If we assume we are at step \( t+1 \), and we know our parameters values at step \( t \):
GMM

\[
E_{\Theta^t_x, \pi^t, \Sigma^t_x} [L(x, z)] = \sum_{k=1}^{K} \sum_{i=1}^{N} E_{z_i \mid x_i, z} [I[z_i = k]] \left[ \log \pi_k + \log p(x_i \mid \mu_k, \Sigma_k) \right]
\]

\[
E_{\Theta^t_x} [I[z_i = k]] = P(z_i = k \mid \Theta^t_x) = \pi_k^t
\]

Using Bayes rule

\[
z_k^t = \frac{P(x_i \mid z_i = k, \mu_k^t, \Sigma_k^t) P(z_i = k \mid \pi_k^t)}{\sum_{k=1}^{K} P(x_i \mid z_i = k, \mu_k^t, \Sigma_k^t) P(z_i = k \mid \pi_k^t)} = \frac{N(\mu_k^t, \Sigma_k^t, \pi_k^t)}{\sum_{k=1}^{K} N(\mu_k^t, \Sigma_k^t, \pi_k^t)}
\]
$$Q_{\text{olot}} = \sum_{k=1}^{K} \sum_{i=1}^{N} Z_{ki} \left[ \log \pi_k + \log N(\mu_k, \Sigma_k) \right]$$

M-step: We have an extra condition that \( \sum_{k=1}^{K} \pi_k = 1 \), we have to consider this when optimizing w.r.t. \( \pi \).

$$\log N(\mu_k, \Sigma_k) = -\frac{n}{2} \log 2\pi - \frac{1}{2} \log |\Sigma| - \frac{1}{2} \text{tr} (S_k \Sigma_k^{-1})$$

$$S_k = (X - \mu_k)^T (X - \mu_k) = \frac{1}{N} \sum_{i=1}^{N} (x_i - \mu_k)^2$$
\[ \frac{\partial Q^m}{\partial \mu_k} = \mathbb{E} \left( \sum_{i=1}^{N} z_i^r (x_i - \mu_k)^\top \Sigma^{-1} (x_i - \mu_k) \right) = 0 \]

\[ \Rightarrow \frac{\partial Q^T}{\partial \mu_k} = 2 \sum_{i=1}^{N} z_i^r (x_i - \mu_k) \Sigma_k^{-1} = 0 \]

\[ \Rightarrow \sum_{i=1}^{N} z_i^k x_i - \sum_{i=1}^{N} z_i^k \mu_k = 0 \Rightarrow \mu_k = \frac{\sum_{i=1}^{N} z_i^k x_i}{\sum_{i=1}^{N} z_i^k} \]
\[
\frac{\partial \Theta}{\partial \Sigma_k} = \sum_{i=1}^{N} z_i^k \left( \frac{1}{2} \log |\Sigma_k| - \frac{1}{2} \text{tr} \left( S_i^k \Sigma_k^{-1} \right) \right) = 0
\]

\[
S_i^k = (x_i - \mu_k^{(t+1)})^T (x_i - \mu_k^{(t+1)})
\]

\[
\Rightarrow \frac{\partial \Theta}{\partial \Sigma} = \sum_{i=1}^{N} z_i^k \left( \sum_{j=1}^{K} S_{ij}^k - S_i^k \right) = 0
\]

\[
\Rightarrow \sum_{i=1}^{N} z_i^k \Sigma_k = \sum_{i=1}^{N} z_i^k S_i^k \Rightarrow \Sigma_k^* = \frac{\sum_{i=1}^{N} z_i^k S_i^k}{\sum_{i=1}^{N} z_i^k}
\]
\[ \frac{\partial \pi_k}{\partial \pi_k} = \frac{\partial}{\partial \pi_k} \sum_{i=1}^{K} \pi_k (\log \pi_k) = 0 \quad \text{s.t.} \quad \sum_{k=1}^{K} \pi_k = 1 \]

\[ \Rightarrow \frac{\partial \pi_k}{\partial \pi_k} = \sum_{i=1}^{N} \pi_k \left[ \sum_{i=1}^{N} \pi_k (\log \pi_k) + \lambda \left( 1 - \sum_{k=1}^{K} \pi_k \right) \right] = 0 \]

