Lecture Pattern Analysis

Part 08: Mean Shift

Christian Riess
IT Security Infrastructures Lab, Friedrich-Alexander-Universität Erlangen-Nürnberg
May 27, 2022

Introduction

- Mean Shift is another widely known algorithm for clustering[1]
- It iteratively performs gradient ascent on the derivative of a kernel
- This iteration converges to a mode (local maximum) of the density **without** estimating the full density

[1] The associated paper by Comaniciu and Meer is available in studOn

Kernel Notation and Constraints

- The kernel has to be **radially symmetric**, i.e.,
  the kernel function may only depend on the Euclidean distance to a sample
- The kernel functions must accept squared differences $||\mathbf{x}_0 - \mathbf{x}||_2^2$ as input,

$$K(\mathbf{x}_0, \mathbf{x}) = c \cdot k(||\mathbf{x}_0 - \mathbf{x}||_2^2) \tag{1}$$

  where c is an arbitrary constant and $k(x)$ is the so-called kernel profile
- This definition admits in particular the
  - Gaussian kernel with

$$k_{\text{Gauss}}(x) = \exp(-\frac{1}{2}x) \tag{2}$$

  - Epanechnikov kernel[2] with

$$k_{\text{Ep}}(x) = \begin{cases} 1 - x & \text{for} |x| \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

- Kernel sizes: Assume that distances $||\mathbf{x}_0 - \mathbf{x}||_2^2$ are already size-normalized[3]

[2] The typical way to write the Epanechnikov kernel is shown in Hastie/Tibshirani/Friedman Eqn. (6.3) and Eqn. (6.4) for the 1-D case. Our notation differs because our input is already squared and pre-factors factors are absorbed in c in Eqn. 1

[3] See paper by Comaniciu/Meer Eqn. (1) and Eqn. (2)

Gradient Computation

- To find the maximum, calculate the gradient of a kernel density estimate

$$\nabla p(\mathbf{x}) = \nabla \frac{1}{N} \sum_{i=1}^{N} K_\lambda(\mathbf{x}_i, \mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} \nabla K_\lambda(\mathbf{x}_i, \mathbf{x}) \tag{4}$$

- Insert $K_\lambda(\mathbf{x}_0, \mathbf{x}) = ck(\|\mathbf{x}_0 - \mathbf{x}\|_2^2)$, and substitute $s = \|\mathbf{x}_0 - \mathbf{x}\|_2^2$
- Then, the first derivative of $k(s)$ w.r.t. $\mathbf{x}$ consists of

$$\frac{\partial k(s)}{\partial s} = k'(s) \qquad \frac{\partial s}{\partial \mathbf{x}} = \frac{\partial (\mathbf{x}_i - \mathbf{x})^T (\mathbf{x}_i - \mathbf{x})}{\partial \mathbf{x}} = -2(\mathbf{x}_i - \mathbf{x}) \tag{5}$$

which is used to find an extremum (maximum!) where the gradient is 0

$$\nabla p(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} c \cdot k'(\|\mathbf{x}_i - \mathbf{x}\|_2^2)(-2(\mathbf{x}_i - \mathbf{x})) \stackrel{!}{=} 0 \tag{6}$$

From the Gradient to the Mean Shift Vector

- The gradient equation directly provides one gradient ascend step:

$$\nabla p(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} c \cdot k'(\|\mathbf{x}_i - \mathbf{x}\|_2^2)(-2(\mathbf{x}_i - \mathbf{x})) = 0 \qquad (7)$$

$$\sum_{i=1}^{N} k'(\|\mathbf{x}_i - \mathbf{x}\|_2^2)(-2(\mathbf{x}_i - \mathbf{x})) = 0 \qquad (8)$$

$$\sum_{i=1}^{N} k'(\|\mathbf{x}_i - \mathbf{x}\|_2^2) \cdot \mathbf{x}_i - \sum_{i=1}^{N} k'(\|\mathbf{x}_i - \mathbf{x}\|_2^2) \cdot \mathbf{x} = 0 \qquad (9)$$

$$\frac{\sum_{i=1}^{N} k'(\|\mathbf{x}_i - \mathbf{x}\|_2^2) \cdot \mathbf{x}_i}{\sum_{i=1}^{N} k'(\|\mathbf{x}_i - \mathbf{x}\|_2^2)} - \mathbf{x} = 0 \qquad (10)$$

- The last row is the normalized gradient, also called the **mean shift vector**

Mean Shift Algorithm

1. Calculate the mean shift vector $m^{(t)}(\mathbf{x})$ for iteration t:

$$m^{(t)}(\mathbf{x}) = \frac{\sum_{i=1}^{N} k'\left(\|\mathbf{x}_i - \mathbf{x}^{(t)}\|_2^2\right) \cdot \mathbf{x}_i}{\sum_{i=1}^{N} k'\left(\|\mathbf{x}_i - \mathbf{x}^{(t)}\|_2^2\right)} - \mathbf{x}^{(t)} \tag{11}$$

2. Update position $\mathbf{x}^{(t)}$:

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} + m^{(t)}(\mathbf{x}) = \frac{\sum_{i=1}^{N} k'\left(\|\mathbf{x}_i - \mathbf{x}^{(t)}\|_2^2\right) \cdot \mathbf{x}_i}{\sum_{i=1}^{N} k'\left(\|\mathbf{x}_i - \mathbf{x}^{(t)}\|_2^2\right)} \tag{12}$$

   (note that $\mathbf{x}^{(t)}$ cancels, since it also occurs in $m^{(t)}(\mathbf{x})$)

