Mean-Shift Clustering Algorithm

Jongha "Jon" Ryu

Department of ECE
University of California, San Diego

November 6, 2018
Mean-Shift Clustering Algorithm

- [Fukunaga and Hostetler, 1975]
Mean-Shift Clustering Algorithm

- [Fukunaga and Hostetler, 1975]
- Density based clustering algorithm
Mean-Shift Clustering Algorithm

- [Fukunaga and Hostetler, 1975]
- Density based clustering algorithm
- Cluster centers ← modes of the underlying density

Fig. 1. Gradient mode clustering.
Mean-Shift Clustering Algorithm

- [Fukunaga and Hostetler, 1975]
- Density based clustering algorithm
- Cluster centers $\leftarrow$ modes of the underlying density

![Gradient mode clustering](image)

- Iterative algorithm
Mean-Shift Clustering Algorithm (ver. 1)

- Data $S = \{X_1, \ldots, X_n\} \subset \mathbb{R}^d$

Want to assign a cluster to a query point $y \in \mathbb{R}^d$

Iterate until convergence:

$$y_{m+1}(y) := \operatorname{msh}(y, S) := \frac{1}{\|y - x\|} \sum_{j : X_j \in B(y, h)} X_j$$

where $B(y, h) := \{x : \|x - y\| \leq h\}$, $l_2$-norm ball
Mean-Shift Clustering Algorithm (ver. 1)

- Data $S = \{X_1, \ldots, X_n\} \subset \mathbb{R}^d$
- Want to assign a cluster to a query point $y \in \mathbb{R}^d$
Mean-Shift Clustering Algorithm (ver. 1)

- Data $S = \{X_1, \ldots, X_n\} \subset \mathbb{R}^d$
- Want to assign a cluster to a query point $y \in \mathbb{R}^d$
- Iterate until convergence:

$$
y \leftarrow m_h(y) := m_h(y; S) := \frac{1}{|\{j: X_j \in B(y, h)\}|} \sum_{j: X_j \in B(y, h)} X_j$$

where $B(y, h) := \{x: ||x - y|| \leq h\}$, $l_2$-norm ball

$$
(1)
$$
Mean-Shift Clustering Algorithm (ver. 1)

- Data $S = \{X_1, \ldots, X_n\} \subset \mathbb{R}^d$
- Want to assign a cluster to a query point $y \in \mathbb{R}^d$
- Iterate until convergence:

$$y \leftarrow m_h(y) := m_h(y; S) := \frac{1}{\left| \{j: X_j \in B(y, h)\} \right|} \sum_{j: X_j \in B(y, h)} X_j$$  \hspace{1cm} (1)

where $B(y, h) := \{x: \|x - y\| \leq h\}$, $l_2$-norm ball
Mean-Shift Clustering Algorithm (ver. 1)

- **Data** $S = \{X_1, \ldots, X_n\} \subset \mathbb{R}^d$
- Want to assign a cluster to a query point $y \in \mathbb{R}^d$
- Iterate until convergence:

$$y \leftarrow m_h(y) := m_h(y; S) := \frac{1}{|\{j: X_j \in B(y, h)\}|} \sum_{j: X_j \in B(y, h)} X_j$$  \hspace{1cm} (1)

where $B(y, h) := \{x: \|x - y\| \leq h\}$, $l_2$-norm ball
Mean-Shift Clustering Algorithm (ver. 1)

- Data $S = \{X_1, \ldots, X_n\} \subset \mathbb{R}^d$
- Want to assign a cluster to a query point $y \in \mathbb{R}^d$
- Iterate until convergence:

$$y \leftarrow m_h(y) := m_h(y; S) := \frac{1}{|\{j: X_j \in B(y, h)\}|} \sum_{j: X_j \in B(y, h)} X_j$$  \hspace{1cm} (1)

where $B(y, h) := \{x: \|x - y\| \leq h\}$, $l_2$-norm ball
Mean-Shift Clustering Algorithm (ver. 1)

- Data $S = \{X_1, \ldots, X_n\} \subset \mathbb{R}^d$
- Want to assign a cluster to a query point $y \in \mathbb{R}^d$
- Iterate until convergence:

$$y \leftarrow m_h(y) := m_h(y; S) := \frac{1}{|\{j: X_j \in B(y, h)\}|} \sum_{j: X_j \in B(y, h)} X_j \quad (1)$$

where $B(y, h) := \{x: \|x - y\| \leq h\}$, $l_2$-norm ball
Mean-Shift Clustering Algorithm (ver. 1)

- Data $S = \{X_1, \ldots, X_n\} \subset \mathbb{R}^d$
- Want to assign a cluster to a query point $y \in \mathbb{R}^d$
- Iterate until convergence:

$$y \leftarrow m_h(y) := m_h(y; S) := \frac{1}{|\{j: X_j \in B(y, h)\}|} \sum_{j: X_j \in B(y, h)} X_j \quad (1)$$

where $B(y, h) := \{x : \|x - y\| \leq h\}$, $l_2$-norm ball
Mean-Shift Clustering Algorithm (ver. 1)

- Data $S = \{X_1, \ldots, X_n\} \subset \mathbb{R}^d$
- Want to assign a cluster to a query point $y \in \mathbb{R}^d$
- Iterate until convergence:

$$y \leftarrow m_h(y) := m_h(y; S) := \frac{1}{|\{j: X_j \in B(y, h)\}|} \sum_{j: X_j \in B(y, h)} X_j$$

where $B(y, h) := \{x: \|x - y\| \leq h\}$, $l_2$-norm ball

$$i=5$$
Mean-Shift Clustering Algorithm (ver. 1)

- Data \( S = \{X_1, \ldots, X_n\} \subset \mathbb{R}^d \)
- Want to assign a cluster to a query point \( y \in \mathbb{R}^d \)
- Iterate until convergence:

\[
y \leftarrow m_h(y) := m_h(y; S) := \frac{1}{|\{j: X_j \in B(y, h)\}|} \sum_{j: X_j \in B(y, h)} X_j
\]

where \( B(y, h) := \{x: \|x - y\| \leq h\} \), \( l_2 \)-norm ball

\[i=6\]

Jon (UCSD)
Mean-Shift Clustering Algorithm (ver. 1)

- Data \( S = \{\mathbf{X}_1, \ldots, \mathbf{X}_n\} \subset \mathbb{R}^d \)
- Want to assign a cluster to a query point \( y \in \mathbb{R}^d \)
- Iterate until convergence:

\[
y \leftarrow m_h(y) := m_h(y; S) := \frac{1}{|\{j: \mathbf{X}_j \in B(y, h)\}|} \sum_{j: \mathbf{X}_j \in B(y, h)} \mathbf{X}_j
\]

where \( B(y, h) := \{\mathbf{x}: \|\mathbf{x} - y\| \leq h\} \), \( l_2 \)-norm ball

![Image of data points and mean-shift algorithm](image.png)
Mean-Shift Clustering Algorithm (ver. 1)

- Data \( S = \{X_1, \ldots, X_n\} \subset \mathbb{R}^d \)
- Want to assign a cluster to a query point \( y \in \mathbb{R}^d \)
- Iterate until convergence:

\[
y \leftarrow m_h(y) := m_h(y; S) := \frac{1}{\left| \{j: X_j \in B(y, h)\} \right|} \sum_{j: X_j \in B(y, h)} X_j \quad (1)
\]

where \( B(y, h) := \{x: \|x - y\| \leq h\}, \) \( l_2 \)-norm ball
Mean-Shift Clustering Algorithm (ver. 1)