\[ \Rightarrow \frac{\partial \pi_k}{\partial \pi_k} = \frac{\sum_{i=1}^{N} \pi_k}{\sum_{k=1}^{K} \pi_k} - \lambda = 0 \Rightarrow \pi_k = \frac{\sum_{i=1}^{N} \pi_k}{\sum_{k=1}^{K} \pi_k} \]

\[ \sum_{k=1}^{K} \pi_k = 1 \Rightarrow \sum_{k=1}^{K} \frac{\sum_{i=1}^{N} \pi_k}{\sum_{k=1}^{K} \pi_k} = 1 \Rightarrow \lambda = \frac{\sum_{k=1}^{K} \sum_{i=1}^{N} \pi_k}{\sum_{k=1}^{K} \pi_k} \]
EM for Semi-supervised learning

- Binary Naïve Bayes Classifier

\[
\text{Let } (x_i, y_i) \text{ be sample data, } i \in \{1 \ldots N\}, y_i \in \{1 \ldots C\}
\]
\[
x^d = (x_{i1}, x_{i2}, \ldots, x_{id}), \ x_{ij} \in \{0, 1\}, \ j \in \{1, d\}
\]

parameters \((\pi, \Theta)\), \(P(y = c|x) = \pi_c\), \(P(x_{ij} = 1|y_i = c, \Theta) = \Theta_{jc}\)

\[
P(x, y|\theta, \pi) = P(y_i|\pi) \prod P(x_{ij}|\theta_j)
\]

Likelihood: \(P(X, Y|\Theta, \pi) = \prod \prod I(y_i = c) \prod \theta_{jc}^{x_{ij}|c} (1 - \theta_j)^{1-x_{ij}} I(y_i \neq c)\)

\[
\Theta_{jc} = \frac{N_{jc}}{N_c}, \ \Theta_c = \frac{N_c}{N}, \ N_c = \sum I(y_i = c), \ N_{jc} = \sum I[x_i = c, x_{ij} = 1]
\]
EM for semi-supervised learning

- Now assume we have small labeled data set \( \{X_L, y_L\} \) and a large unlabelled data set \( \{X_U\} \).
- Assume all variables are binary and we want to use Naïve bayes for classification
- Let \( N \) be the size of labelled set and \( M \) of unlabelled
- Parameter set: \( \Theta = \{ \theta_1, \theta_2, \theta_1, \theta_2, \ldots, \theta_d, \theta_d \} \)
- We want to drive EM algorithms step which treat \( y_U \) as hidden variables.
\[ P(y_L, x_L, y_u, x_u) = \prod_{i=1}^{N} P[x_i, y_i | \theta] \prod_{m=1}^{M} P[y_m, x_m | \theta] \]

We don't know \( y_u \). So treat it as hidden variables and marginalize over all \( y_u \).

\[ P(y_L, x_L, x_u) = \sum_{y_1} \sum_{y_2} \cdots \sum_{y_M} \prod_{m=1}^{N} P(y_i, x_i | \theta) \prod_{m=1}^{M} P[y_m, x_m | \theta] \]
\[ E - \text{step:} \]
\[ l = \mathbb{P}(y_t, x_t, X_n | \theta) = \prod_{i=1}^{N} \mathbb{P}(y_i, x_i | \theta) \prod_{m=1}^{M} \left( \sum_{y_c \in \{0, 1\}} \mathbb{P}(y_m, x_m | \theta) \right) \]

\[ \log L = \sum_{i=1}^{N} \log \mathbb{P}(y_i, x_i | \theta) + \sum_{m=1}^{M} \log \sum_{y_m \in \{0, 1\}} \mathbb{P}(y_m, x_m | \theta) \]

\[ \geq \sum_{i=1}^{N} \log \mathbb{P}(y_i, x_i | \theta) + \sum_{m=1}^{M} \sum_{y_m \in \{0, 1\}} \mathbb{P}(y_m | x_m, \theta)^t \log \mathbb{P}(y_m, x_m | \theta) \]

\[ = Q(\theta | \theta^t) \quad \text{at step } t! \]
\[ x_m^{t} = P(y_m = 0 \mid x_m, \theta^t) = \frac{P(x_m \mid y_m = 0, \theta^t) P(y_m = 0 \mid \theta^t)}{\sum_{y_m \in \{0,1\}} P(x_m \mid y_m, \theta^t) P(y_m \mid \theta^t)} \]