3. Goto 1) until convergence (i.e., at a mode where gradient is 0)

Mean Shift for Clustering

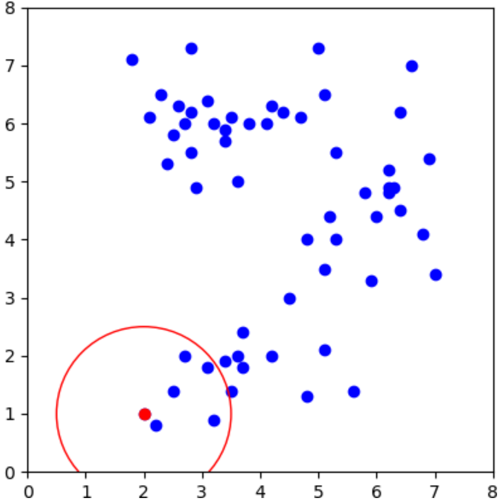
- Approach for using mean shift for clustering:
  - Run mean shift for each sample
  - Group samples that converge to nearby locations into the same cluster
- Required parameters for mean shift clustering:
  - The kernel parameters (only window size for Epanechnikov and Gauss)
  - Cluster linking parameters for postprocessing (e.g., a distance threshold)
- Larger kernels lead to less clusters (less local maxima)
- Smaller kernels lead to more clusters (more local maxima)
- The shape of mean shift clusters is potentially less regular than the shape of GMM or k-means clusters

Remarks

- With the Epanechnikov kernel, the update is just the mean of the samples within a D-dimensional sphere, hence the name “mean shift”
- General caveat: the Euclidean distance is sensitive to scaling differences $\rightarrow$ normalize the dimensions of the samples (this also applies to k-means)

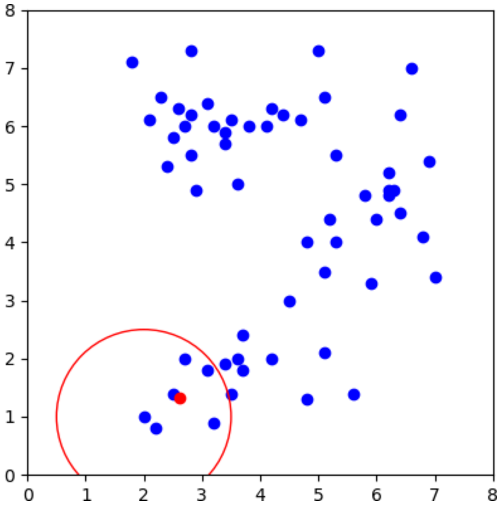
Example Run for Kernel Size 1.5

- Two steps per iteration are shown: calculation of mean shift vector and update of the sample position
- Red dot: current location on the sample path
- Red circle: kernel support
- Gray: path of a sample
- Red cross: Mode at the end of a path of a sample



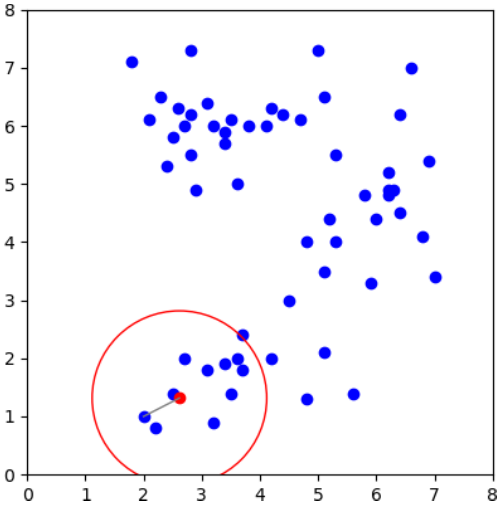
Example Run for Kernel Size 1.5

- Two steps per iteration are shown: calculation of mean shift vector and update of the sample position
- Red dot: current location on the sample path
- Red circle: kernel support
- Gray: path of a sample
- Red cross: Mode at the end of a path of a sample



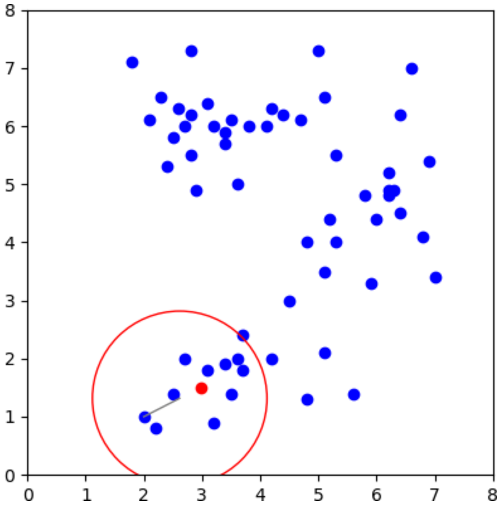
Example Run for Kernel Size 1.5

- Two steps per iteration are shown: calculation of mean shift vector and update of the sample position
- Red dot: current location on the sample path
- Red circle: kernel support
- Gray: path of a sample
- Red cross: Mode at the end of a path of a sample



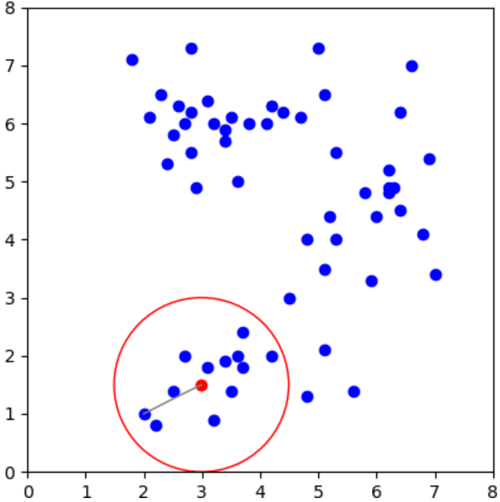
Example Run for Kernel Size 1.5

- Two steps per iteration are shown: calculation of mean shift vector and update of the sample position
- Red dot: current location on the sample path
- Red circle: kernel support
- Gray: path of a sample
- Red cross: Mode at the end of a path of a sample