- Data $S = \{X_1, \ldots, X_n\} \subset \mathbb{R}^d$
- Want to assign a cluster to a query point $y \in \mathbb{R}^d$
- Iterate until convergence:

\[
y \leftarrow m_h(y) := m_h(y; S) := \frac{1}{|\{j: X_j \in B(y, h)\}|} \sum_{j: X_j \in B(y, h)} X_j
\]

where $B(y, h) := \{x: \|x - y\| \leq h\}$, $l_2$-norm ball

![Image of data points and mean-shift process]
Mean-Shift Clustering Algorithm (ver. 1)

- Data \( S = \{X_1, \ldots, X_n\} \subset \mathbb{R}^d \)
- Want to assign a cluster to a query point \( y \in \mathbb{R}^d \)
- Iterate until convergence:

\[
y \leftarrow m_h(y) := m_h(y; S) := \frac{1}{|\{j: X_j \in B(y, h)\}|} \sum_{j: X_j \in B(y, h)} X_j
\]

where \( B(y, h) := \{x: \|x - y\| \leq h\} \), \( l_2 \)-norm ball

\[\text{(1)}\]
Mean-Shift Clustering Algorithm (ver. 2)

- Data $S = \{X_1, \ldots, X_n\} \subset \mathbb{R}^d$
Mean-Shift Clustering Algorithm (ver. 2)

- Data $S = \{X_1, \ldots, X_n\} \subset \mathbb{R}^d$
- Update $S$ along the iterations; $T^{(0)} = \{t_1^{(0)}, \ldots, t_n^{(0)}\} \leftarrow S$
Mean-Shift Clustering Algorithm (ver. 2)

- Data \( S = \{X_1, \ldots, X_n\} \subset \mathbb{R}^d \)
- Update \( S \) along the iterations; \( T^{(0)} = \{t^{(0)}_1, \ldots, t^{(0)}_n\} \leftarrow S \)
- Iterate for \( i = 1, \ldots, \) until convergence:

\[
\begin{align*}
    t^{(i)}_k &\leftarrow m_h(t^{(i-1)}_k; T^{(i-1)}), \quad k = 1, \ldots, n \\
    T^{(i)} &\leftarrow \{t^{(i)}_1, \ldots, t^{(i)}_n\}
\end{align*}
\]
Mean-Shift Clustering Algorithm (ver. 2)

- Data $S = \{X_1, \ldots, X_n\} \subset \mathbb{R}^d$
- Update $S$ along the iterations; $T^{(0)} = \{t_1^{(0)}, \ldots, t_n^{(0)}\} \leftarrow S$
- Iterate for $i = 1, \ldots$, until convergence:
  \[
  t_k^{(i)} \leftarrow m_h(t_k^{(i-1)}; T^{(i-1)}), \quad k = 1, \ldots, n
  \]
  \[
  T^{(i)} \leftarrow \{t_1^{(i)}, \ldots, t_n^{(i)}\}
  \]
- So-called blurring process
Mean-Shift Clustering Algorithm (ver. 2)

- Data $S = \{X_1, \ldots, X_n\} \subset \mathbb{R}^d$
- Update $S$ along the iterations; $T^{(0)} = \{t_1^{(0)}, \ldots, t_n^{(0)}\} \leftarrow S$
- Iterate for $i = 1, \ldots$, until convergence:

$$\begin{align*}
  t_k^{(i)} & \leftarrow m_k(t_k^{(i-1)}; T^{(i-1)}), \quad k = 1, \ldots, n \\
  T^{(i)} & \leftarrow \{t_1^{(i)}, \ldots, t_n^{(i)}\}
\end{align*}$$

- So-called blurring process

![Diagram](image)
Mean-Shift Clustering Algorithm (ver. 2)

- Data \( S = \{X_1, \ldots, X_n\} \subset \mathbb{R}^d \)
- Update \( S \) along the iterations; \( T^{(0)} = \{t_1^{(0)}, \ldots, t_n^{(0)}\} \leftarrow S \)
- Iterate for \( i = 1, \ldots \), until convergence:
  \[
  t_k^{(i)} \leftarrow m_h(t_k^{(i-1)}; T^{(i-1)}), \quad k = 1, \ldots, n
  \]
  \[
  T^{(i)} \leftarrow \{t_1^{(i)}, \ldots, t_n^{(i)}\}
  \]
- So-called *blurring process*

![Plot of Mean-Shift Clustering](image)
Mean-Shift Clustering Algorithm (ver. 2)

- Data \( S = \{X_1, \ldots, X_n\} \subset \mathbb{R}^d \)
- Update \( S \) along the iterations; \( T^{(0)} = \{t_1^{(0)}, \ldots, t_n^{(0)}\} \leftarrow S \)
- Iterate for \( i = 1, \ldots, \) until convergence:
  \[
  t_k^{(i)} \leftarrow m_h(t_k^{(i-1)}; T^{(i-1)}), \quad k = 1, \ldots, n
  \]
  \[
  T^{(i)} \leftarrow \{t_1^{(i)}, \ldots, t_n^{(i)}\}
  \]
- So-called blurring process
Mean-Shift Clustering Algorithm (ver. 2)

- Data $S = \{X_1, \ldots, X_n\} \subset \mathbb{R}^d$
- Update $S$ along the iterations; $T^{(0)} = \{t_1^{(0)}, \ldots, t_n^{(0)}\} \leftarrow S$
- Iterate for $i = 1, \ldots$, until convergence:
  \[
  t_k^{(i)} \leftarrow m_h(t_k^{(i-1)}; T^{(i-1)}) , \quad k = 1, \ldots, n
  \] 
  \[
  T^{(i)} \leftarrow \{t_1^{(i)}, \ldots, t_n^{(i)}\}
  \]

- So-called \textit{blurring process}
Mean-Shift Clustering Algorithm (ver. 2)

- Data $S = \{X_1, \ldots, X_n\} \subset \mathbb{R}^d$
- Update $S$ along the iterations; $T^{(0)} = \{t_1^{(0)}, \ldots, t_n^{(0)}\} \leftarrow S$
- Iterate for $i = 1, \ldots$, until convergence:
  \[
  t_k^{(i)} \leftarrow m_h(t_k^{(i-1)}; T^{(i-1)}), \quad k = 1, \ldots, n
  \]
  \[
  T^{(i)} \leftarrow \{t_1^{(i)}, \ldots, t_n^{(i)}\}
  \]
- So-called *blurring process*
Mean-Shift Clustering Algorithm (ver. 2)

- Data $S = \{\mathbf{X}_1, \ldots, \mathbf{X}_n\} \subset \mathbb{R}^d$
- Update $S$ along the iterations; $T^{(0)} = \{\mathbf{t}_1^{(0)}, \ldots, \mathbf{t}_n^{(0)}\} \leftarrow S$
- Iterate for $i = 1, \ldots$, until convergence:

\[
\begin{align*}
\mathbf{t}_k^{(i)} &\leftarrow m_h(\mathbf{t}_k^{(i-1)}; T^{(i-1)}), \quad k = 1, \ldots, n \\
T^{(i)} &\leftarrow \{\mathbf{t}_1^{(i)}, \ldots, \mathbf{t}_n^{(i)}\}
\end{align*}
\]

- So-called *blurring process*
Mean-Shift Clustering Algorithm (ver. 2)