We can define \( r_m^{t} \) similarly:

\[ P(y_i \mid x_i, \theta^t) dP(y_i, x_i \mid \theta^t) = P(y_i \mid \theta^t) \prod_{j=1}^d P(x_{ij} \mid y_i, \theta^t) \]

\[ = (\theta_{ij}^t)^{y_i} (1 - \theta_{ij}^t)^{1-y_i} \prod_{j=1}^d (\theta_{y_{ij}}^t)^{x_{ij}} (1 - \theta_{y_{ij}}^t)^{1-x_{ij}} \]
M-STEP:

For a single point we have:

\[ \log p(y_i, x_i; \Theta) = \log p(y_i; \theta) + \sum_{i=1}^{d} \log (p(x_{ij} \mid y_i, \theta_{x_{ij}})) \]

\[ \frac{\partial \log p(y_i, x_i; \Theta)}{\partial \theta_1} = \frac{\partial \log p(y_i; \theta)}{\partial \theta_1} + \frac{2c}{\Theta_1} \sim 0 \]

\[ = \frac{y_i}{\theta} - \frac{1-y_i}{1-\theta} \]

\[ = \frac{y_i}{\theta_1} - \frac{1-y_i}{1-\theta_1} \]
Now we take derivative of $Q(\theta, \theta^t)$ w.r.t. $\theta_t$,

\[
\frac{\partial Q}{\partial \theta_t} = \sum_{i=1}^{N} \frac{\partial \log P(y_i; x_i; \theta)}{\partial \theta_t} + \sum_{i=1}^{N} \sum_{y_i \in \{0, 1\}} \frac{\partial y_i^t \log P(y_i = y, x_i; \theta)}{\partial \theta_t}
\]

\[
+ \sum_{m=1}^{M} \frac{\partial \log P(y_m = 0, x_m; \theta)}{\partial \theta_t}
\]

\[
+ \sum_{m=1}^{M} \sum_{y_m \in \{0, 1\}} \frac{\partial y_m^t \log P(y_m = y, x_m; \theta)}{\partial \theta_t}
\]

\[
+ \sum_{m=1}^{M} \frac{\partial \log P(y_m = 1, x_m; \theta)}{\partial \theta_t}
\]
\[
\frac{\partial Q}{\partial \theta_1} = \sum_{i=1}^{N} \left[ \frac{y_i}{\theta_1} - \frac{1-y_i}{1-\theta_1} \right] + \sum_{m=1}^{M} x_{i0}^t \left[ \frac{\theta_0}{\theta_1} - \frac{1-\theta_0}{1-\theta_1} \right] + \sum_{x=1}^{T} \left[ \frac{1}{\theta_1} - \frac{1-1}{-1-\theta_1} \right]
\]

\[
= \sum_{i=1}^{N} \left[ \frac{y_i}{\theta_1} - \frac{1-y_i}{1-\theta_1} \right] - \sum_{m=1}^{M} \frac{x_{i0}^t}{1-\theta_1} + \sum_{x=1}^{T} \frac{x_{i0}^t}{\theta_1}
\]

\[
= \frac{\sum_{i=1}^{N} y_i + \sum_{m=1}^{M} x_{i0}^t}{\theta_1} - \frac{\sum_{i=1}^{N} (1-y_i)}{1-\theta_1} + \sum_{x=1}^{T} \frac{x_{i0}^t}{\theta_1}
\]

\[
= \frac{N_1 + R_{1i}^t}{\theta_1} - \frac{N_0 + R_{0i}^t}{1-\theta_1}
\]

\[
N_j = \sum_{i=1}^{N} I[y_i = j] \quad R_j = \sum_{m=1}^{M} x_{mj}^t \quad \delta \in \{0, 1\}
\]
Setting $\frac{\partial Q}{\partial \theta} = 0$ we get