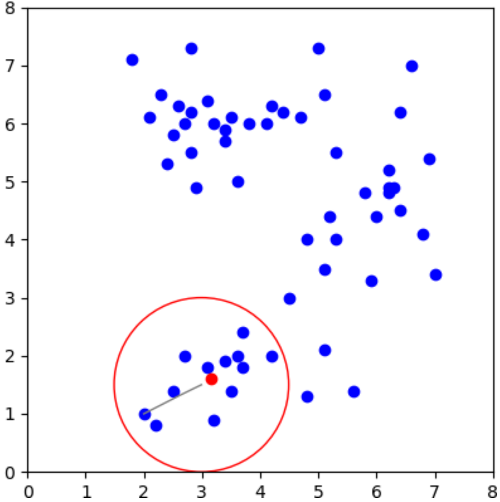
Example Run for Kernel Size 1.5

- Two steps per iteration are shown: calculation of mean shift vector and update of the sample position
- Red dot: current location on the sample path
- Red circle: kernel support
- Gray: path of a sample
- Red cross: Mode at the end of a path of a sample



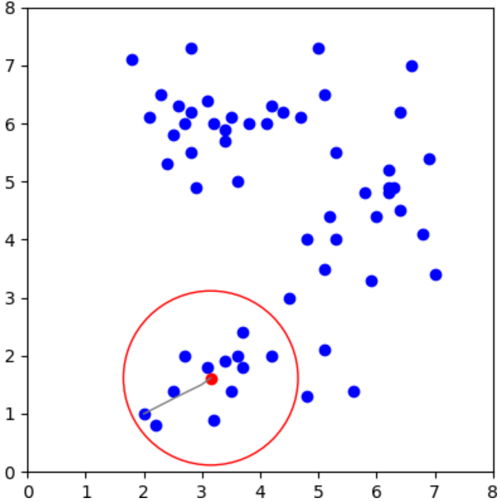
Example Run for Kernel Size 1.5

- Two steps per iteration are shown: calculation of mean shift vector and update of the sample position
- Red dot: current location on the sample path
- Red circle: kernel support
- Gray: path of a sample
- Red cross: Mode at the end of a path of a sample



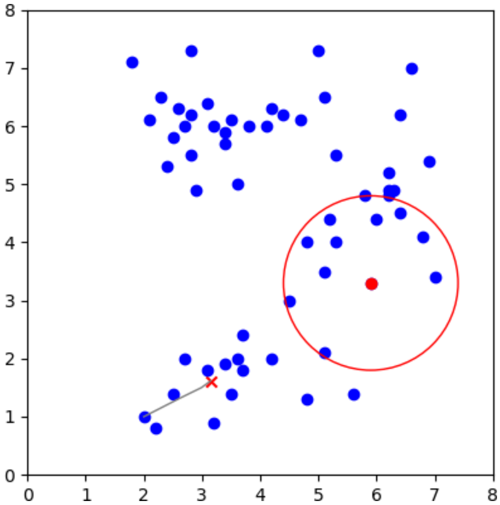
Example Run for Kernel Size 1.5

- Two steps per iteration are shown: calculation of mean shift vector and update of the sample position
- Red dot: current location on the sample path
- Red circle: kernel support
- Gray: path of a sample
- Red cross: Mode at the end of a path of a sample



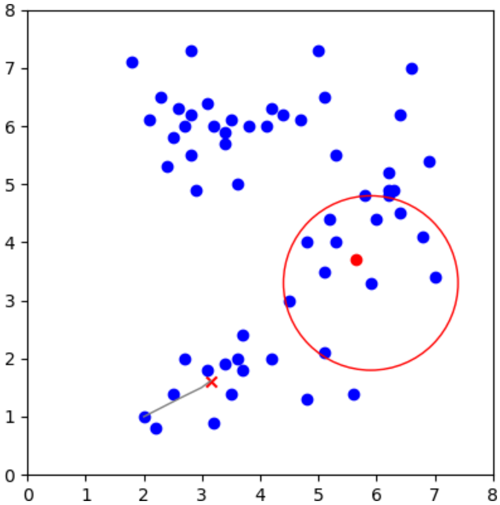
Example Run for Kernel Size 1.5

- Two steps per iteration are shown: calculation of mean shift vector and update of the sample position
- Red dot: current location on the sample path
- Red circle: kernel support
- Gray: path of a sample
- Red cross: Mode at the end of a path of a sample



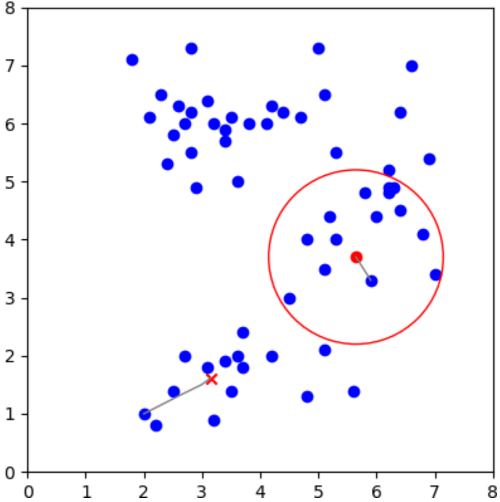
Example Run for Kernel Size 1.5

- Two steps per iteration are shown: calculation of mean shift vector and update of the sample position
- Red dot: current location on the sample path
- Red circle: kernel support
- Gray: path of a sample
- Red cross: Mode at the end of a path of a sample



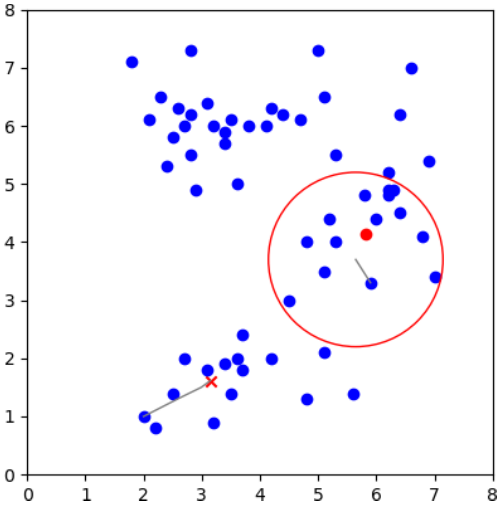
Example Run for Kernel Size 1.5

- Two steps per iteration are shown: calculation of mean shift vector and update of the sample position
- Red dot: current location on the sample path
- Red circle: kernel support
- Gray: path of a sample
- Red cross: Mode at the end of a path of a sample