- Data $S = \{X_1, \ldots, X_n\} \subset \mathbb{R}^d$
- Update $S$ along the iterations; $T^{(0)} = \{t_1^{(0)}, \ldots, t_n^{(0)}\} \leftarrow S$
- Iterate for $i = 1, \ldots$, until convergence:
  \[ t_k^{(i)} \leftarrow m_h(t_k^{(i-1)}; T^{(i-1)}), \quad k = 1, \ldots, n \]
  \[ T^{(i)} \leftarrow \{t_1^{(i)}, \ldots, t_n^{(i)}\} \]  

- So-called \textit{blurring process}
Mean-Shift Clustering Algorithm (ver. 2)

- Data $S = \{X_1, \ldots, X_n\} \subset \mathbb{R}^d$
- Update $S$ along the iterations; $T^{(0)} = \{t_1^{(0)}, \ldots, t_n^{(0)}\} \leftarrow S$
- Iterate for $i = 1, \ldots$, until convergence:
  
  $t_{k}^{(i)} \leftarrow m_h(t_{k}^{(i-1)}; T^{(i-1)})$, $k = 1, \ldots, n$  \hspace{1cm} (2)
  
  $T^{(i)} \leftarrow \{t_1^{(i)}, \ldots, t_n^{(i)}\}$  \hspace{1cm} (3)

- So-called *blurring process*
Mean-Shift Clustering Algorithm (ver. 2)

- Data $S = \{X_1, \ldots, X_n\} \subset \mathbb{R}^d$
- Update $S$ along the iterations; $T^{(0)} = \{t_1^{(0)}, \ldots, t_n^{(0)}\} \leftarrow S$
- Iterate for $i = 1, \ldots$, until convergence:
  \[
  t_k^{(i)} \leftarrow m_n(t_k^{(i-1)}; T^{(i-1)}), \quad k = 1, \ldots, n \tag{2}
  \\
  T^{(i)} \leftarrow \{t_1^{(i)}, \ldots, t_n^{(i)}\} \tag{3}
  
- So-called blurring process

![Graph of Mean-Shift Clustering Algorithm](image.png)
Mean-Shift Clustering Algorithm (ver. 2)

- Data \( S = \{X_1, \ldots, X_n\} \subset \mathbb{R}^d \)
- Update \( S \) along the iterations; \( T^{(0)} = \{t_1^{(0)}, \ldots, t_n^{(0)}\} \leftarrow S \)
- Iterate for \( i = 1, \ldots, \) until convergence:
  \[
  t_k^{(i)} \leftarrow m_h(t_k^{(i-1)}; T^{(i-1)}), \quad k = 1, \ldots, n
  \]
  \[
  T^{(i)} \leftarrow \{t_1^{(i)}, \ldots, t_n^{(i)}\}
  \]
- So-called *blurring process*
Mean-Shift Clustering Algorithm (ver. 2)

- Data $S = \{ \mathbf{X}_1, \ldots, \mathbf{X}_n \} \subset \mathbb{R}^d$
- Update $S$ along the iterations; $T^{(0)} = \{ \mathbf{t}_1^{(0)}, \ldots, \mathbf{t}_n^{(0)} \} \leftarrow S$
- Iterate for $i = 1, \ldots$, until convergence:
  \[
  \mathbf{t}_k^{(i)} \leftarrow m_h(\mathbf{t}_k^{(i-1)}; T^{(i-1)}), \quad k = 1, \ldots, n \tag{2} \]
  \[
  T^{(i)} \leftarrow \{ \mathbf{t}_1^{(i)}, \ldots, \mathbf{t}_n^{(i)} \} \tag{3} \]

- So-called *blurring process*
Mean-Shift Clustering Algorithm (ver. 2)

- Data $S = \{X_1, \ldots, X_n\} \subset \mathbb{R}^d$
- Update $S$ along the iterations; $T^{(0)} = \{t_1^{(0)}, \ldots, t_n^{(0)}\} \leftarrow S$
- Iterate for $i = 1, \ldots$, until convergence:

$$t_k^{(i)} \leftarrow m_h(t_k^{(i-1)}; T^{(i-1)}), \quad k = 1, \ldots, n$$  \hspace{1cm} (2)

$$T^{(i)} \leftarrow \{t_1^{(i)}, \ldots, t_n^{(i)}\}$$  \hspace{1cm} (3)

- So-called **blurring process**
Mean-Shift Clustering Algorithm (ver. 2)

- Data $S = \{X_1, \ldots, X_n\} \subset \mathbb{R}^d$
- Update $S$ along the iterations; $T^{(0)} = \{t_1^{(0)}, \ldots, t_n^{(0)}\} \leftarrow S$
- Iterate for $i = 1, \ldots$, until convergence:
  \[
  t_k^{(i)} \leftarrow m_h(t_k^{(i-1)}; T^{(i-1)}), \quad k = 1, \ldots, n
  \]
  (2)
  \[
  T^{(i)} \leftarrow \{t_1^{(i)}, \ldots, t_n^{(i)}\}
  \]
  (3)
- So-called *blurring process*
Mean-Shift Clustering Algorithm (ver. 2)

- Data $S = \{X_1, \ldots, X_n\} \subset \mathbb{R}^d$
- Update $S$ along the iterations; $T^{(0)} = \{t_1^{(0)}, \ldots, t_n^{(0)}\} \leftarrow S$
- Iterate for $i = 1, \ldots$, until convergence:
  \[
  t_k^{(i)} \leftarrow m_h(t_k^{(i-1)}; T^{(i-1)}), \quad k = 1, \ldots, n
  \]
  
  \[
  T^{(i)} \leftarrow \{t_1^{(i)}, \ldots, t_n^{(i)}\}
  \]

- So-called *blurring process*
Mean-Shift Clustering Algorithm (ver. 2)

- Data $S = \{X_1, \ldots, X_n\} \subset \mathbb{R}^d$
- Update $S$ along the iterations; $T^{(0)} = \{t_1^{(0)}, \ldots, t_n^{(0)}\} \leftarrow S$
- Iterate for $i = 1, \ldots$, until convergence:
  \begin{align}
  t_k^{(i)} &\leftarrow m_h(t_k^{(i-1)}; T^{(i-1)}), \quad k = 1, \ldots, n \\
  T^{(i)} &\leftarrow \{t_1^{(i)}, \ldots, t_n^{(i)}\}
  \end{align}

- So-called blurring process
Mean-Shift Clustering Algorithm (ver. 2)

- Data $S = \{X_1, \ldots, X_n\} \subset \mathbb{R}^d$
- Update $S$ along the iterations; $T^{(0)} = \{t_1^{(0)}, \ldots, t_n^{(0)}\} \leftarrow S$
- Iterate for $i = 1, \ldots$, until convergence:

$$t_k^{(i)} \leftarrow m_h(t_k^{(i-1)}; T^{(i-1)}), \quad k = 1, \ldots, n$$  \hspace{1cm} (2)

$$T^{(i)} \leftarrow \{t_1^{(i)}, \ldots, t_n^{(i)}\}$$  \hspace{1cm} (3)

- So-called blurring process

![Plot of data points after iteration i=15]
Mean-Shift Clustering Algorithm (ver. 2)