\[
\frac{\theta_i}{1 - \theta_i} = \frac{N_1 + R_i^t}{N_0 + R_0^t} \quad \Rightarrow \quad \theta_i = \frac{N_1 + R_i^t}{N_0 + N_1 + R_i^t + R_0^t} = \frac{N_1 + R_i}{N + M}
\]
\[
\frac{\partial \log p(x_i, y_i; \Theta)}{\partial \theta_{ij}} = \frac{\partial \log p(y_i; \Theta)}{\partial \theta_{ij}} + \sum_{x_i} \frac{\partial \log p(x_{ij} | y_i, \Theta)}{\partial \theta_{ij}}
\]

\[
= \frac{\partial \log p(x_{ij} | y_i; \Theta)}{\partial \theta_{ij}} = \log \theta_{ij}^{x_i I[y_i = 1]} (1 - \theta_{ij})^{1 - x_i I[y_i = 1]}
\]

\[
= \sum_{x_i} \left\{ x_i y_i \log \theta_{ij} + (1 - x_i y_i) \log (1 - \theta_{ij}) \right\}
\]
\[ \frac{\partial \log P(x, y; \theta)}{\partial \theta_{ij}} = \frac{x_i y_i}{\theta_{ij}} - \frac{1 - x_i y_i}{1 - \theta_{ij}} \]

Similarly, we can do for \( \theta_{0j} \):

\[ \frac{\partial \log P(x, y; \theta)}{\partial \theta_{0j}} = \frac{x_i (1 - y_i)}{\theta_{0j}} - \frac{1 - x_i (1 - y_i)}{1 - \theta_{0j}} \]
Similar to \( \theta_1 \) if we compute \( \frac{\partial \theta_{ij}}{\partial \theta_{ij}} \) and \( \frac{\partial \theta}{\partial \theta_{ij}} \) and set it to zero we get:

\[
\theta_{ij} = \frac{N_i^j + R_i^j}{N_i + R_i}, \quad \theta_{oij} = \frac{N_{o1}^j + R_{o1}^j}{N_{o1} + R_{o1}}
\]

\[
N_{ii} = \sum_{i=1}^{N} I(y_i = 1) I(x_{ij} = 1), \quad N_{oi} = \sum_{i=1}^{N} I(y_i = 0) I(x_{ij} = 1)
\]

\[
R_{ii} = \sum_{i=1}^{M} r_{i1} I(x_{ij} = 1), \quad R_{o1} = \sum_{i=1}^{M} r_{o1} I(x_{ij} = 1)
\]
Robust PCA

• Suppose we are given a large data matrix $M$ and we may know it can decompose it to a low rank matrix $L$ and a sparse matrix $S$:

$$M = L + S \quad M \in \mathbb{R}^{m \times n}$$

• So the question is how can we compute $L$ and $S$ in a tractable manner?
Some Application

- Video Surveillance: Given a sequence of surveillance video frames, we often need to identify activities that stand out from the background. If we stack the video frames as columns of a matrix $M$, then the low-rank component $L$ naturally corresponds to the stationary background and the sparse component $S$ captures the moving objects in the foreground. However, each image frame has thousands or tens of thousands of pixels, and each video fragment contains hundreds or thousands of frames.
• Face Recognition: Images of a human’s face can be well-approximated by a low-dimensional subspace. Being able to correctly retrieve this subspace is crucial in many applications such as face recognition and alignment. However, realistic face images often suffer from self-shadowing, specularities, or saturations in brightness, which make this a difficult task and subsequently compromise the recognition performance.
More application

• Latent Semantic Indexing. Web search engines often need to analyze and index the content of an enormous corpus of documents. A popular scheme is the Latent Semantic Indexing (LSI). The basic idea is to gather a document-versus-term matrix $M$ whose entries typically encode the relevance of a term (or a word) to a document such as the frequency it appears in the document (e.g. the TF/IDF). PCA (or SVD) has traditionally been used to decompose the matrix as a low-rank part plus a residual, which is not necessarily sparse (as we would like). If we were able to decompose $M$ as a sum of a low-rank component $L$ and a sparse component $S$, then $L$ could capture common words used in all the documents while $S$ captures the few key words that best distinguish each document from others.
Problem formulation as optimization

• Traditional PCA

\[
\begin{align*}
\min & \quad \| M - L \|_F \\
\text{s.t.} & \quad \text{rank}(L) \leq k
\end{align*}
\]

• Robust PCA

\[
\begin{align*}
\min & \quad \| L \|_* + \lambda \| S \|_1 \\
\text{s.t.} & \quad M = L + S
\end{align*}
\]

\[
\| L \|_* := \sum_{i} \sigma_i^2(L), \quad \sigma_i \geq 0
\]