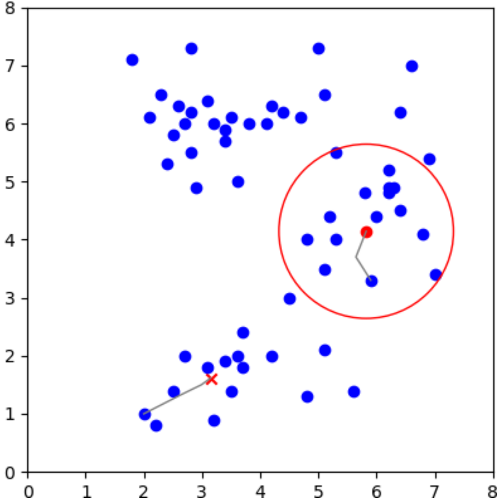
Example Run for Kernel Size 1.5

- Two steps per iteration are shown: calculation of mean shift vector and update of the sample position
- Red dot: current location on the sample path
- Red circle: kernel support
- Gray: path of a sample
- Red cross: Mode at the end of a path of a sample



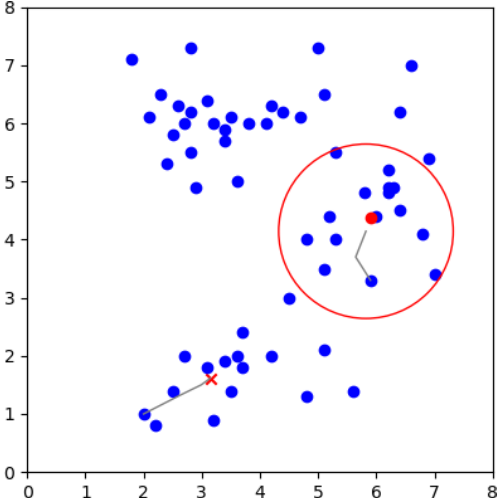
Example Run for Kernel Size 1.5

- Two steps per iteration are shown: calculation of mean shift vector and update of the sample position
- Red dot: current location on the sample path
- Red circle: kernel support
- Gray: path of a sample
- Red cross: Mode at the end of a path of a sample



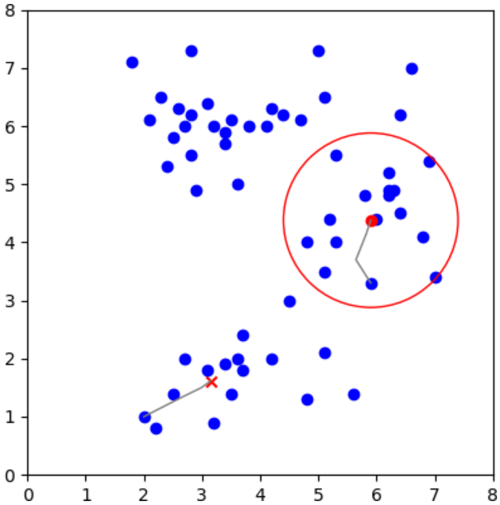
Example Run for Kernel Size 1.5

- Two steps per iteration are shown: calculation of mean shift vector and update of the sample position
- Red dot: current location on the sample path
- Red circle: kernel support
- Gray: path of a sample
- Red cross: Mode at the end of a path of a sample



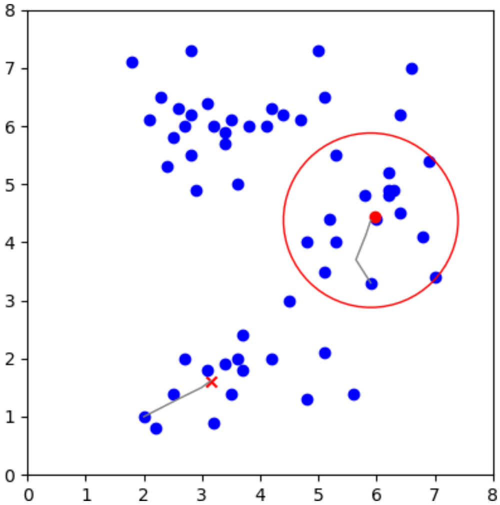
Example Run for Kernel Size 1.5

- Two steps per iteration are shown: calculation of mean shift vector and update of the sample position
- Red dot: current location on the sample path
- Red circle: kernel support
- Gray: path of a sample
- Red cross: Mode at the end of a path of a sample



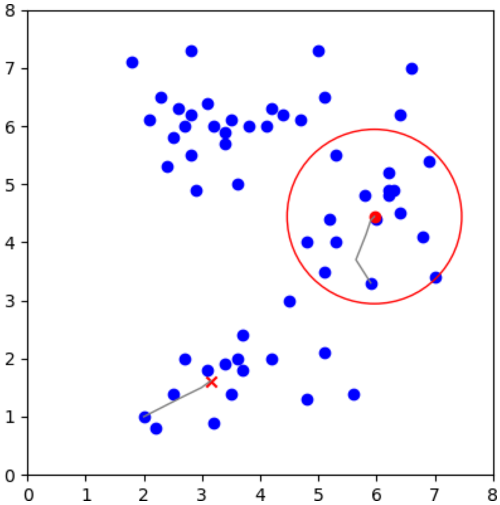
Example Run for Kernel Size 1.5

- Two steps per iteration are shown: calculation of mean shift vector and update of the sample position
- Red dot: current location on the sample path
- Red circle: kernel support
- Gray: path of a sample
- Red cross: Mode at the end of a path of a sample