- Data $S = \{X_1, \ldots, X_n\} \subset \mathbb{R}^d$
- Update $S$ along the iterations; $T^{(0)} = \{t_1^{(0)}, \ldots, t_n^{(0)}\} \leftarrow S$
- Iterate for $i = 1, \ldots$, until convergence:
  \[
  t_k^{(i)} \leftarrow m_h(t_k^{(i-1)}; T^{(i-1)}), \quad k = 1, \ldots, n
  \]  
  
  \[
  T^{(i)} \leftarrow \{t_1^{(i)}, \ldots, t_n^{(i)}\}
  \]
- So-called \textit{blurring process}
Mean-Shift Clustering Algorithm (ver. 2)

- Data $S = \{X_1, \ldots, X_n\} \subset \mathbb{R}^d$
- Update $S$ along the iterations; $T^{(0)} = \{t_1^{(0)}, \ldots, t_n^{(0)}\} \leftarrow S$
- Iterate for $i = 1, \ldots$, until convergence:

$$t_k^{(i)} \leftarrow m_h(t_k^{(i-1)}; T^{(i-1)}), \quad k = 1, \ldots, n$$ (2)

$$T^{(i)} \leftarrow \{t_1^{(i)}, \ldots, t_n^{(i)}\}$$ (3)

- So-called *blurring process*
Mean-Shift Clustering Algorithm (ver. 2)

- Data \( S = \{ \mathbf{X}_1, \ldots, \mathbf{X}_n \} \subset \mathbb{R}^d \)
- Update \( S \) along the iterations; \( T^{(0)} = \{ t_1^{(0)}, \ldots, t_n^{(0)} \} \leftarrow S \)
- Iterate for \( i = 1, \ldots, \) until convergence:
  \[
  t_k^{(i)} \leftarrow m_h(t_k^{(i-1)}; T^{(i-1)}), \quad k = 1, \ldots, n
  \]
  \[
  T^{(i)} \leftarrow \{ t_1^{(i)}, \ldots, t_n^{(i)} \}
  \]
- So-called \textit{blurring process}

\[
\begin{align*}
  i &= 18 \\
  \text{i=18}
\end{align*}
\]
Mean-Shift Clustering Algorithm (ver. 2)

- Data $S = \{X_1, \ldots, X_n\} \subset \mathbb{R}^d$
- Update $S$ along the iterations; $T^{(0)} = \{t_1^{(0)}, \ldots, t_n^{(0)}\} \leftarrow S$
- Iterate for $i = 1, \ldots,$ until convergence:
  $$t_k^{(i)} \leftarrow m_h(t_k^{(i-1)}; T^{(i-1)}), \quad k = 1, \ldots, n$$
  $$T^{(i)} \leftarrow \{t_1^{(i)}, \ldots, t_n^{(i)}\}$$

- So-called *blurring process*
Mean-Shift Clustering Algorithm (ver. 2)

- Data $S = \{X_1, \ldots, X_n\} \subset \mathbb{R}^d$
- Update $S$ along the iterations; $T^{(0)} = \{t_1^{(0)}, \ldots, t_n^{(0)}\} \leftarrow S$
- Iterate for $i = 1, \ldots, n$, until convergence:
  \begin{align*}
  t_k^{(i)} &\leftarrow m_h(t_k^{(i-1)}; T^{(i-1)}) , \quad k = 1, \ldots, n \tag{2} \\
  T^{(i)} &\leftarrow \{t_1^{(i)}, \ldots, t_n^{(i)}\} \tag{3}
  \end{align*}
- So-called *blurring process*
Mean-Shift Clustering Algorithm

\[ \mathbf{y} \leftarrow \mathbf{y} + (m_h(\mathbf{y}; S) - \mathbf{y}) \]

Looks like gradient ascent step...

\[ m_h(\mathbf{y}; S) \]

\[ \text{mean-shift} \]
Mean-Shift Clustering Algorithm

\[ y \leftarrow y + \left( m_h(y; S) - y \right) \]  \hspace{1cm} (4)

- Looks like gradient ascent step...
- Indeed, mean-shift can be viewed as an estimate of density gradient, implying:

\[ y \leftarrow y + \nabla \log f(y) \]  \hspace{1cm} (5)
Mean-Shift Clustering Algorithm

\[ \mathbf{y} \leftarrow \mathbf{y} + (m_h(\mathbf{y}; \mathcal{S}) - \mathbf{y}) \]

(4)

- Looks like gradient ascent step...
- Indeed, mean-shift can be viewed as an estimate of density gradient, implying:

\[ \mathbf{y} \leftarrow \mathbf{y} + \nabla \log f(\mathbf{y}) \]

(5)

- The rest of the talk will focus on:
Mean-Shift Clustering Algorithm

\[ y \leftarrow y + (m_h(y; S) - y) \]  \hspace{1cm} (4)

- Looks like gradient ascent step...
- Indeed, mean-shift can be viewed as an estimate of density gradient, implying:

\[ y \leftarrow y + \nabla \log f(y) \]  \hspace{1cm} (5)

- The rest of the talk will focus on:
  1. Connection to a density gradient estimate
Mean-Shift Clustering Algorithm

\[ \mathbf{y} \leftarrow \mathbf{y} + \left( m_h(\mathbf{y}; S) - \mathbf{y} \right) \]  \hspace{1cm} (4)

- Looks like gradient ascent step...
- Indeed, mean-shift can be viewed as an estimate of density gradient, implying:

\[ \mathbf{y} \leftarrow \mathbf{y} + \hat{\nabla} \log f(\mathbf{y}) \]  \hspace{1cm} (5)

- The rest of the talk will focus on:
  1. Connection to a density gradient estimate
  2. Generalized mean-shift

Jon (UCSD)
Mean-Shift Clustering Algorithm

\[ \mathbf{y} \leftarrow \mathbf{y} + \left( m_h(\mathbf{y}; S) - \mathbf{y} \right) \]

(4)

- Looks like gradient ascent step...
- Indeed, mean-shift can be viewed as an estimate of density gradient, implying:

\[ \mathbf{y} \leftarrow \mathbf{y} + \hat{\nabla} \log f(\mathbf{y}) \]

(5)

- The rest of the talk will focus on:
  1. Connection to a density gradient estimate
  2. Generalized mean-shift
- Siva will cover: consistency of the generalized mean-shift (ver. 1) and its convergence rate
Kernel Density Estimate

- Let $f(x)$ be a density over $\mathbb{R}^d$
Kernel Density Estimate

- Let $f(x)$ be a density over $\mathbb{R}^d$
- Let $X_1, \ldots, X_n \sim \text{i.i.d. } f(x)$
Kernel Density Estimate

- Let \( f(x) \) be a density over \( \mathbb{R}^d \)
- Let \( X_1, \ldots, X_n \sim \text{i.i.d. } f(x) \)
- A kernel function \( \Phi: \mathbb{R}^d \to \mathbb{R} \) with \( \int_{\mathbb{R}^d} \Phi(x) \, dx = 1 \)
Kernel Density Estimate

Let $f(x)$ be a density over $\mathbb{R}^d$

Let $X_1, \ldots, X_n \sim$ i.i.d. $f(x)$

A kernel function $\Phi: \mathbb{R}^d \rightarrow \mathbb{R}$ with $\int_{\mathbb{R}^d} \Phi(x) \, dx = 1$

For bandwidth $h > 0$,

$$\Phi_h(u) := \frac{1}{h^d} \Phi \left( \frac{u}{h} \right) \quad (6)$$
Kernel Density Estimate