\text{singular value of } L
Singular Value Decomposition

Let \( M \in \mathbb{R}^{m \times n} \) and \( n \leq m \)

We can decompose \( M \) in a way s. t.

\[
M = U \Sigma V^T \quad \text{and} \quad V^T V = I, \quad U^T U = I
\]

\( U \in \mathbb{R}^{m \times m}, \Sigma \in \mathbb{R}^{m \times n} \), \( V \in \mathbb{R}^{n \times n} \)

\( \Sigma \) is a diagonal matrix of \( \sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \ldots \geq 0 \)
Robust PCA optimization

• The objective and constraint are convex 😊
• Multiple ways to formulate it
  • Two possible one:

\[
1 - \min \|L\|_\alpha + \lambda \|S\|_1 + \langle Y, M - S - L \rangle + \frac{1}{2\mu} \|M - L - F\|_F^2
\]

2-or simpler one:

\[
\min \|L\|_\alpha + \lambda \|S\|_1 + \frac{1}{2\mu} \|M - L - F\|_F^2
\]

Assignment is about this one!
Solving the optimization problem

• Finding the lambda:
  
  \[ \lambda = \frac{1}{\sqrt{0.1}} \]

• Block coordinate descent:
  
  If we have \( \min_{w,v} g(w) + f(v) + h(w,v) \)

  we can optimize w.r.t. each \( w \) & \( v \) separately and

  use a new value of one to update the other one!
Solving the optimization problem

\[
\min_{w,v} f(v) + g(w) + h(v, w)
\]

\[
w^{t+1} = \operatorname{arg\,min}_w g(w) + h(v^t, w)
\]

\[
v^{t+1} = \operatorname{arg\,min}_v f(v) + h(v, w^{t+1})
\]

So in Assignment Q4 you can update S and L in 2 steps!
• Now the problem is dealing with nuclear norm and L1 norm
  • For updating S we just need to deal with L1 norm and like previous assignment we use soft-threshold:

\[ S_{i}^{t+1} = \max_{x_i} \| S_{i}^{t} - \alpha \cdot (M - \hat{L}^{t} - S^{t}) \|_{1} \]
• For updating L with nuclear norm, let’s consider the following problem:

\[
L^{n+1} = \text{prox}_{\frac{1}{2} \lambda} (L^n) = \arg\min_L \left\{ \frac{1}{2} \| L - L^n \|^2_F + \lambda \| L \|_* \right\}
\]

• It defines proximal operator in matrix domain with Frobenius norm and nuclear norm!
In-exact intuition and solution!

- Let $L^t = U \Sigma^t V^T$ and we want to find $L^*$ s.t. $L^* = U \Sigma^t V^T$, $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_t)$.

So:

$$\arg \min_L \{ \| L - L^t \|_F^2 + \lambda \| L \|_F \}$$

$$= \arg \min_\Sigma \left\{ \sum_{i=1}^t \| \Sigma_i - \Sigma_i^t \|_2^2 + \lambda \sum \sigma_i \right\}$$
• Now we can optimize w.r.t. each $\sigma_i^{-}$

$$\text{arg min} \left\{ \frac{1}{2} \| \sigma_i - \sigma_i^{-} \|^2 + \lambda \sigma_i \right\}$$

taking gradient w.r.t. $\sigma_i$:

$$\sigma_i - \sigma_i^{-} + \lambda = 0 \Rightarrow \sigma_i = \sigma_i^{-} + \lambda$$

but $\sigma_i \geq 0 \Rightarrow \sigma_i^{-} = \max \left\{ 0, \sigma_i^{-} + \lambda \right\}$

It is similar to soft-threshold but on $\sigma_i^{-}$. 
• We can recover the $L$:

$$L^{t+1} = U \Sigma^{t+1} V^T$$
Exact solution for $L$

- All objective functions are convex
- Nuclear norm is convex in matrix domain!

\[
L^* = \arg \min \left\{ \frac{1}{2} \| L - L^T \|_F^2 + \lambda \| L \|_* \right\}
\]

Optimality condition:

\[
0 \in L^* - L^T + \lambda \text{arg} \min \| L^* \|_*
\]