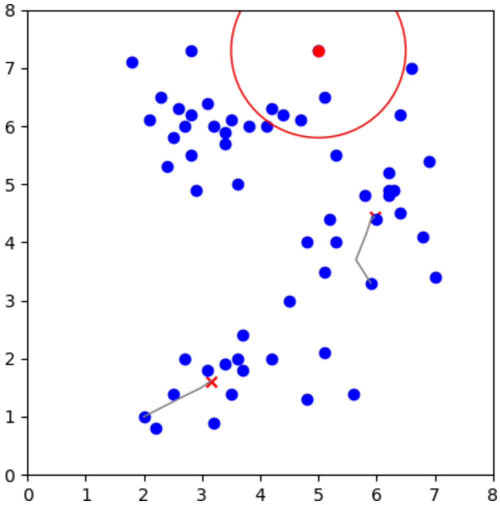
Example Run for Kernel Size 1.5

- Two steps per iteration are shown: calculation of mean shift vector and update of the sample position
- Red dot: current location on the sample path
- Red circle: kernel support
- Gray: path of a sample
- Red cross: Mode at the end of a path of a sample



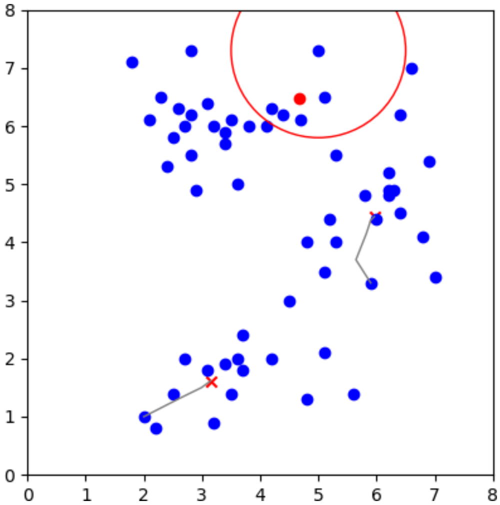
Example Run for Kernel Size 1.5

- Two steps per iteration are shown: calculation of mean shift vector and update of the sample position
- Red dot: current location on the sample path
- Red circle: kernel support
- Gray: path of a sample
- Red cross: Mode at the end of a path of a sample



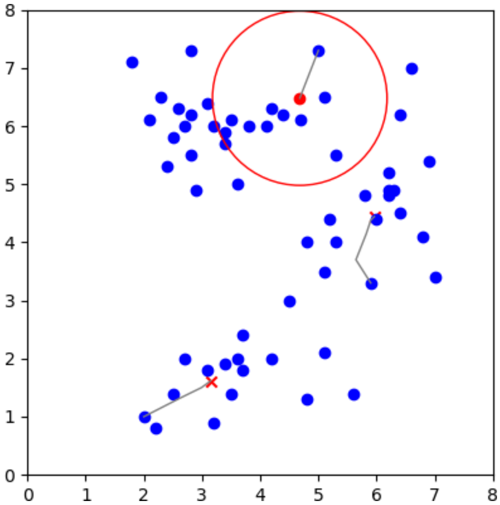
Example Run for Kernel Size 1.5

- Two steps per iteration are shown: calculation of mean shift vector and update of the sample position
- Red dot: current location on the sample path
- Red circle: kernel support
- Gray: path of a sample
- Red cross: Mode at the end of a path of a sample



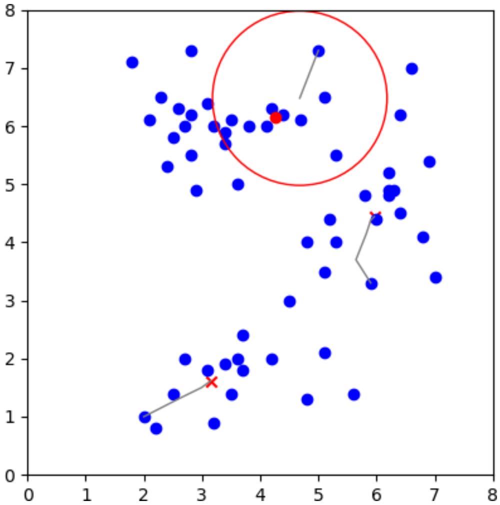
Example Run for Kernel Size 1.5

- Two steps per iteration are shown: calculation of mean shift vector and update of the sample position
- Red dot: current location on the sample path
- Red circle: kernel support
- Gray: path of a sample
- Red cross: Mode at the end of a path of a sample



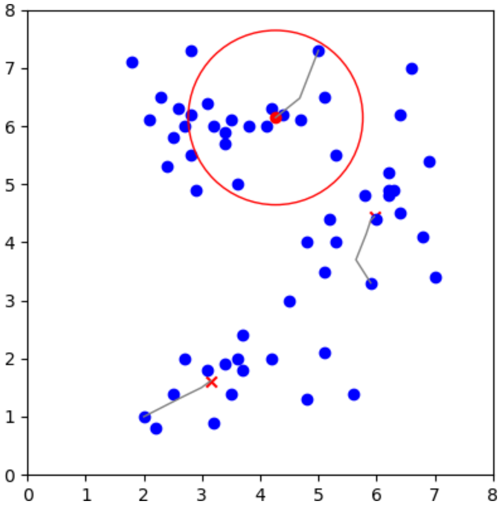
Example Run for Kernel Size 1.5

- Two steps per iteration are shown: calculation of mean shift vector and update of the sample position
- Red dot: current location on the sample path
- Red circle: kernel support
- Gray: path of a sample
- Red cross: Mode at the end of a path of a sample