- Let $f(x)$ be a density over $\mathbb{R}^d$
- Let $X_1, \ldots, X_n \sim \text{i.i.d. } f(x)$
- A kernel function $\Phi: \mathbb{R}^d \to \mathbb{R}$ with $\int_{\mathbb{R}^d} \Phi(x) \, dx = 1$
- For bandwidth $h > 0$,

$$\Phi_h(u) := \frac{1}{h^d} \Phi \left( \frac{u}{h} \right) \quad (6)$$

- Kernel density estimate (KDE) for $f$ based on the sample $X_1, \ldots, X_n$

$$\hat{f}_{n,h}(x) := \frac{1}{n} \sum_{j=1}^{n} \Phi_h(x - X_j) \quad (7)$$
Kernel Density Estimate

- Let $f(x)$ be a density over $\mathbb{R}^d$
- Let $X_1, \ldots, X_n \sim$ i.i.d. $f(x)$
- A kernel function $\Phi : \mathbb{R}^d \to \mathbb{R}$ with $\int_{\mathbb{R}^d} \Phi(x) \, dx = 1$
- For bandwidth $h > 0$,

$$\Phi_h(u) := \frac{1}{h^d} \Phi \left( \frac{u}{h} \right)$$  

- Kernel density estimate (KDE) for $f$ based on the sample $X_1, \ldots, X_n$

$$\hat{f}_{n,h}(x) := \frac{1}{n} \sum_{j=1}^{n} \Phi_h(x - X_j)$$  

- A profile function $\phi(x) : \mathbb{R}^d \to \mathbb{R}$ is a scalar function such that

$$\Phi(x) = \phi(\|x\|^2),$$  

nonnegative, nonincreasing, piecewise continuous, and $\int_0^\infty \phi(r) \, dr < \infty$
Kernel Density Estimate (cont’d)

\[
\hat{f}_{n,h}^\phi(x) := \frac{1}{n} \sum_{j=1}^{n} \Phi_h(x - X_j) = \frac{1}{nh^d} \sum_{j=1}^{n} \phi\left( \frac{\|x - X_j\|^2}{h^2} \right)
\]
Kernel Density Estimate (cont’d)

\[ \hat{f}_{n,h}^{\phi}(x) := \frac{1}{n} \sum_{j=1}^{n} \Phi_h(x - X_j) = \frac{1}{nh^d} \sum_{j=1}^{n} \phi \left( \frac{\|x - X_j\|^2}{h^2} \right) \]

- (Gaussian kernel) \( \phi(r) = (2\pi)^{-d/2} e^{-r/2} \)
Kernel Density Estimate (cont’d)

\[ f_{n,h}(x) := \frac{1}{n} \sum_{j=1}^{n} \Phi_h(x - X_j) = \frac{1}{nh^d} \sum_{j=1}^{n} \phi \left( \frac{\|x - X_j\|^2}{h^2} \right) \]

▶ (Gaussian kernel) \[ \phi(r) = (2\pi)^{-d/2} e^{-r/2} \]
Kernel Density Estimate (cont’d)

\[ \hat{f}_{n,h}^\phi(x) := \frac{1}{n} \sum_{j=1}^{n} \Phi_h(x - X_j) = \frac{1}{nh^d} \sum_{j=1}^{n} \phi \left( \frac{\|x - X_j\|^2}{h^2} \right) \]

- (Gaussian kernel) \( \phi(r) = (2\pi)^{-d/2} e^{-r/2} \)
Kernel Density Estimate (cont’d)

\[ \hat{f}_{n,h}^{\phi}(x) := \frac{1}{n} \sum_{j=1}^{n} \Phi_h(x - X_j) = \frac{1}{nh^d} \sum_{j=1}^{n} \phi \left( \frac{\|x - X_j\|^2}{h^2} \right) \]

- (Gaussian kernel) \( \phi(r) = (2\pi)^{-d/2} e^{-r/2} \)
Kernel Density Estimate (cont’d)

\[
\hat{f}_{n,h}^\phi(x) := \frac{1}{n} \sum_{j=1}^{n} \Phi_h (x - X_j) = \frac{1}{nh^d} \sum_{j=1}^{n} \phi \left( \frac{\|x - X_j\|^2}{h^2} \right)
\]

- (Gaussian kernel) \( \phi(r) = (2\pi)^{-d/2} e^{-r/2} \)

- Note that: If \( h = h_n \to 0 \) properly as \( n \to \infty \), KDE becomes consistent
Kernel Density Estimate (cont’d)

\[
\hat{f}_{n,h}(x) := \frac{1}{n} \sum_{j=1}^{n} \Phi_h(x - X_j) = \frac{1}{nh^d} \sum_{j=1}^{n} \phi \left( \frac{\|x - X_j\|^2}{h^2} \right)
\]

- (Gaussian kernel) \( \phi(r) = (2\pi)^{-d/2} e^{-r/2} \)

- Note that: If \( h = h_n \to 0 \) properly as \( n \to \infty \), KDE becomes consistent
  - \( \lim_{n \to \infty} h_n = 0 \Rightarrow \) asymptotic unbiasedness
Kernel Density Estimate (cont’d)

\[
\hat{f}_{n,h}^\phi(x) := \frac{1}{n} \sum_{j=1}^{n} \Phi_h(x - X_j) = \frac{1}{nh^d} \sum_{j=1}^{n} \phi \left( \frac{\|x - X_j\|^2}{h^2} \right)
\]

▶ (Gaussian kernel) \( \phi(r) = (2\pi)^{-d/2} e^{-r/2} \)

▶ Note that: If \( h = h_n \to 0 \) properly as \( n \to \infty \), KDE becomes consistent
  ▶ \( \lim_{n \to \infty} h_n = 0 \Rightarrow \) asymptotic unbiasedness
  ▶ \( \lim_{n \to \infty} nh_n^d = 0 \Rightarrow \) mean-square consistency
Kernel Density Estimate (cont’d)

\[
\hat{f}_{n,h}(x) := \frac{1}{n} \sum_{j=1}^{n} \Phi_h(x - X_j) = \frac{1}{nh^d} \sum_{j=1}^{n} \phi \left( \frac{\|x - X_j\|^2}{h^2} \right)
\]

- (Gaussian kernel) \( \phi(r) = \left(2\pi\right)^{-d/2} e^{-r/2} \)

- Note that: If \( h = h_n \to 0 \) properly as \( n \to \infty \), KDE becomes consistent
  - \( \lim_{n \to \infty} h_n = 0 \) ⇒ asymptotic unbiasedness
  - \( \lim_{n \to \infty} nh_n^d = 0 \) ⇒ mean-square consistency
  - \( \lim_{n \to \infty} nh_n^{2d} = 0 \) ⇒ uniform consistency (in probability) provided that \( f(x) \) is uniformly continuous
Kernel Density Gradient Estimate

For differentiable $f$, define kernel density gradient estimate

$$\hat{\nabla}_xf_{n,h}(x)$$

(Gaussian kernel)

$$\phi(r) = \frac{2^d}{\sqrt{\pi}^d} e^{-r^2/2}$$

$$\hat{\nabla}_xf_{n,h}(x) = \frac{1}{n} \sum_{i=1}^{n} \left( X_i x \right)^{1/h} \phi(\|x-X_i\|)$$

(Epanechnikov kernel)

$$\phi(r) = c_d (1 - r^2)^{1/2}$$

$$\hat{\nabla}_xf_{n,h}(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{\left( X_i x \right)^{1/h}}{\phi(\|x-X_i\|)^{1/2}}$$
Kernel Density Gradient Estimate