(subgradient of nuclear norm)
Lemma for sub-gradient in matrix space

If $X = UV^T$ and $X \in \mathbb{R}^{m \times n}$,

$\mathcal{N}_{\| \cdot \|_F} \left\{ UV^T W : W \in \mathbb{R}^{m \times n}, U^T W = 0, W V = 0, \| W \|_F \leq 1 \right\}$

Claim: If $L^t = U_0 \Sigma_0 V_0^T + U_1 \Sigma_1 V_1^T$, where $U_0, \Sigma_0$ (resp. $U_1, \Sigma_1$) are singular vectors corresponding with singular value $> \lambda$ (resp. $\leq \lambda$). we have

$L^t = U_0 (\Sigma_0 - \lambda I) V_0^T, W = \lambda^{-1} U_1 \Sigma_1 V_1^T$
\[ 0 = L^* - L^T + \Theta 1_\lambda^* 1_{\lambda^*} \]

\[ L^* - L^T = \lambda U_0 V_0^T + U_1 \Sigma_1^T V_1^T = \lambda (U_0 V_0^T W) \]

\[ U_0^T W = 0, \quad W V_0 = 0, \quad \text{arg min} (\Sigma^t) \leq \lambda \Rightarrow \| W \| \leq 1 \]

So: \[ L^* - L^* \in \lambda \Theta 1_\lambda^* 1_{\lambda^*} \]

Note: \[ L^* = U_0 (\Sigma_0 - \lambda I) V_0^T = U (\Sigma^t - \lambda I) V \]

So: \[ L^* = U \left( \text{soft-threshold} (...) \right) V \]
• Termination Condition:
  • Noise On Signal Ratio (NOSR)

\[
\text{NOSR} = \frac{\| M - L \cdot S \|_F}{\| M \|_F} \leq \delta
\]
Fun Time 😊

• Image segmentation
  • You are given an image and asked as an AI expert to segment the image based on the meaningful object existing in the image.

• How do you do that?
Mean-Shift Clustering

• The problem is a clustering problem so we are looking for a clustering algorithm.
• K-means is a possible answer but you need to know the means and number of cluster already which makes it hard to apply for any arbitrary Image
• An alternative algorithm is Mean-Shift clustering.
  • Basic Idea
  • Estimate a density over the data point using kernel density estimation
    • For this you need to pick kernel parameter: for example bandwidth
  • Find the pick or modes of the density as clusters mean or center.
  • Assign each point to the closest or appropriate center
Mean-Shift Clustering

- Gaussian Kernel Density Estimation
Mean-Shift Clustering

• How can we find modes?
• This is the KDE (assuming the kernel is normalized)

\[ f(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K \left( \frac{x - x_i}{h} \right) 
K(x) = c_{k,d} k(\|x\|^2) \]

• To find peaks or modes as usual take gradient and set it to 0!

\[ \nabla f(x) = \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^{n} (x_i - x)g \left( \frac{\|x - x_i\|}{h} \right) \]
\[ = \frac{2c_{k,d}}{nh^{d+2}} \left[ \sum_{i=1}^{n} g \left( \frac{\|x - x_i\|}{h} \right) \right] \left[ \frac{\sum_{i=1}^{n} x_i g \left( \frac{\|x - x_i\|}{h} \right)}{\sum_{i=1}^{n} g \left( \frac{\|x - x_i\|}{h} \right)} - x \right] \]
Mean-Shift Clustering

• Mean shift vector $m(x)$ always moves toward increasing $f(x)$

$$m_h(x) = \frac{\sum_{i=1}^{n} x_i g \left( \frac{\|x-x_i\|}{h} \right)}{\sum_{i=1}^{n} g \left( \frac{\|x-x_i\|}{h} \right)} - x$$

• Mean shift procedure
  • Compute $m_h(x^t)$
  • Move:
    $$x^{t+1} = x^t + \alpha m_h(x^t)$$

Do this for all points!
Mean-Shift Clustering

• Cluster each point based on the mode that point moves toward!
• We can use kernel methods when the data sample size is small and our feature set or base function set is huge. But in big data era we don’t have a “small” sample size. How can we use kernels in this settings?

• Think about it and we may discus about it in my next tutorial or maybe Mark will do in class!