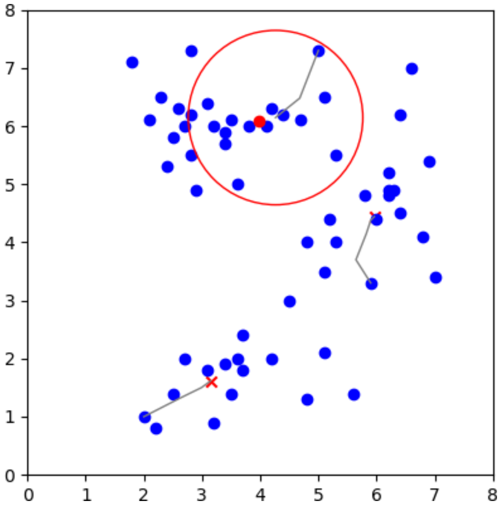
Example Run for Kernel Size 1.5

- Two steps per iteration are shown: calculation of mean shift vector and update of the sample position
- Red dot: current location on the sample path
- Red circle: kernel support
- Gray: path of a sample
- Red cross: Mode at the end of a path of a sample



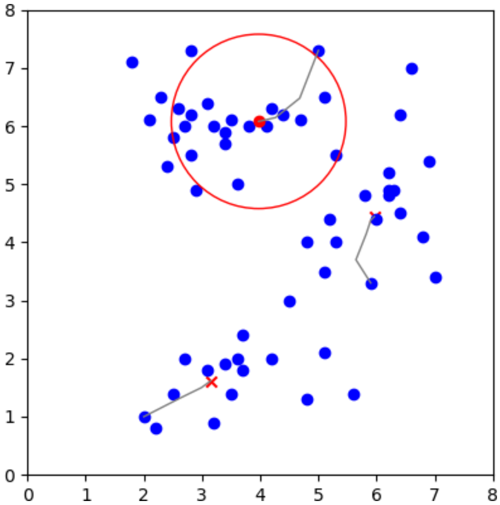
Example Run for Kernel Size 1.5

- Two steps per iteration are shown: calculation of mean shift vector and update of the sample position
- Red dot: current location on the sample path
- Red circle: kernel support
- Gray: path of a sample
- Red cross: Mode at the end of a path of a sample



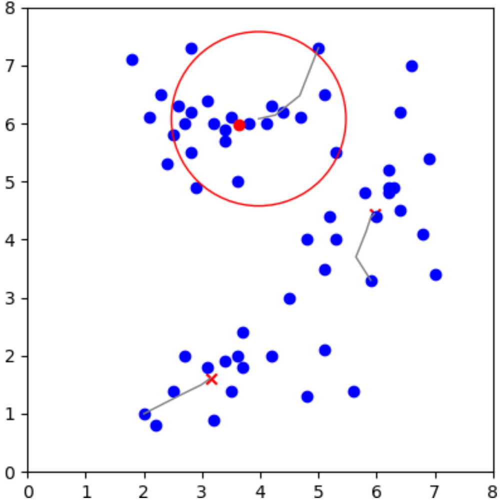
Example Run for Kernel Size 1.5

- Two steps per iteration are shown: calculation of mean shift vector and update of the sample position
- Red dot: current location on the sample path
- Red circle: kernel support
- Gray: path of a sample
- Red cross: Mode at the end of a path of a sample



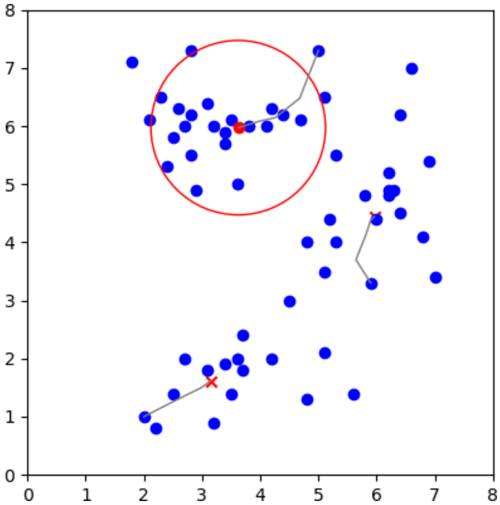
Example Run for Kernel Size 1.5

- Two steps per iteration are shown: calculation of mean shift vector and update of the sample position
- Red dot: current location on the sample path
- Red circle: kernel support
- Gray: path of a sample
- Red cross: Mode at the end of a path of a sample



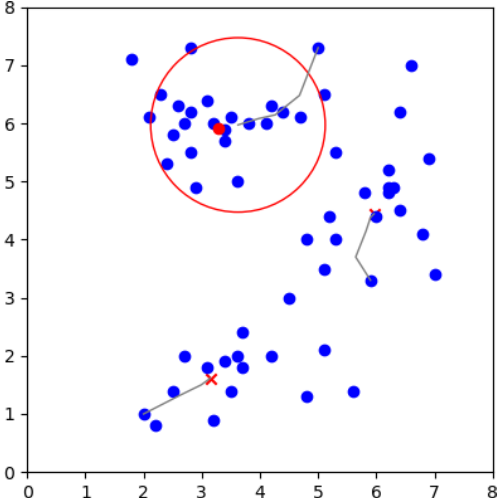
Example Run for Kernel Size 1.5

- Two steps per iteration are shown: calculation of mean shift vector and update of the sample position
- Red dot: current location on the sample path
- Red circle: kernel support
- Gray: path of a sample
- Red cross: Mode at the end of a path of a sample



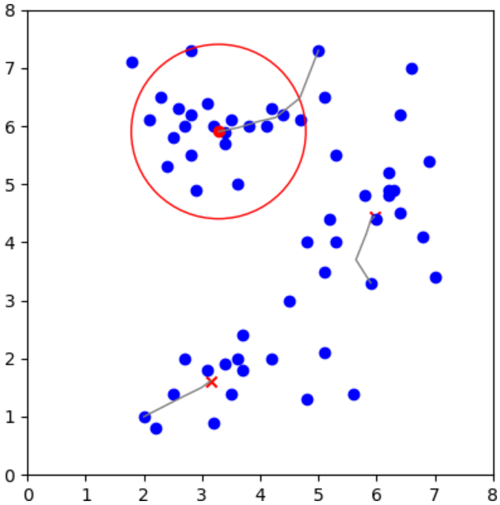
Example Run for Kernel Size 1.5

- Two steps per iteration are shown: calculation of mean shift vector and update of the sample position
- Red dot: current location on the sample path
- Red circle: kernel support
- Gray: path of a sample
- Red cross: Mode at the end of a path of a sample