For differentiable $f$, define kernel density gradient estimate

$$\nabla_x f_{n,h}(x) := \nabla_x \hat{f}_{n,h}(x)$$  \hfill (9)

$$\sum_{i=1}^{n} \frac{1}{h} \nabla_x \hat{f}_{n,h}(X_i)$$  \hfill (10)

=Gaussian kernel

$$\phi(r) = (2\pi)^d/2 e^{-r^2/2} b \nabla_x f_{\phi}(n) ; h(x) = \frac{1}{n} \sum_{i=1}^{n} (X_i x) h$$

=Epanechnikov kernel

$$\phi(r) = c_d (1-r) + c_d \max f \frac{1}{r} ; g$$
Kernel Density Gradient Estimate

For differentiable $f$, define kernel density gradient estimate

$$
\hat{\nabla}_x f_{n,h} (x) := \nabla_x \hat{f}_{n,h}(x) = \frac{1}{n} \sum_{i=1}^{n} \nabla_x \Phi_h(x - X_i)
$$

(9)

(10)
Kernel Density Gradient Estimate

For differentiable $f$, define kernel density gradient estimate

$$\hat{\nabla}_x f_{n,h}(x) := \nabla_x \hat{f}_{n,h}(x) = \frac{1}{n} \sum_{i=1}^{n} \nabla_x \Phi_h(x - X_i)$$

$$= \frac{1}{n} \sum_{i=1}^{n} (x - X_i) \frac{2}{h^{d+2}} \phi' \left( \frac{\|x - X_i\|^2}{h^2} \right)$$
Kernel Density Gradient Estimate

▶ For differentiable $f$, define kernel density gradient estimate

$$\hat{\nabla}_x f_{n,h}(x) := \nabla_x \hat{f}_{n,h}(x) = \frac{1}{n} \sum_{i=1}^{n} \nabla_x \Phi_h(x - X_i)$$

(9)

$$= \frac{1}{n} \sum_{i=1}^{n} (x - X_i) \frac{2}{h^{d+2}} \phi' \left( \frac{\|x - X_i\|^2}{h^2} \right)$$

(10)

▶ (Gaussian kernel) $\phi(r) = (2\pi)^{-d/2} e^{-r/2}$

$$\hat{\nabla}_x f_{n,h}(x) = \frac{1}{n} \sum_{i=1}^{n} (X_i - x) \frac{1}{h^{d+2}} \frac{\exp \left( - \frac{\|x - X_i\|^2}{2h^2} \right)}{(2\pi)^{d/2}}$$

(11)
 Kernel Density Gradient Estimate

▶ For differentiable $f$, define kernel density gradient estimate

$$\hat{\nabla}_x f_{n,h}(x) := \nabla_x \hat{f}_{n,h}(x) = \frac{1}{n} \sum_{i=1}^{n} \nabla_x \Phi_h(x - X_i)$$

$$= \frac{1}{n} \sum_{i=1}^{n} (x - X_i) \frac{2}{h^{d+2}} \phi' \left( \frac{\|x - X_i\|^2}{h^2} \right)$$  \hfill (9)

▶ (Gaussian kernel) $\phi(r) = (2\pi)^{-d/2} e^{-r/2}$

$$\hat{\nabla}_x f_{n,h}(x) = \frac{1}{n} \sum_{i=1}^{n} (X_i - x) \frac{1}{h^{d+2}} \exp \left( - \frac{\|x - X_i\|^2}{2h^2} \right)$$

$$\frac{(2\pi)^{d/2}}{c_d(1-r)+ := c_d \max\{1-r, 0\}} $$

▶ (Epanechnikov kernel) $\phi(r) = c_d(1 - r)_+ := c_d \max\{1 - r, 0\}$

$$\hat{\nabla}_x f_{n,h}(x) = \frac{1}{n} \sum_{i=1}^{n} (X_i - x) \frac{2c_d}{h^{d+2}} 1\{\|x - X_i\| \leq h\}$$  \hfill (11)
Closer look of Epanechnikov kernel

- (Epanechnikov kernel) \( \phi(r) = c_d(1 - r)_+ := c_d \max\{1 - r, 0\} \)

\[
\hat{\nabla} x f_{n,h}(x) = \frac{1}{n} \sum_{i=1}^{n} (X_i - x) \frac{2c_d}{h^{d+2}} \mathbf{1\{\|x - X_i\| \leq h\}}
\]

(13)
Closer look of Epanechnikov kernel

(Epanechnikov kernel) \( \phi(r) = c_d (1 - r)_+ := c_d \max\{1 - r, 0\} \)

\[
\hat{\nabla}_x f_{n,h}(x) = \frac{1}{n} \sum_{i=1}^{n} (X_i - x) \frac{2c_d}{h^{d+2}} 1\{\|x - X_i\| \leq h\} \\
= \frac{2c_d}{nh^{d+2}} \sum_{X_i \in B(x,h)} (X_i - x)
\]
Close look of Epanechnikov kernel

- (Epanechnikov kernel) \( \phi(r) = c_d(1 - r)_+ := c_d \max\{1 - r, 0\} \)

\[
\hat{\nabla}_x f_{n,h}^\phi(x) = \frac{1}{n} \sum_{i=1}^{n} (X_i - x) \frac{2c_d}{h^{d+2}} 1\{|x - X_i| \leq h\} \\
= \frac{2c_d}{nh^{d+2}} \sum_{X_i \in B(x,h)} (X_i - x) \\
= \frac{\#\{i: X_i \in B(x, h)\}}{n \text{vol}(B(x, h))} \frac{d + 2}{h^2} (m_h(x) - x)
\]
Closer look of Epanechnikov kernel

(Epanechnikov kernel) \( \phi(r) = c_d(1 - r)_+ := c_d \max\{1 - r, 0\} \)

\[
\hat{\nabla}_x f_{n,h}(x) = \frac{1}{n} \sum_{i=1}^{n} (X_i - x) \frac{2c_d}{h^{d+2}} \mathbb{1}\{\|x - X_i\| \leq h\}
\]

\[
= \frac{2c_d}{nh^{d+2}} \sum_{X_i \in B(x,h)} (X_i - x)
\]

\[
= \frac{\#\{i: X_i \in B(x,h)\}}{n \text{vol}(B(x,h))} \frac{d + 2}{h^2} (m_h(x) - x)
\]

\[
= \frac{\hat{f}_\psi}{f_{n,h}(x)} \frac{d + 2}{h^2} (m_h(x) - x)
\]

KDE with flat kernel

mean-shift
Closer look of Epanechnikov kernel

- (Epanechnikov kernel) \( \phi(r) = c_d(1 - r)_+ := c_d \max\{1 - r, 0\} \)

\[
\hat{\nabla}_x f_{n,h}(x) = \frac{1}{n} \sum_{i=1}^{n} (X_i - x) \frac{2c_d}{h^{d+2}} \mathbb{1}\{\|x - X_i\| \leq h\}
\]

\[
= \frac{2c_d}{nh^{d+2}} \sum_{X_i \in B(x,h)} (X_i - x)
\]

\[
= \frac{\#\{i: X_i \in B(x,h)\}}{n \text{vol}(B(x,h))} \frac{d + 2}{h^2} (m_h(x) - x)
\]

\[
= \mathbb{f}_{n,h}(x) \underbrace{\frac{d + 2}{h^2} (m_h(x) - x)}_{\text{mean-shift}}
\]

KDE with flat kernel

where \( \psi(r) := \mathbb{1}\{r \leq 1\} \) is the flat profile function, and we recall

\[
m_h(x) := \frac{1}{\#\{i: X_i \in B(x,h)\}} \sum_{i: X_i \in B(x,h)} X_i
\]
Mean-Shift Normalized Gradient Estimates

\[ \widehat{\nabla}_x f_{n,h}^\phi(x) = \frac{d + 2}{h^2} \left( m_h(x) - x \right) \]

(18)

\[ \widehat{\nabla}_x \log f_{n,h}^\phi,\psi(x) := \frac{\widehat{\nabla}_x f_{n,h}^\phi(x)}{\widehat{f}_n^\psi(x)} = \frac{d + 2}{h^2} \left( m_h(x) - x \right) \]

(19)

Then, naturally, we define

KDE with flat kernel

mean-shift

\[ \widehat{\nabla}_x \log f_{n,h}^\phi,\psi(x) := \frac{\widehat{\nabla}_x f_{n,h}^\phi(x)}{\widehat{f}_n^\psi(x)} = \frac{d + 2}{h^2} \left( m_h(x) - x \right) \]
\[
\hat{\nabla}_x f_{\phi,h}(x) = \frac{\hat{f}_\psi(x)}{\hat{f}_{n,h}(x)} \frac{d + 2}{h^2} (m_h(x) - x)
\]

(18)

- Then, naturally, we define

\[
\hat{\nabla}_x \log \hat{f}_{\phi,\psi}(x) := \frac{\hat{\nabla}_x f_{\phi,h}(x)}{\hat{f}_{n,h}(x)} = \frac{d + 2}{h^2} (m_h(x) - x)
\]

(19)

\[
\nabla_x \log f(x) = \nabla_x f(x)/f(x)
\]

is called the *normalized gradient*
Mean-Shift Normalized Gradient Estimates

\[ \hat{\nabla}_x f_{n,h}(x) = \frac{\hat{f}_{n,h}(x)}{\int_{\mathbb{R}^d} f_{n,h}(x) \, dx} \quad \begin{cases} \frac{d + 2}{h^2} (m_h(x) - x) \end{cases} \]  

KDE with flat kernel \quad \text{mean-shift} \hspace{1cm} (18)

> Then, naturally, we define

\[ \hat{\nabla}_x \log f_{\phi,\psi}(x) := \frac{\hat{\nabla}_x f_{\phi,\psi}(x)}{\hat{f}_{\phi,\psi}(x)} = \frac{d + 2}{h^2} (m_h(x) - x) \]  

\hspace{1cm} (19)

> \nabla_x \log f(x) = \nabla_x f(x) / f(x) \text{ is called the } \text{normalized gradient}

> Hence, the mean-shift algorithm becomes (approximate) gradient ascent updates:

\[ y \leftarrow y + \eta \hat{\nabla}_y \log f_{\phi,\psi}(y) \]  

\hspace{1cm} (20)

for some \( \eta > 0 \)
Mean-Shift Normalized Gradient Estimates

\[
\hat{\nabla}_x f_{n,h}^\phi(x) = \frac{\hat{f}_n^\psi(x)}{h^2} \left( \frac{d + 2}{h^2} (m_h(x) - x) \right)
\]

KDE with flat kernel  
mean-shift

Then, naturally, we define

\[
\hat{\nabla}_x \log f_{n,h}^{\phi,\psi}(x) := \frac{\hat{\nabla}_x f_{n,h}^\phi(x)}{\hat{f}_n^\psi(x)} = \frac{d + 2}{h^2} (m_h(x) - x)
\]

\[
\hat{\nabla}_x \log f(x) = \nabla_x f(x)/f(x)
\]

is called the **normalized gradient**

Hence, the mean-shift algorithm becomes (approximate) gradient ascent updates:

\[
y \leftarrow y + \eta \hat{\nabla}_y \log f_{n,h}^{\phi,\psi}(y)
\]

for some \(\eta > 0\)

This will find root \(\nabla_x \log f(x) = 0\), or equivalently, \(\nabla_x f(x) = 0\) (i.e., finding modes)
Why update with $\nabla \log f(x)$?

1. We can escape local minima or tails faster with additional $f(x)^{-1}$
Why update with $\nabla \log f(x)$?

1. We can escape local minima or tails faster with additional $f(x)^{-1}$
   ▶ Or equivalently, the problematic plateau of $f(x)$ can be magnified by logarithm
Why update with $\nabla \log f(x)$?

1. We can escape local minima or tails faster with additional $f(x)^{-1}$
   - Or equivalently, the problematic plateau of $f(x)$ can be magnified by logarithm.

2. For Gaussian density, a proper $\eta$ gives one-step convergence to the mean.
Why update with $\nabla \log f(x)$?

1. We can escape local minima or tails faster with additional $f(x)^{-1}$
   - Or equivalently, the problematic plateau of $f(x)$ can be magnified by logarithm
2. For Gaussian density, a proper $\eta$ gives one-step convergence to the mean
3. Can show convergence in a mean-square sense (in terms of blurring process)
Why update with $\nabla \log f(x)$?

1. We can escape local minima or tails faster with additional $f(x)^{-1}$
   - Or equivalently, the problematic plateau of $f(x)$ can be magnified by logarithm
2. For Gaussian density, a proper $\eta$ gives one-step convergence to the mean
3. Can show convergence in a mean-square sense (in terms of blurring process)
   - Consider the transformation of random variables

$$Y = X + \eta \nabla_x \log f(X)$$  \hspace{1cm} (21)
Why update with $\nabla \log f(x)$?

1. We can escape local minima or tails faster with additional $f(x)^{-1}$
   ▶ Or equivalently, the problematic plateau of $f(x)$ can be magnified by logarithm

2. For Gaussian density, a proper $\eta$ gives one-step convergence to the mean

3. Can show convergence in a mean-square sense (in terms of blurring process)
   ▶ Consider the transformation of random variables
     \[
     Y = X + \eta \nabla_x \log f(X)
     \] (21)
   ▶ Choosing a proper $\eta$ will ensure that the covariance of the samples get smaller, i.e.,
     \[
     \mathbb{E}[(Y - \mathbb{E}[Y])^T(Y - \mathbb{E}[Y])] \leq \mathbb{E}[(X - \mathbb{E}[X])^T(X - \mathbb{E}[X])]
     \]
Why update with $\nabla \log f(x)$?

1. We can escape local minima or tails faster with additional $f(x)^{-1}$
   - Or equivalently, the problematic plateau of $f(x)$ can be magnified by logarithm
2. For Gaussian density, a proper $\eta$ gives one-step convergence to the mean
3. Can show convergence in a mean-square sense (in terms of blurring process)
   - Consider the transformation of random variables
     \[
     Y = X + \eta \nabla_x \log f(X)
     \] (21)
   - Choosing a proper $\eta$ will ensure that the covariance of the samples get smaller, i.e.,
     \[
     \mathbb{E}[(Y - \mathbb{E}[Y])^T(Y - \mathbb{E}[Y])] \leq \mathbb{E}[(X - \mathbb{E}[X])^T(X - \mathbb{E}[X])]
     \]
4. Can be directly estimated from mean-shift
Generalized Mean-Shift

So far, the sample mean has been defined with the flat kernel
\( \Psi(x) = \psi(\|x\|^2), \psi(r) = 1 \{ r \leq 1 \} \):

\[
m_h(x) := \frac{\sum_{s \in S} \Psi_h(s - x)s}{\sum_{s \in S} \Psi_h(s - x)}
\] (22)
Generalized Mean-Shift