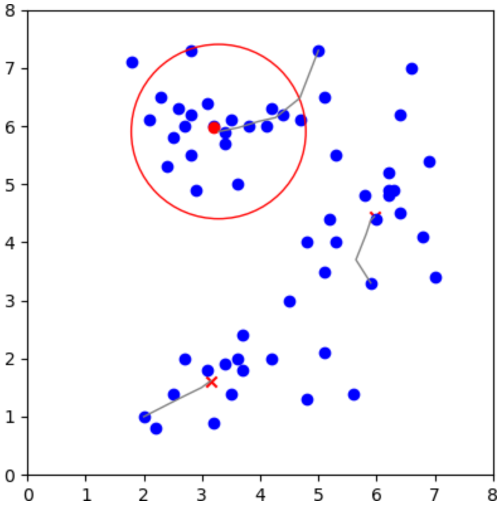
Example Run for Kernel Size 1.5

- Two steps per iteration are shown: calculation of mean shift vector and update of the sample position
- Red dot: current location on the sample path
- Red circle: kernel support
- Gray: path of a sample
- Red cross: Mode at the end of a path of a sample



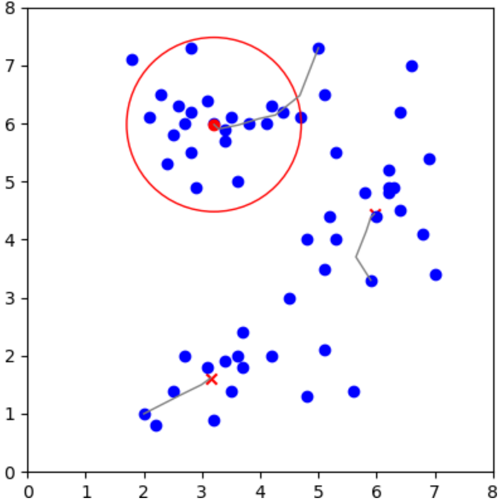
Example Run for Kernel Size 1.5

- Two steps per iteration are shown: calculation of mean shift vector and update of the sample position
- Red dot: current location on the sample path
- Red circle: kernel support
- Gray: path of a sample
- Red cross: Mode at the end of a path of a sample



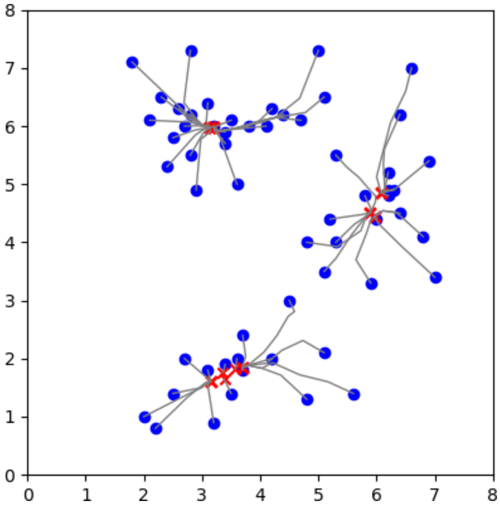
Example Run for Kernel Size 1.5

- Two steps per iteration are shown: calculation of mean shift vector and update of the sample position
- Red dot: current location on the sample path
- Red circle: kernel support
- Gray: path of a sample
- Red cross: Mode at the end of a path of a sample



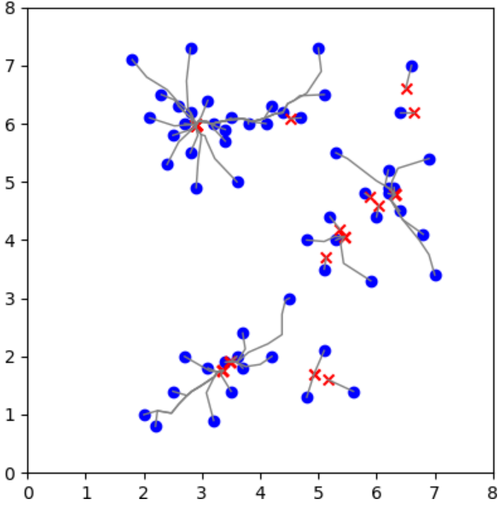
Example Run for Kernel Size 1.5

- Two steps per iteration are shown: calculation of mean shift vector and update of the sample position
- Red dot: current location on the sample path
- Red circle: kernel support
- Gray: path of a sample
- Red cross: Mode at the end of a path of a sample



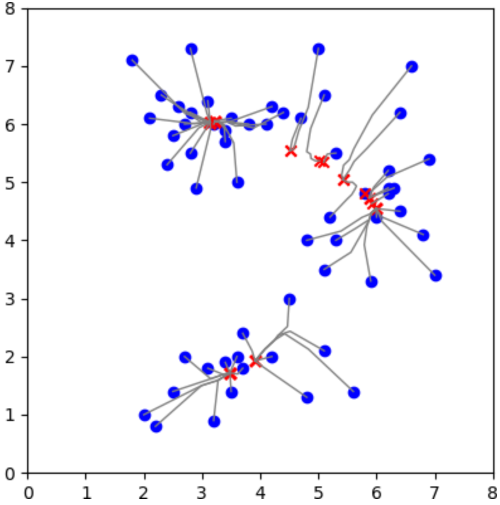
Example Run for Kernel Size 1

- Smaller kernel (1.5 → 1): more clusters
- Gray: path of a sample
- Red cross: Mode at the end of a path of a sample