- So far, the sample mean has been defined with the flat kernel
  \( \Psi(x) = \psi(||x||^2), \psi(r) = 1 \{ r \leq 1 \} \):

  \[
  m_h^\psi(x) := \frac{\sum_{s \in S} \Psi_h(s - x)s}{\sum_{s \in S} \Psi_h(s - x)} \quad (22)
  \]

- And we showed that

  \[
  m_h^\psi(x) - x \propto \frac{\hat{\nabla}_x f_{n,h}(x)}{f_{n,h}(x)} \quad (23)
  \]

  for \( \Phi(x) = \phi(||x||^2), \phi(r) = c_d(1 - r)_+ \) (the Epanechnikov kernel)
Generalized Mean-Shift

So far, the sample mean has been defined with the flat kernel 
\[ \Psi(x) = \psi(\|x\|^2), \psi(r) = 1\{r \leq 1\} \]:

\[
m^\psi_h(x) := \frac{\sum_{s \in S} \Psi_h(s - x)s}{\sum_{s \in S} \Psi_h(s - x)} \tag{22}
\]

And we showed that

\[
m^\psi_h(x) - x \propto \frac{\hat{\nabla}_x f_{n,h}(x)}{\hat{f}_\psi(x)} \tag{23}
\]

for \( \Phi(x) = \phi(\|x\|^2) \), \( \phi(r) = c_d(1 - r)_+ \) (the Epanechnikov kernel)

[Cheng, 1995] generalizes this as

\[
m^\psi_h(x) = \frac{\sum_{s \in S} \Psi_h(s - x)w(s)s}{\sum_{s \in S} \Psi_h(s - x)w(s)} \tag{24}
\]

where \( \Psi(x) = \psi(\|x\|^2) \) is an arbitrary kernel function and \( w: S \to (0, \infty) \) is a weight function on the sample \( S \)
Generalized Mean-Shift

So far, the sample mean has been defined with the flat kernel
\[ \Psi(x) = \psi(||x||^2), \quad \psi(r) = 1 \{ r \leq 1 \} \]

\[ m_h^\psi(x) := \frac{\sum_{s \in S} \Psi_h(s - x) s}{\sum_{s \in S} \Psi_h(s - x)} \tag{22} \]

And we showed that

\[ m_h^\psi(x) - x \propto \frac{\hat{\nabla}_x f_{n,h}(x)}{\hat{f}_n^\psi(x)} \tag{23} \]

for \( \Phi(x) = \phi(||x||^2), \quad \phi(r) = c_d (1 - r)_+ \) (the Epanechnikov kernel)

[Cheng, 1995] generalizes this as

\[ m_h^\psi(x) = \frac{\sum_{s \in S} \Psi_h(s - x) w(s) s}{\sum_{s \in S} \Psi_h(s - x) w(s)} \tag{24} \]

where \( \Psi(x) = \psi(||x||^2) \) is an arbitrary kernel function and \( w: S \to (0, \infty) \) is a weight function on the sample \( S \)

Q) When does \( m_h^\psi(x) - x \) become gradient mapping?
Kernel $\bar{\Phi}$ is said to be a *shadow* of kernel $\bar{\Psi}$, if

$$m_{\psi}^h(x) - x \propto \hat{\nabla}_{x,f_{n,h}}(x)$$  \hspace{1cm} (25)
Shadow Kernel

Kernel $\Phi$ is said to be a *shadow* of kernel $\Psi$, if

$$m_\Psi(x) - x \propto \hat{\nabla}_x f_{n,h}(x)$$  \hspace{1cm} (25)

We know that the shadow of the flat kernel $\Psi$ is the Epanechnikov kernel $\Phi$. 

Shadow Kernel

Kernel $\Phi$ is said to be a *shadow* of kernel $\Psi$, if

$$m_h^\Psi(x) - x \propto \hat{\nabla}_x f_{n,h}^\phi(x)$$  \hspace{1cm} (25)

We know that the shadow of the flat kernel $\Psi$ is the Epanechnikov kernel $\Phi$.

\begin{tcolorbox}
\textbf{Theorem 1.}

$\Phi$ is a shadow of $\Psi$ \textbf{if and only if} their profile functions, $\psi$ and $\phi$, satisfy that

$$\phi(r) = g(r) + c \int_r^\infty \psi(t) \, dt,$$  \hspace{1cm} (26)

where $c > 0$ is a constant and $g$ is a piecewise constant function.
\end{tcolorbox}
Shadow Kernel

Kernel $\Phi$ is said to be a *shadow* of kernel $\Psi$, if

$$m_h^\psi(x) - x \propto \hat{\nabla}_x f_n^\phi(x)$$  \hspace{1cm} (25)

We know that the shadow of the flat kernel $\Psi$ is the Epanechnikov kernel $\Phi$.

**Theorem 1.**

$\Phi$ is a shadow of $\Psi$ if and only if their profile functions, $\psi$ and $\phi$, satisfy that

$$\phi(r) = g(r) + c \int_r^\infty \psi(t) \, dt,$$  \hspace{1cm} (26)

where $c > 0$ is a constant and $g$ is a piecewise constant function.

In this case,

$$m_h^\psi(x) - x = \frac{1}{2c} \frac{\hat{\nabla}_x f_n^\phi(x)}{\hat{f}_n^\psi(x)}$$  \hspace{1cm} (27)
Speical Case: Gaussian Kernels

Theorem 2.

- The only kernels that are their own shadows are the Gaussian kernel and its truncated version

\[ m_h(x) = \frac{1}{h^2} \nabla_x \log f_n(x) \]

Will be mentioned in the next presentation as a benign example.
Theorem 2.

- The only kernels that are their own shadows are the Gaussian kernel and its truncated version.
- In this case, the mean shift is equal to

\[ m_h^\psi(x) - x = \frac{1}{2h^2} \nabla_x \log f_{n,h}(x) \]  

(28)
Special Case: Gaussian Kernels

Theorem 2.

- The only kernels that are their own shadows are the Gaussian kernel and its truncated version.
- In this case, the mean shift is equal to

\[ m_h^\psi(x) - x = \frac{1}{2h^2} \nabla_x \log f_{n,h}^\psi(x) \]  

(28)

- Will be mentioned in the next presentation as a benign example.
Concluding Remarks

- Mean-shift is an elegant, simple, intuitive clustering algorithm.
Concluding Remarks

- Mean-shift is an elegant, simple, intuitive clustering algorithm
- Connection to kernel density gradient estimate
Concluding Remarks

▶ Mean-shift is an elegant, simple, intuitive clustering algorithm
▶ Connection to kernel density gradient estimate
▶ Generalized mean-shift with shadow kernel
Concluding Remarks

- Mean-shift is an elegant, simple, intuitive clustering algorithm
- Connection to kernel density gradient estimate
- Generalized mean-shift with shadow kernel
- Convergence guarantee
Concluding Remarks

- Mean-shift is an elegant, simple, intuitive clustering algorithm
- Connection to kernel density gradient estimate
- Generalized mean-shift with shadow kernel
- Convergence guarantee
  - Convergence of ver. 2, the blurring process (See [Cheng, 1995])
Concluding Remarks

- Mean-shift is an elegant, simple, intuitive clustering algorithm
- Connection to kernel density gradient estimate
- Generalized mean-shift with shadow kernel
- Convergence guarantee
  - Convergence of ver. 2, the blurring process (See [Cheng, 1995])
  - Convergence of ver. 1 (to be continued)
References


Any Questions?