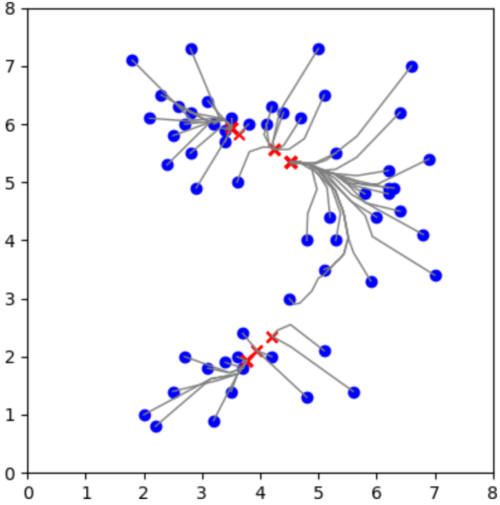
Example Run for Kernel Size 2

- Larger kernel (1.5 $\rightarrow$ 2): clusters start to merge
- Note the characteristic mode ridge in the upper part
- Gray: path of a sample
- Red cross: Mode at the end of a path of a sample



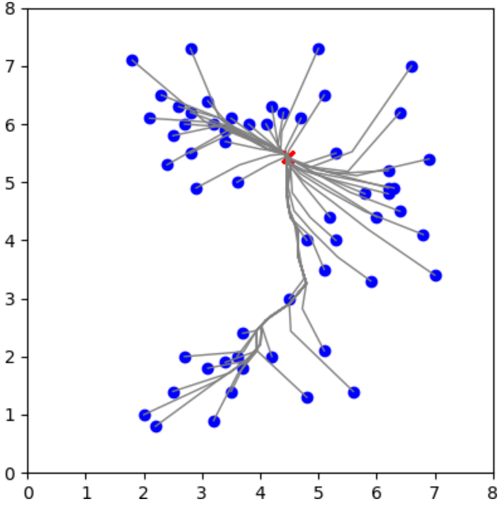
Example Run for Kernel Size 2.5

- Larger kernel (1.5 $\rightarrow$ 2.5): upper 2 clusters almost merged
- Gray: path of a sample
- Red cross: Mode at the end of a path of a sample



Example Run for Kernel Size 3

- Larger kernel ($1.5 \rightarrow 3$): only a single cluster
- Gray: path of a sample
- Red cross: Mode at the end of a path of a sample



Lecture Pattern Analysis

Part 09: Model Selection for K-Means

Christian Riess
IT Security Infrastructures Lab, Friedrich-Alexander-Universität Erlangen-Nürnberg
May 29, 2022

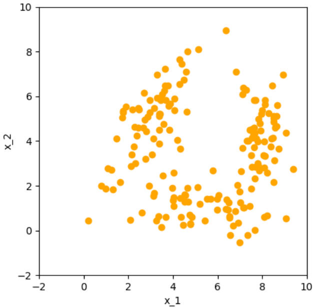
Introduction

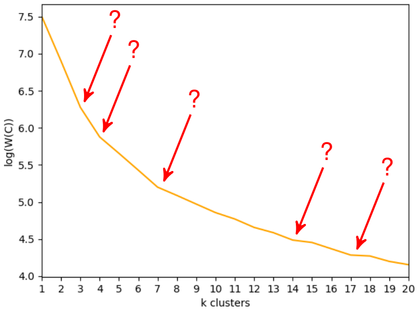
- Clustering is unsupervised, and does not provide an objective function for model selection
- So, specifically for k-means: what k shall we choose?
- Even if the application demands, e.g., the “3 most important clusters”, $k = 3$ could be a poor choice if the intrinsic number of clusters is larger
- In this lecture, we investigate the **Gap-Statistics** as a statistical way to determine k[1]
- The idea is to
  - examine the k-means optimization criterion, the **Within-Cluster Distance** $W(C)$, for different k,
  - and to select the smallest k for which $W(C)$ is substantially better than the $W(C)$ of $k + 1$ clusters

[1] The gap statistics is covered in the book by Hastie/Tibshirani/Friedman Sec. 14.3.11

Examining the Within-Cluster-Distance $W(C)$

- Investigate the progression of $W(C)$ for different k
- For increasing k, $W(C)$ has to decrease (exceptions are bad local minima):





- Hence, the optimum k can not be found by searching for the minimal $W(C)$
- An alternative is the “elbow method”, to search for a elbow on the curve
- However, which elbow is significant? At $k = \{3, 4, 7, 14, 17\}$?

Gap Statistics

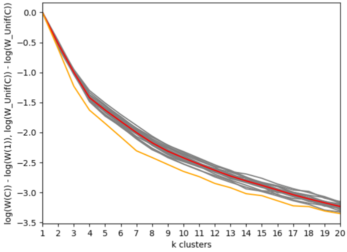
- Tibshirani *et al.* propose to relate $W(C)$ of our samples to the $W(C)$ of an artificially created reference
- This reference are clusterings of uniform sample distributions
- More specifically:
  1. Draw B sets of uniformly distributed samples (Tibshirani uses $B = 20$)
  2. On those distributions, calculate for different k the mean of the log of $W(C)$, denote the result $\log(W_{\text{unif}}(C))$
  3. For k clusters, calculate the gap $G(k)$ as the difference between the reference $\log(W_{\text{unif}}(C))$ and our log-within cluster distances $\log(W(C))$
  4. Select the optimum k as

$$k^* = \operatorname*{argmin}_k \{k | G(k) \geq G(k+1) - s'_{k+1}\} \tag{1}$$

where $s'_{k+1} = s_k \cdot \sqrt{1 + 1/B}$ is an unbiased estimate of the standard deviation s_k of $\log(W_{\text{unif}}(C))$

Example: Within-Cluster Distances on the Uniform Distribution

- Offset-corrected $\log(W(C))$ (orange) and $\log(W_{\text{unif}}(C))$ (red), and the $B = 20$ individual reference curves (gray):



- Gaps and standard deviations for curve differences. $k^* = 3$ is selected: