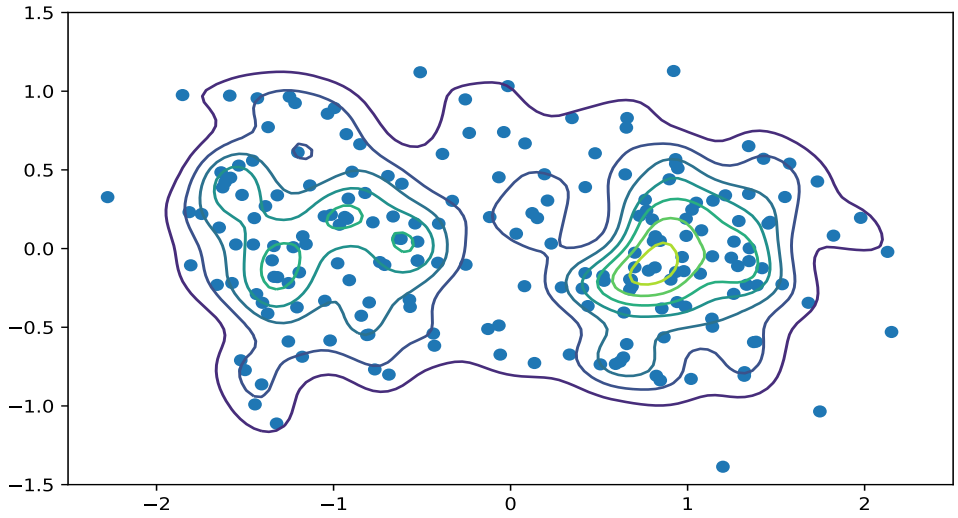
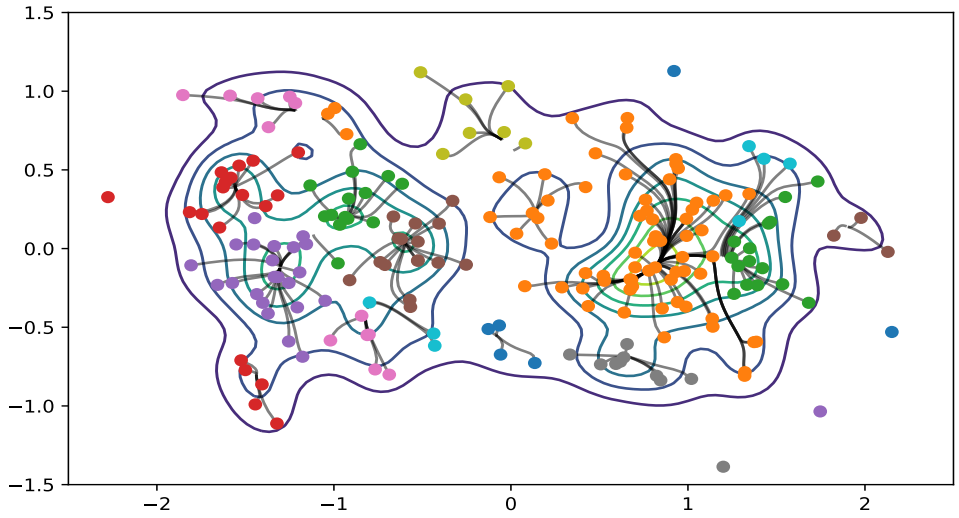
Karlsruher Institut für Technologie

Kernel Density Estimates and Mean Shift Clustering

Jonas Spinner | February 4, 2019

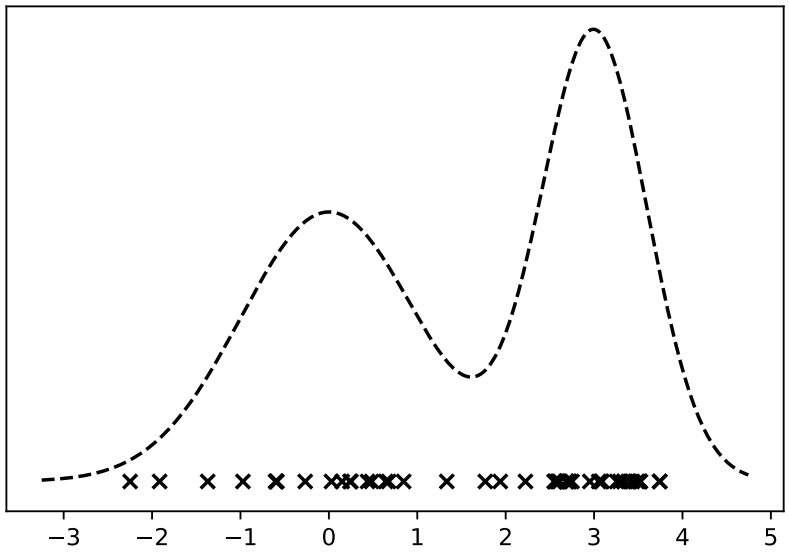
ANALYTICS AND STATISTICS AT THE INSTITUTE OF OPERATIONS RESEARCH





Outline

Density estimation



- **Task**: Given samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d$, estimate the underlying density.

The kernel density estimate

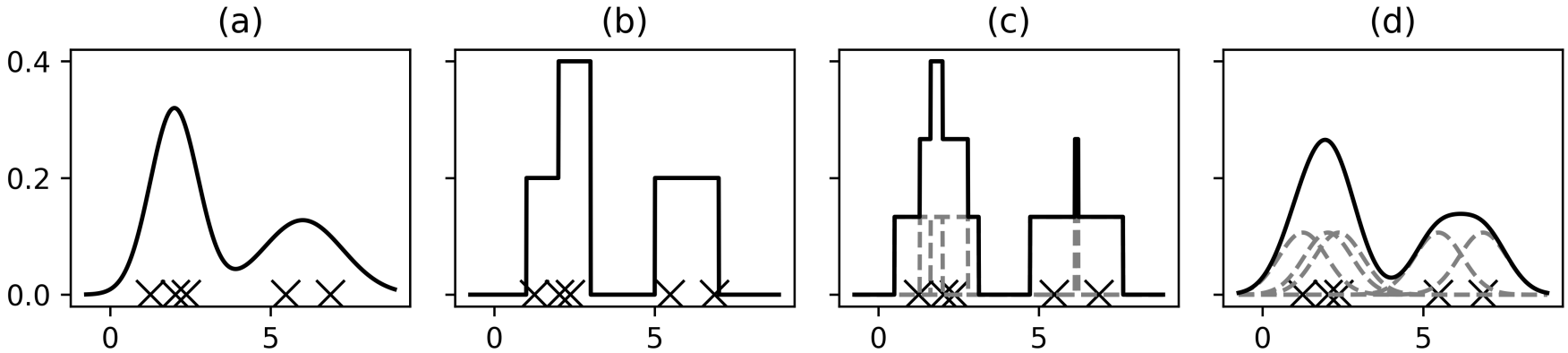
The **kernel density estimate** is defined as

$$\hat{f}(\boldsymbol{x}) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{\boldsymbol{x} - \boldsymbol{x}_i}{h}\right)$$

- K is a *kernel function*.
- h is a *bandwidth parameter*.
- When $\int_{\mathbb{R}^d} K(\boldsymbol{x}) d\boldsymbol{x} = 1$ and $K(\boldsymbol{x}) \geq 0$ for all $\boldsymbol{x}$ then $\hat{f}(\boldsymbol{x})$ is a valid probability density function.

The kernel density estimate

$$\hat{f}(\boldsymbol{x}) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{\boldsymbol{x} - \boldsymbol{x}_i}{h}\right)$$



(a) Underlying density.

(b) Histogram.

(c) KDE with uniform kernel.

(d) KDE with gaussian kernel.

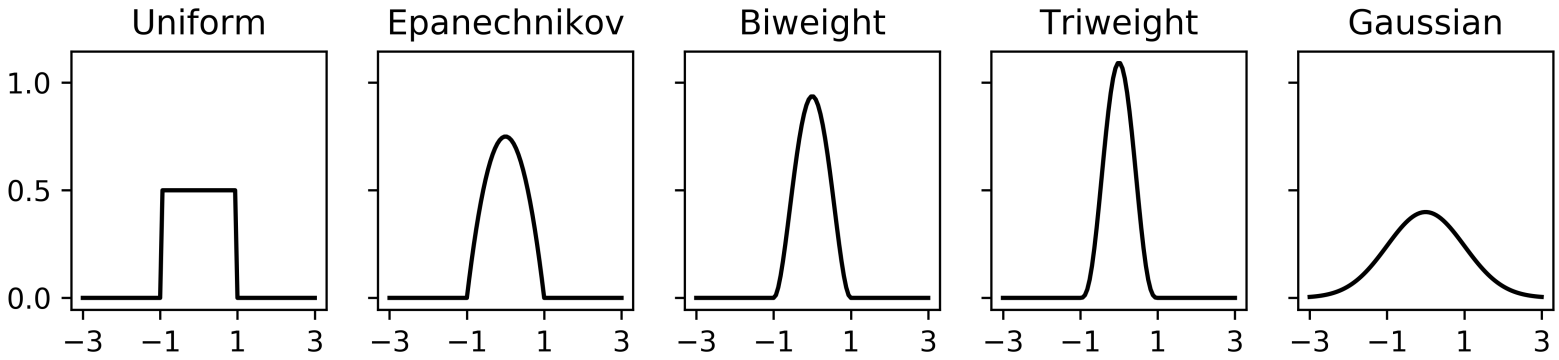
Popular kernel functions

Radially symmetric kernel functions are kernel functions which can be represented as

$$K(\boldsymbol{x}) = c_{k,d}\ k\left(\|\boldsymbol{x}\|^2\right)$$

- $k(u) : [0, \infty) \to [0, \infty)$ is called the **profile** of K.
- For example the gaussian kernel has the profile $k(u) = \exp(-\frac{1}{2}u)$.
- Nearly all popular kernel belong to this class of kernels.

Popular kernel functions



Name	Profile support	$k(u)$	$-k'(u)$	$K(\boldsymbol{x})$
Uniform	$u \in [0,1]$	1	0	$\text{vol}\,(S_d)^{-1}$
Epanechnikov	$u \in [0,1]$	$1-u$	1	$\frac{1}{2}\text{vol}(S_d)^{-1}(d+2)\left(1-\|\boldsymbol{x}\|^2\right)$
Biweight	$u \in [0,1]$	$(1-u)^2$	$2(1-u)$	$\propto \left(1-\|\boldsymbol{x}\|^2\right)^2$
Triweight	$u \in [0,1]$	$(1-u)^3$	$3(1-u)^2$	$\propto \left(1-\|\boldsymbol{x}\|^2\right)^3$
Gaussian	$u \in [0,\infty)$	$\exp\left(-\frac{1}{2}u\right)$	$\frac{1}{2}\exp\left(-\frac{1}{2}u\right)$	$(2\pi)^{-d/2}\exp\left(-\frac{1}{2}\|\boldsymbol{x}\|^2\right)$

Bandwidth

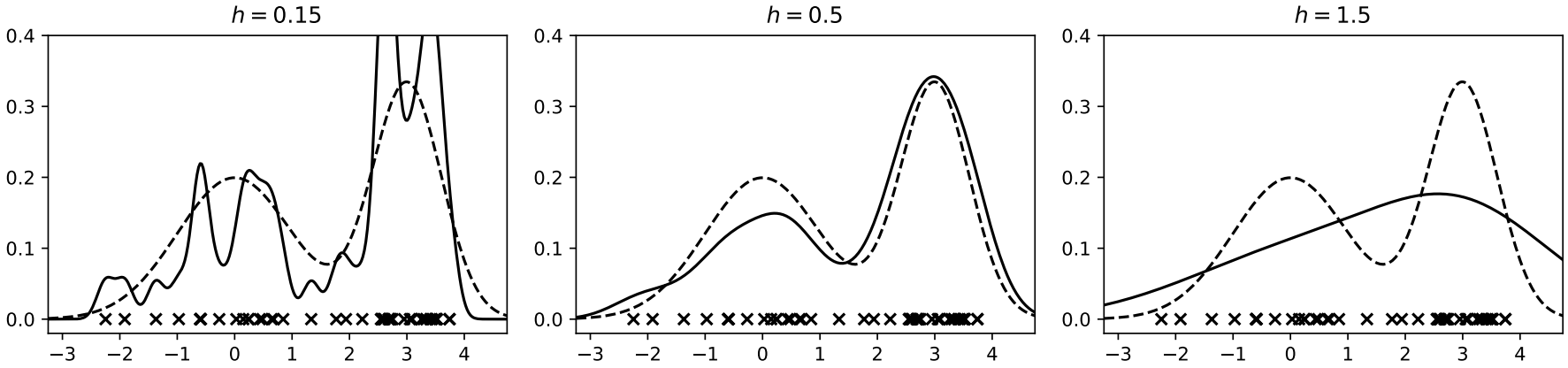


Abbildung: Gaussian kernel, $n = 50$.

- The choice of bandwidth is a bias-variance tradeoff for the estimate $\hat{f}(\boldsymbol{x})$.
- A small bandwidth results in high variance, a large bandwidth introduces a bias.

Bandwidth

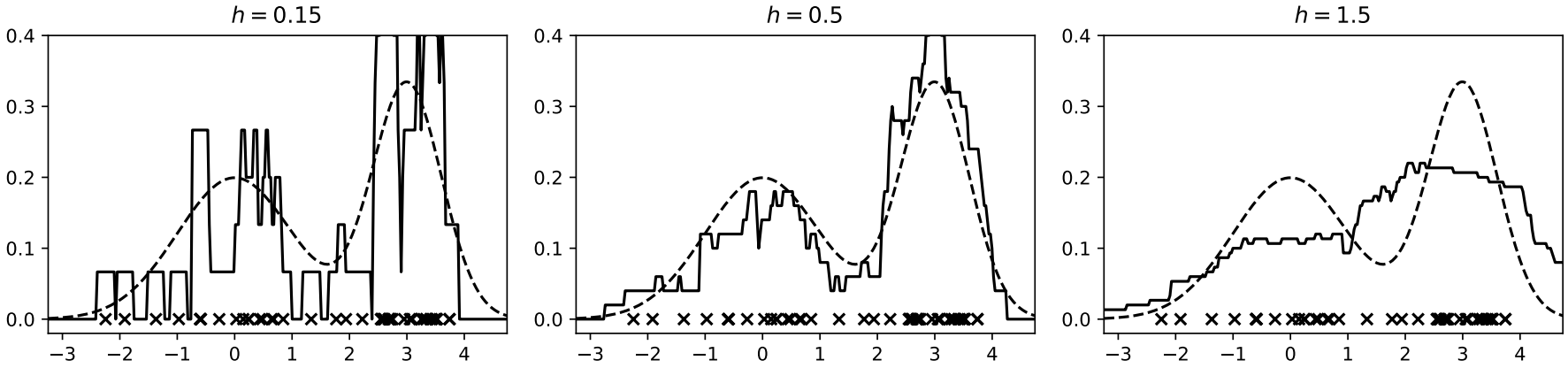


Abbildung: Uniform kernel, $n = 50$.

- The choice of bandwidth is a bias-variance tradeoff for the estimate $\hat{f}(\boldsymbol{x})$.
- A small bandwidth results in high variance, a large bandwidth introduces a bias.

Bandwidth

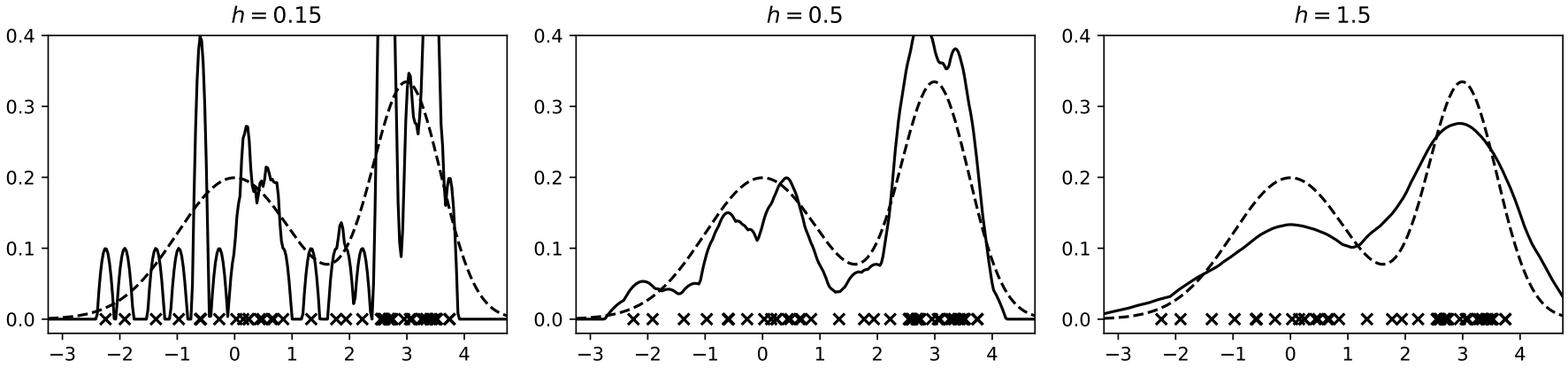
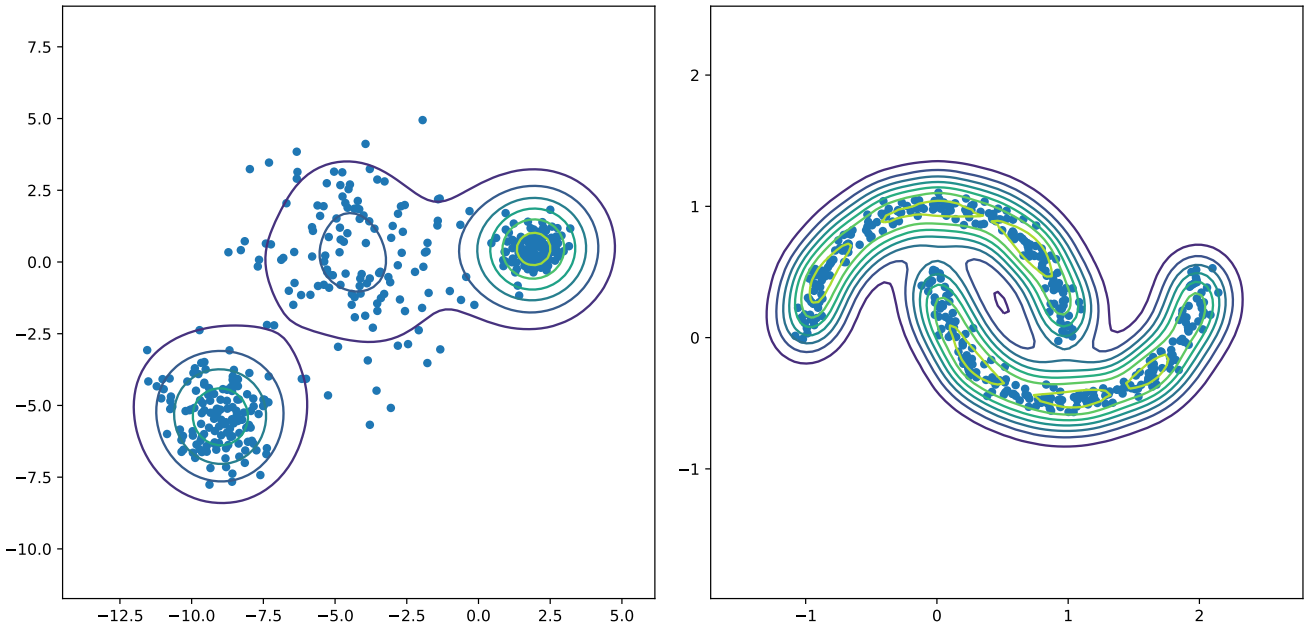


Abbildung: Epanechnikov kernel, $n = 50$.

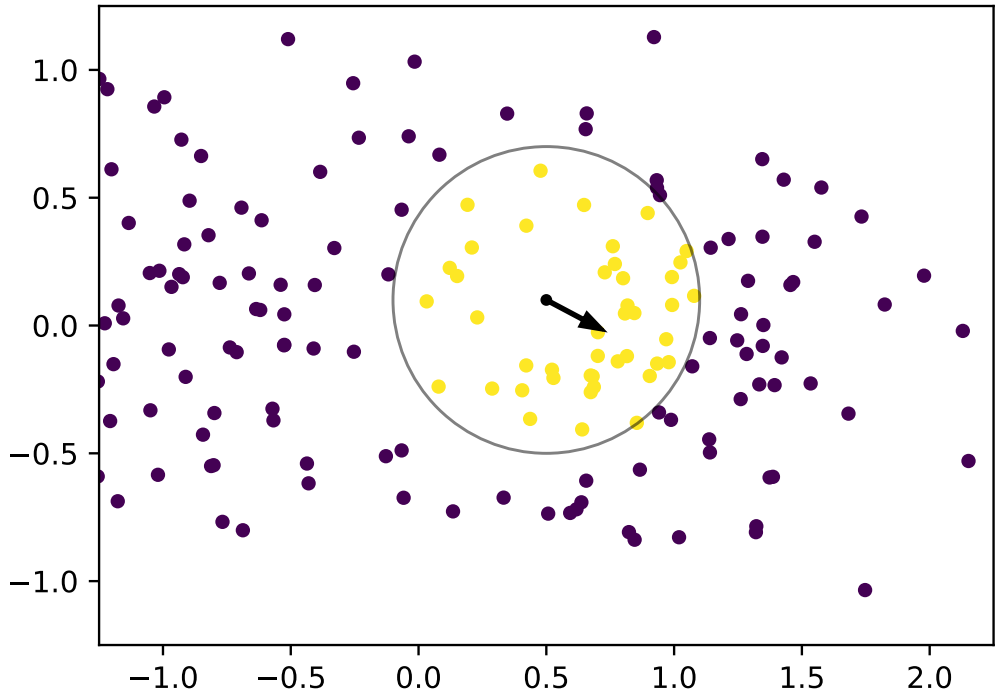
- The choice of bandwidth is a bias-variance tradeoff for the estimate $\hat{f}(\boldsymbol{x})$.
- A small bandwidth results in high variance, a large bandwidth introduces a bias.

Mean shift clustering



- Modes of the kernel density estimate can be used to identify clusters.
- Mode finding procedure results in a non-parametric clustering algorithm.

The idea



- Iteratively move point to “high-density” region.
- Direction is determined by local neighborhood.f

Locally weighted mean

The weighted mean is

$$\boldsymbol{\mu}^* = \frac{\sum_{i=1}^{n} \boldsymbol{x}_i w_i}{\sum_{i=1}^{n} w_i}$$

The **locally weighted mean** has weights depending on the distance to a data point $\boldsymbol{x}$

$$\boldsymbol{\mu}^*(\boldsymbol{x}) = \frac{\sum_{i=1}^{n} \boldsymbol{x}_i \; g\left(\left\| \frac{\boldsymbol{x}-\boldsymbol{x}_i}{h} \right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\| \frac{\boldsymbol{x}-\boldsymbol{x}_i}{h} \right\|^2\right)}$$

Mean shift procedure

The **mean shift vector** is defined as

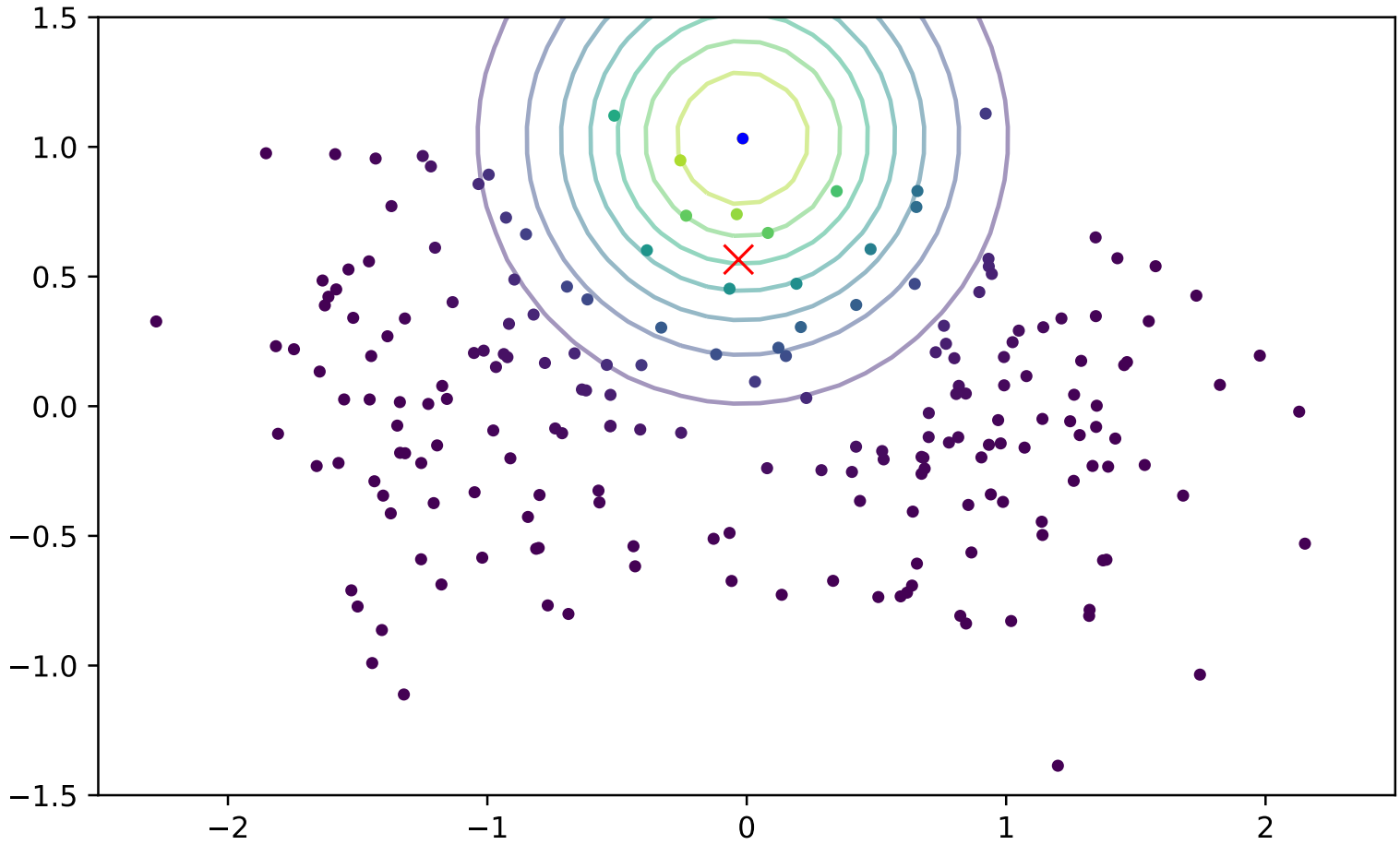
$$\boldsymbol{m}(\boldsymbol{x}) = \boldsymbol{\mu}^*(\boldsymbol{x}) - \boldsymbol{x}$$

The **mean shift procedure** is

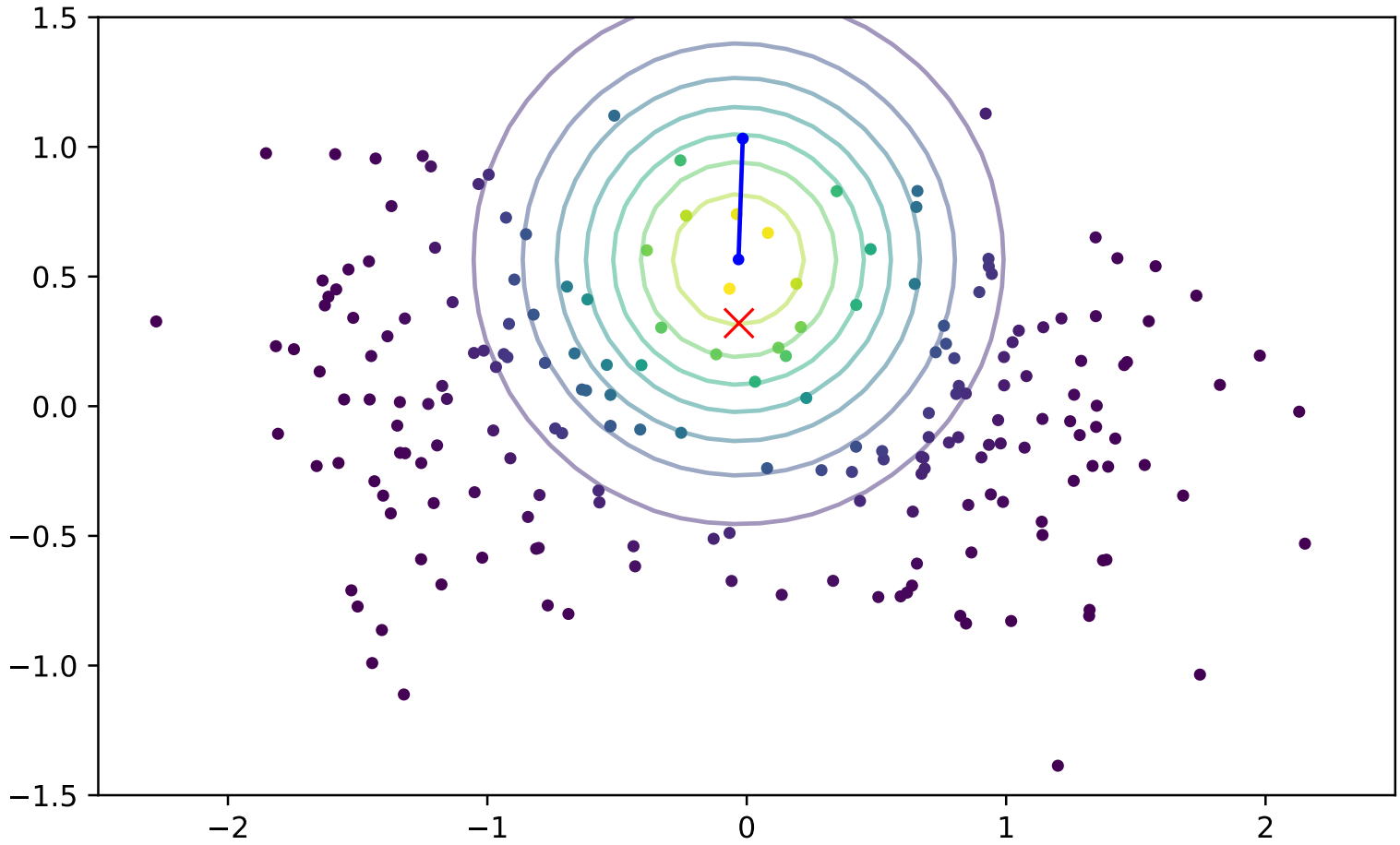
$$\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \boldsymbol{m}\left(\boldsymbol{x}^{(t)}\right) \qquad \text{for } i = 1, 2, \ldots$$

- When the process converges we assign $\boldsymbol{x}$ to the mode $\boldsymbol{x}^{(\infty)}$.
- Points at the same mode are considered to be in the same cluster.

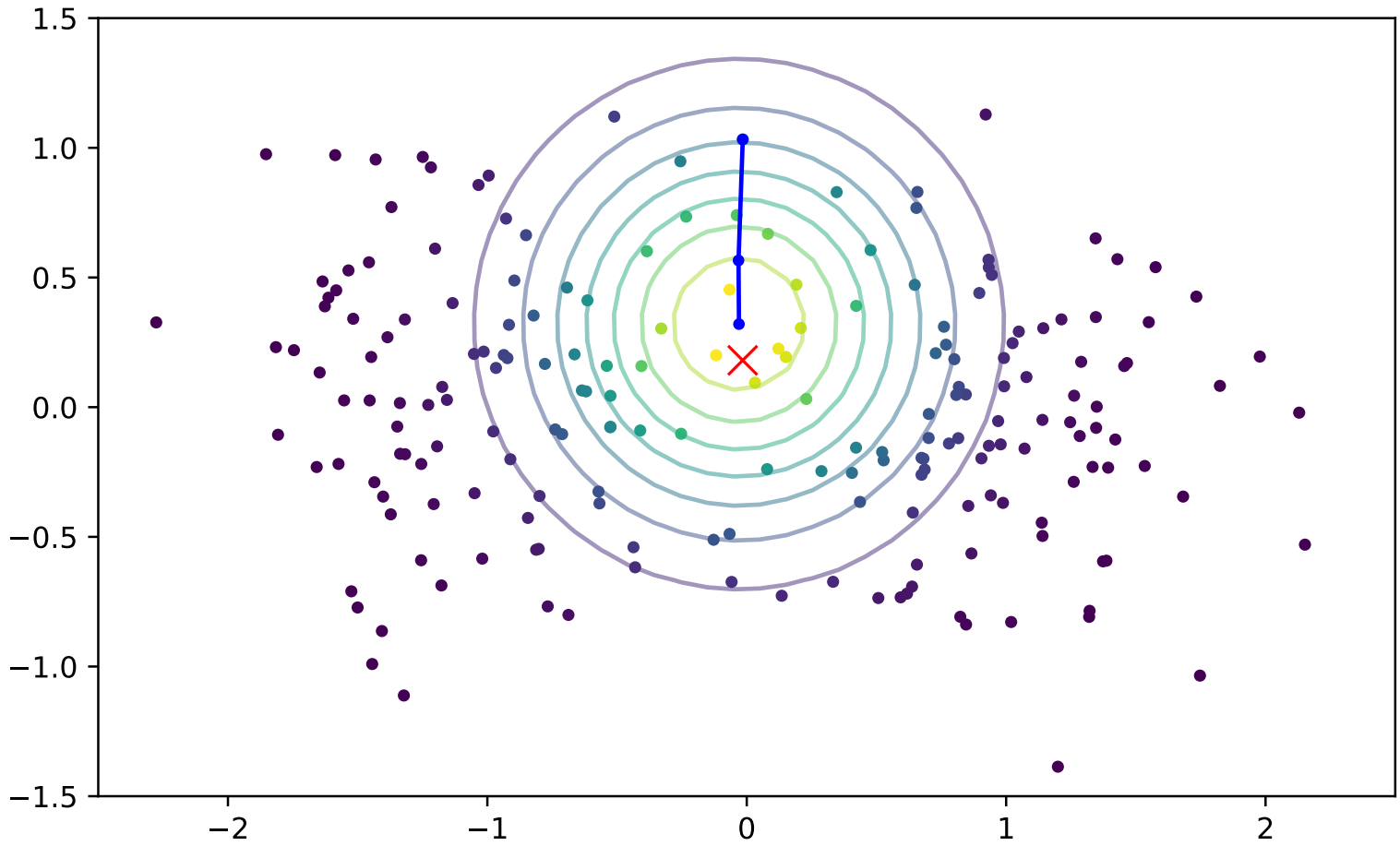
Convergence – Gaussian kernel



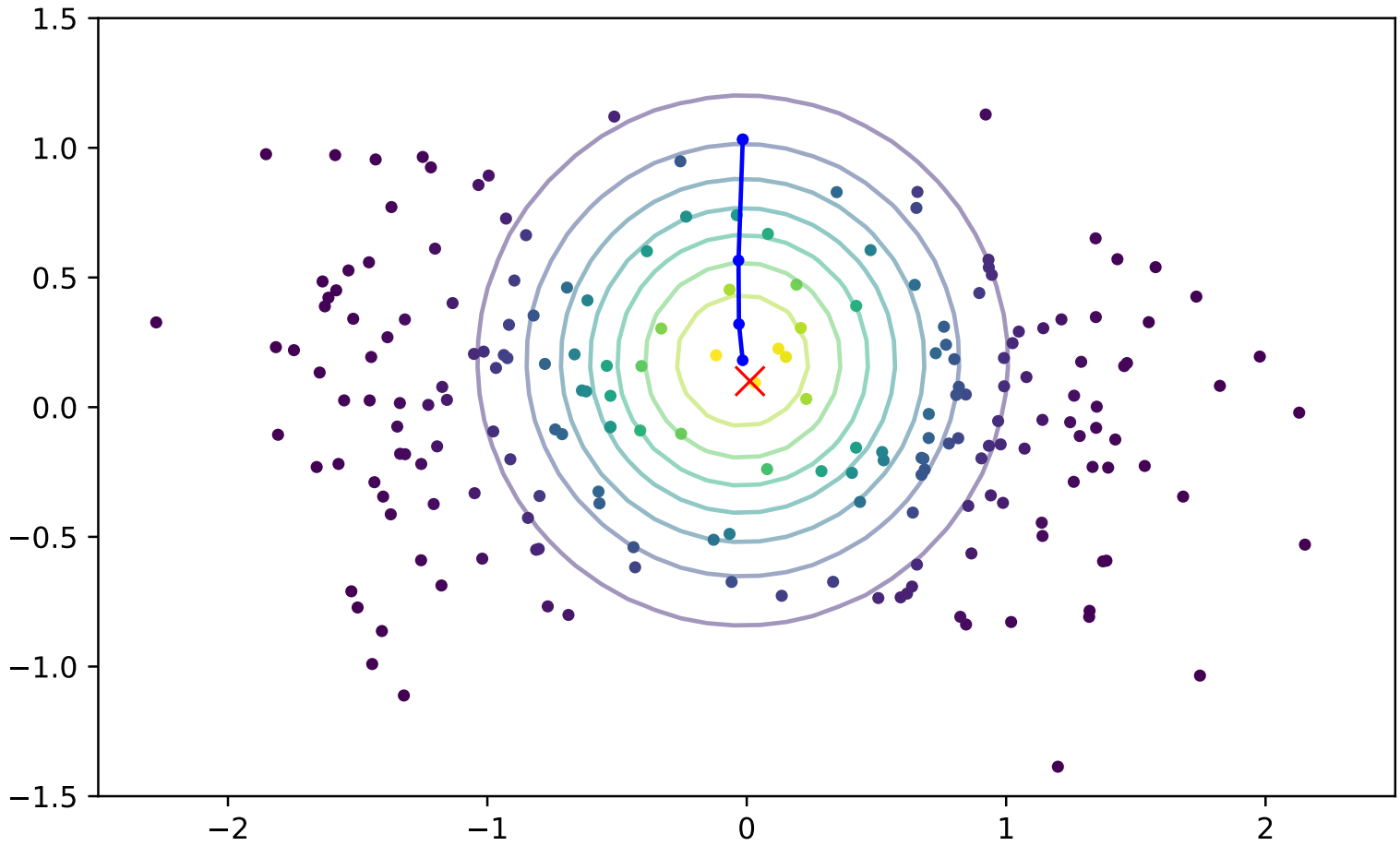
Convergence – Gaussian kernel



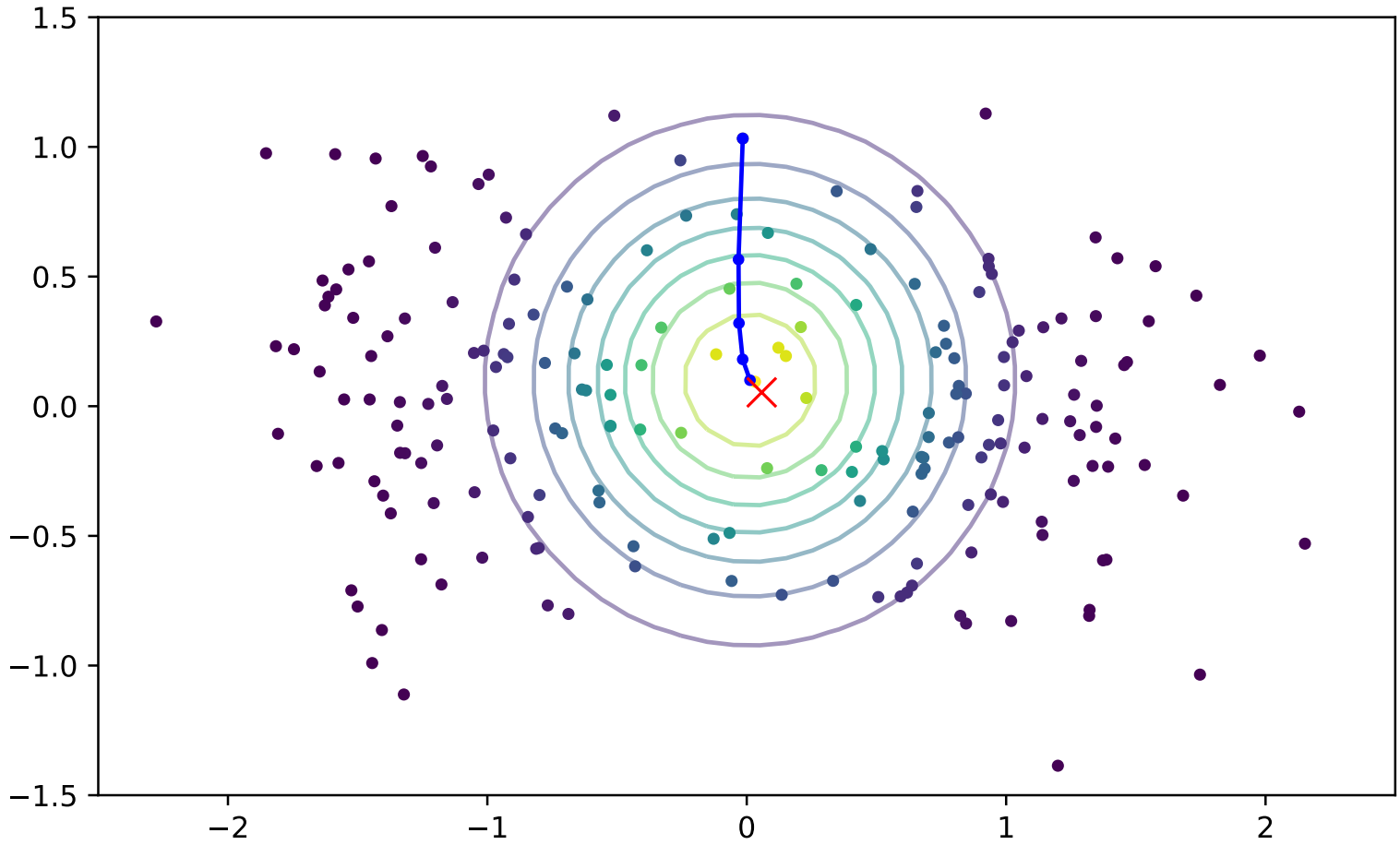
Convergence – Gaussian kernel



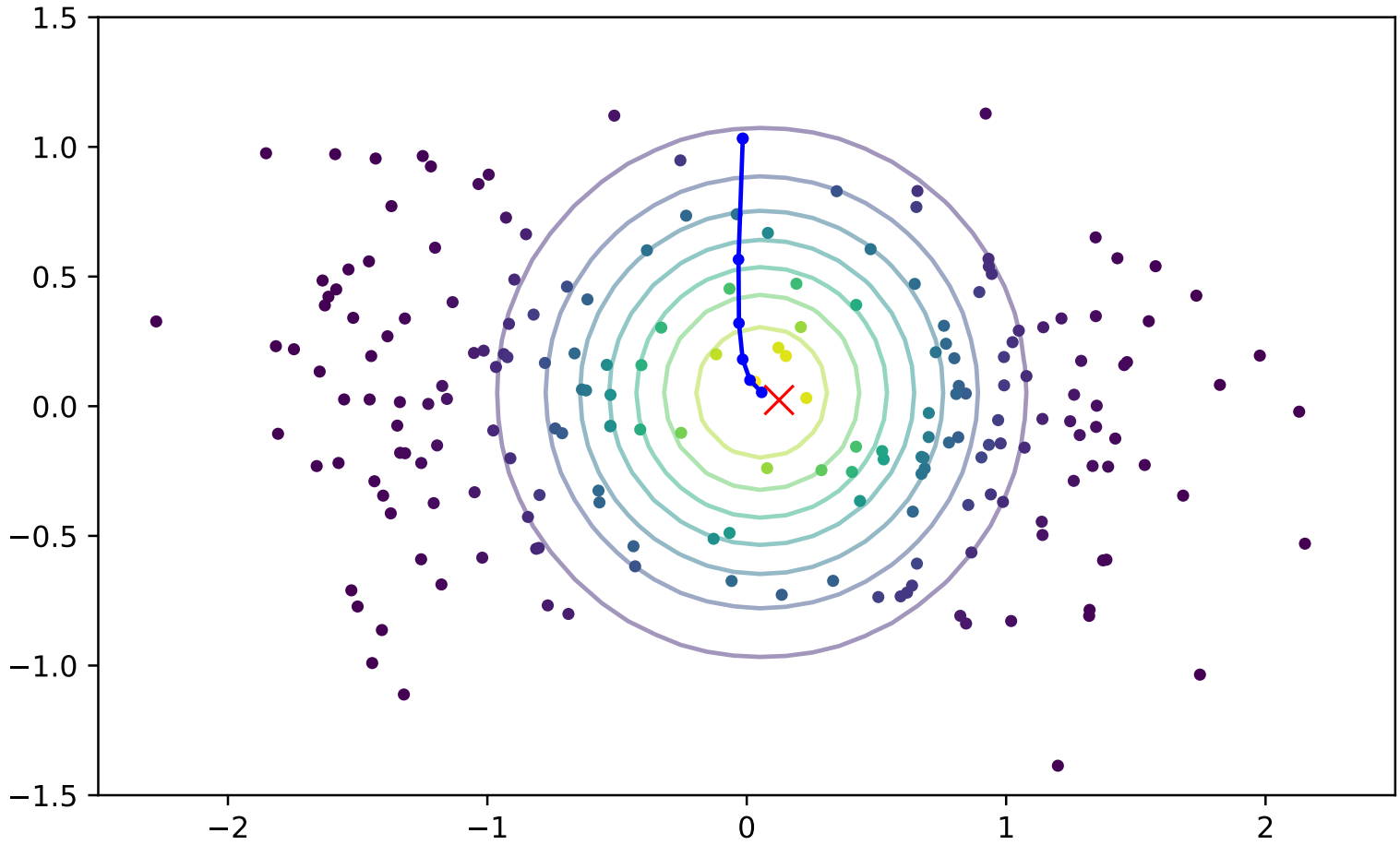
Convergence – Gaussian kernel



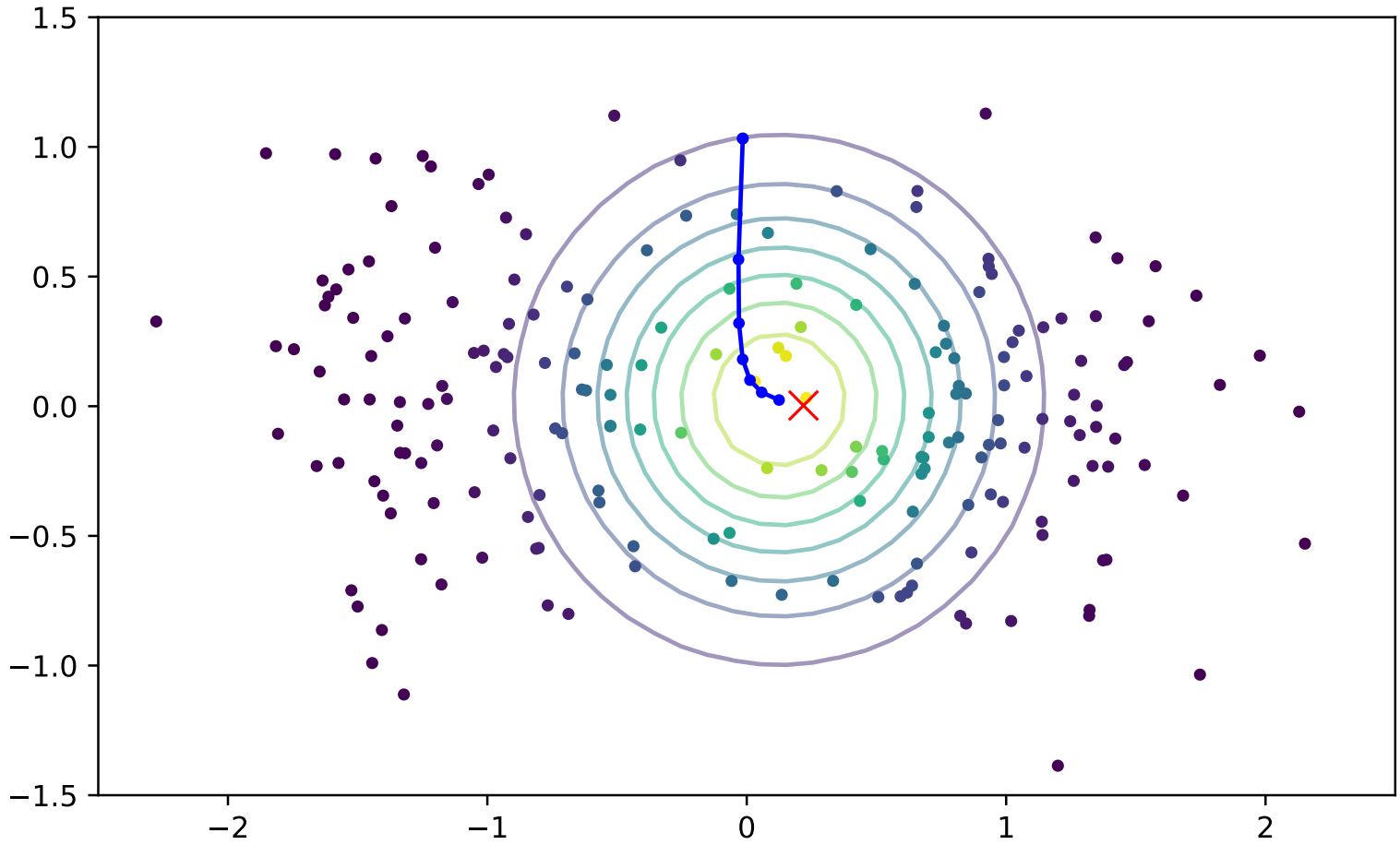
Convergence – Gaussian kernel



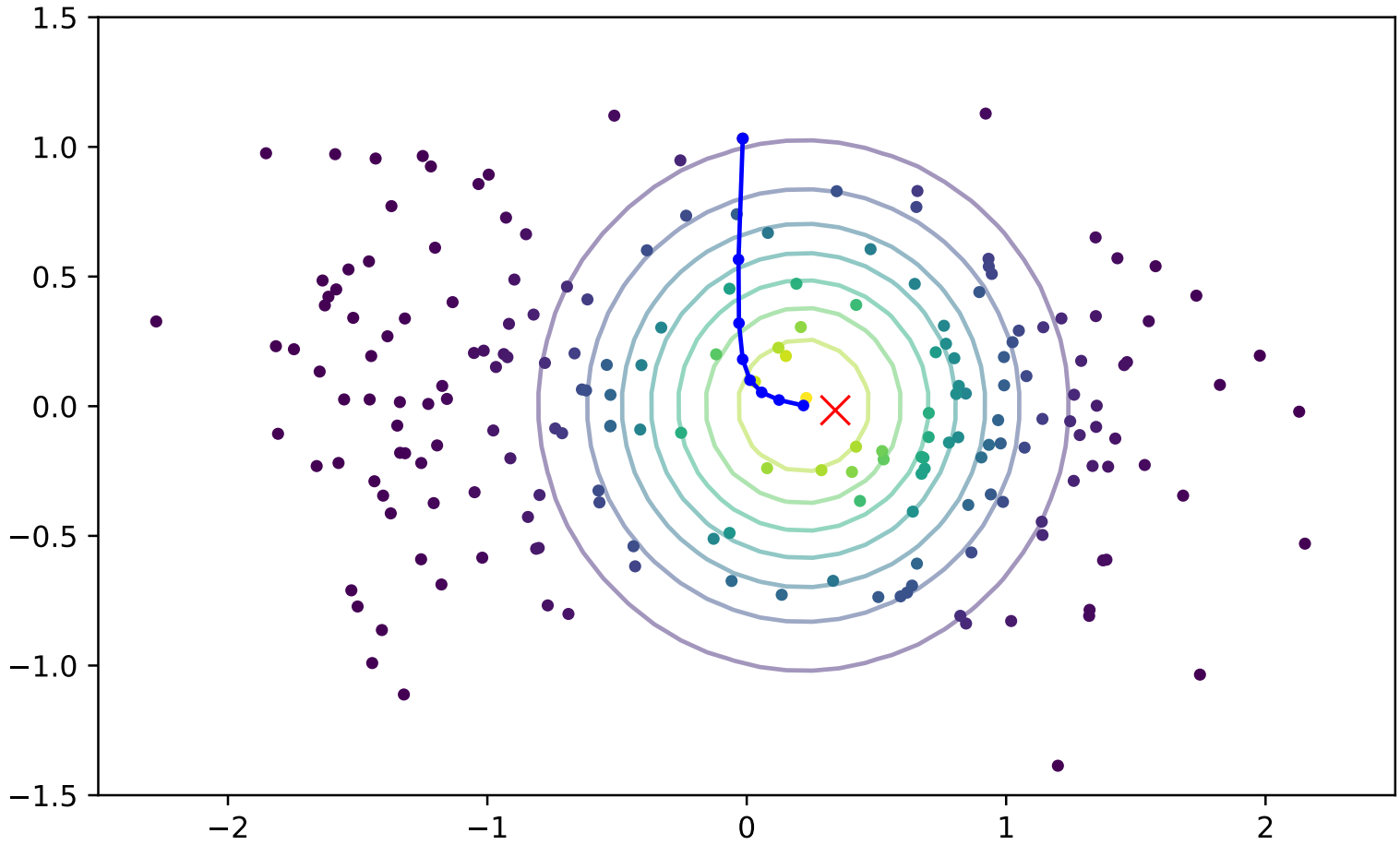
Convergence – Gaussian kernel



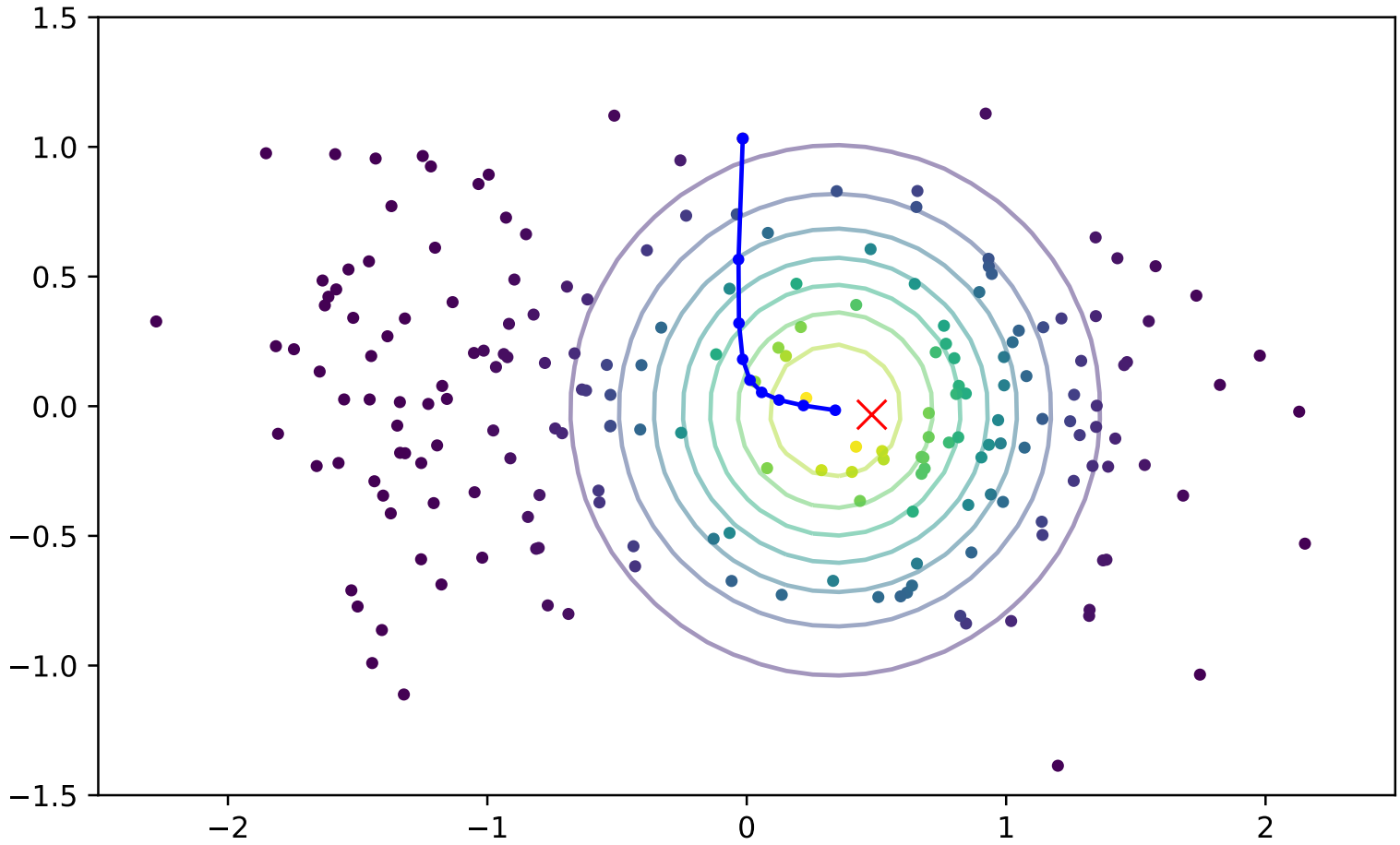
Convergence – Gaussian kernel



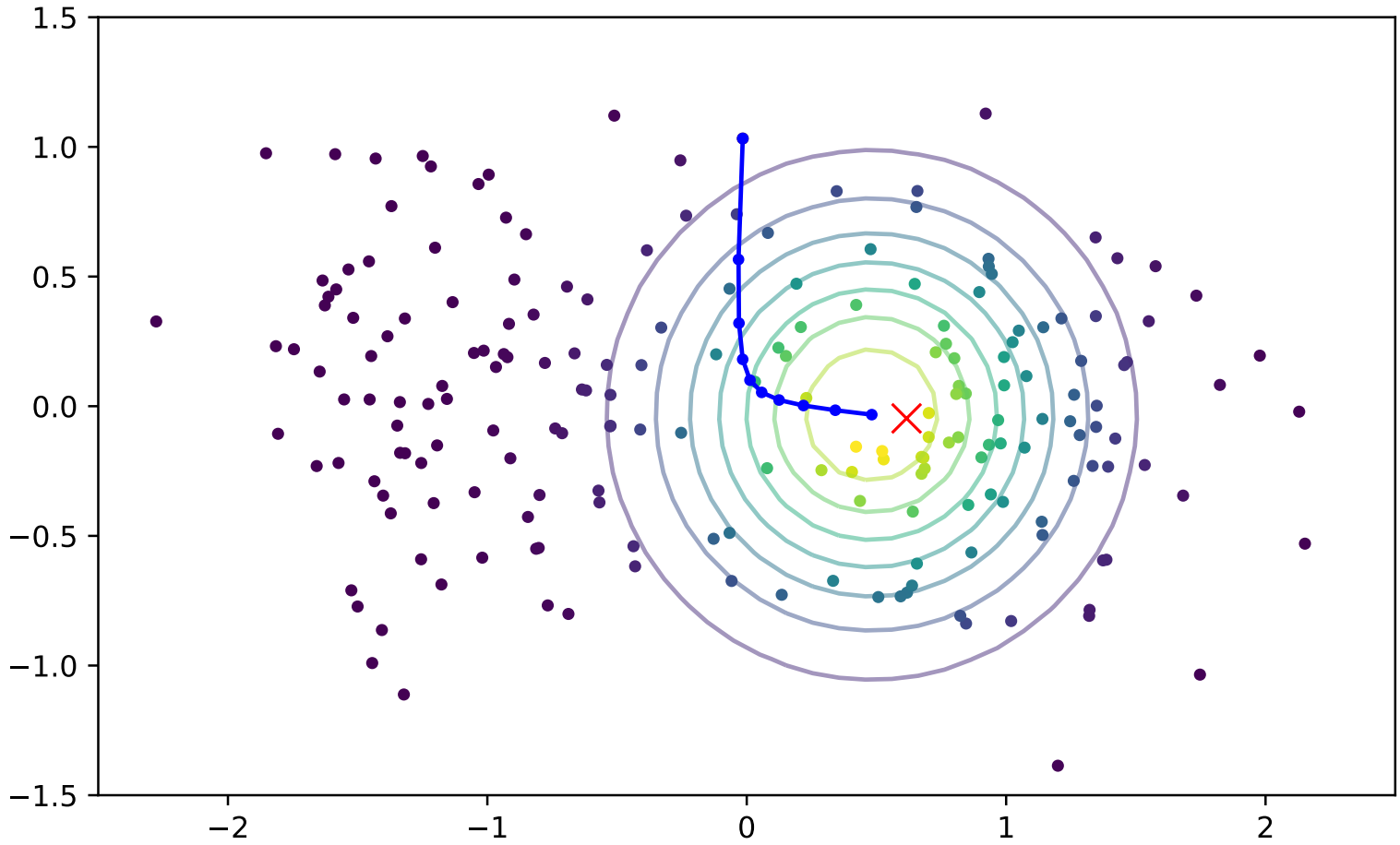
Convergence – Gaussian kernel



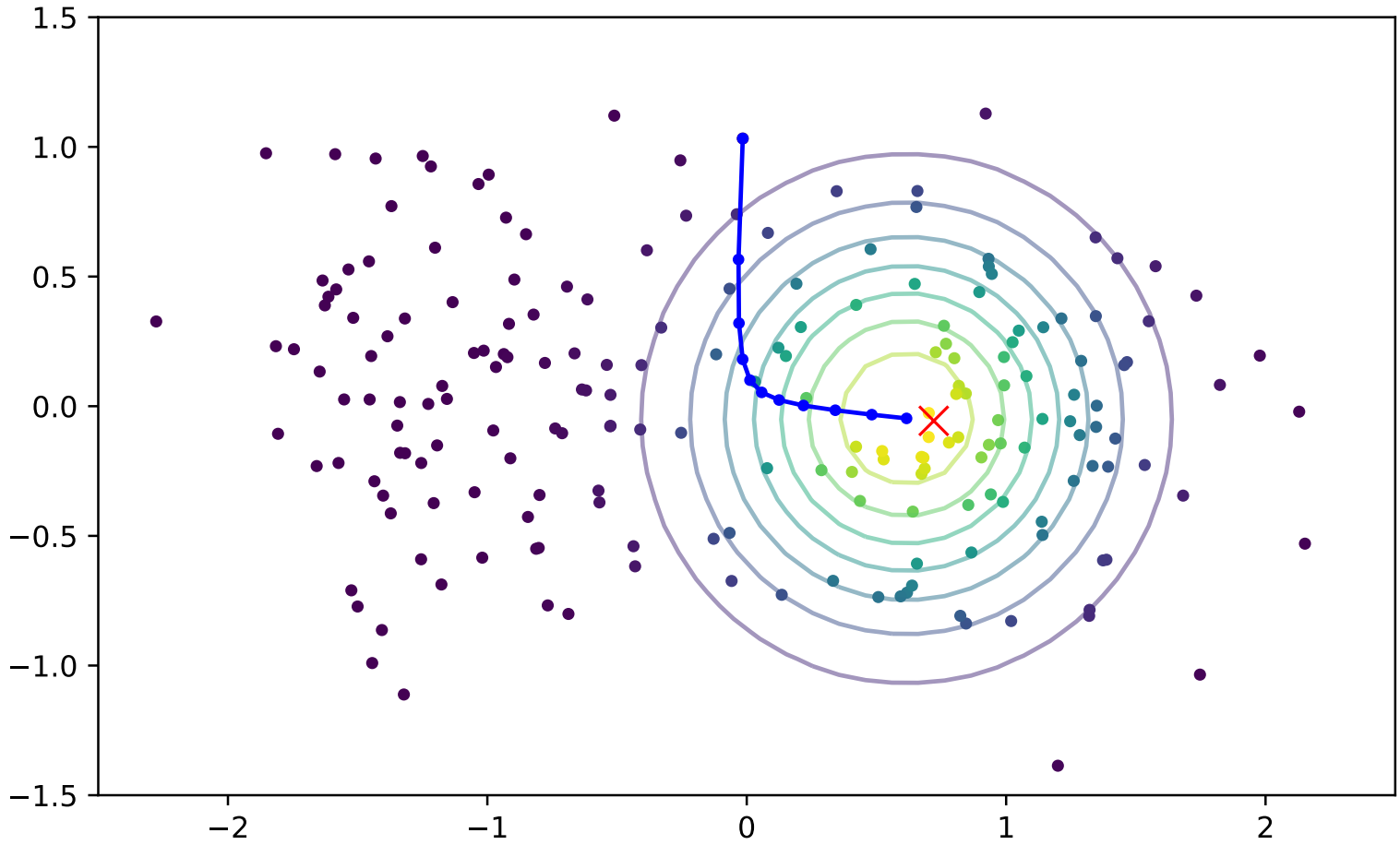
Convergence – Gaussian kernel



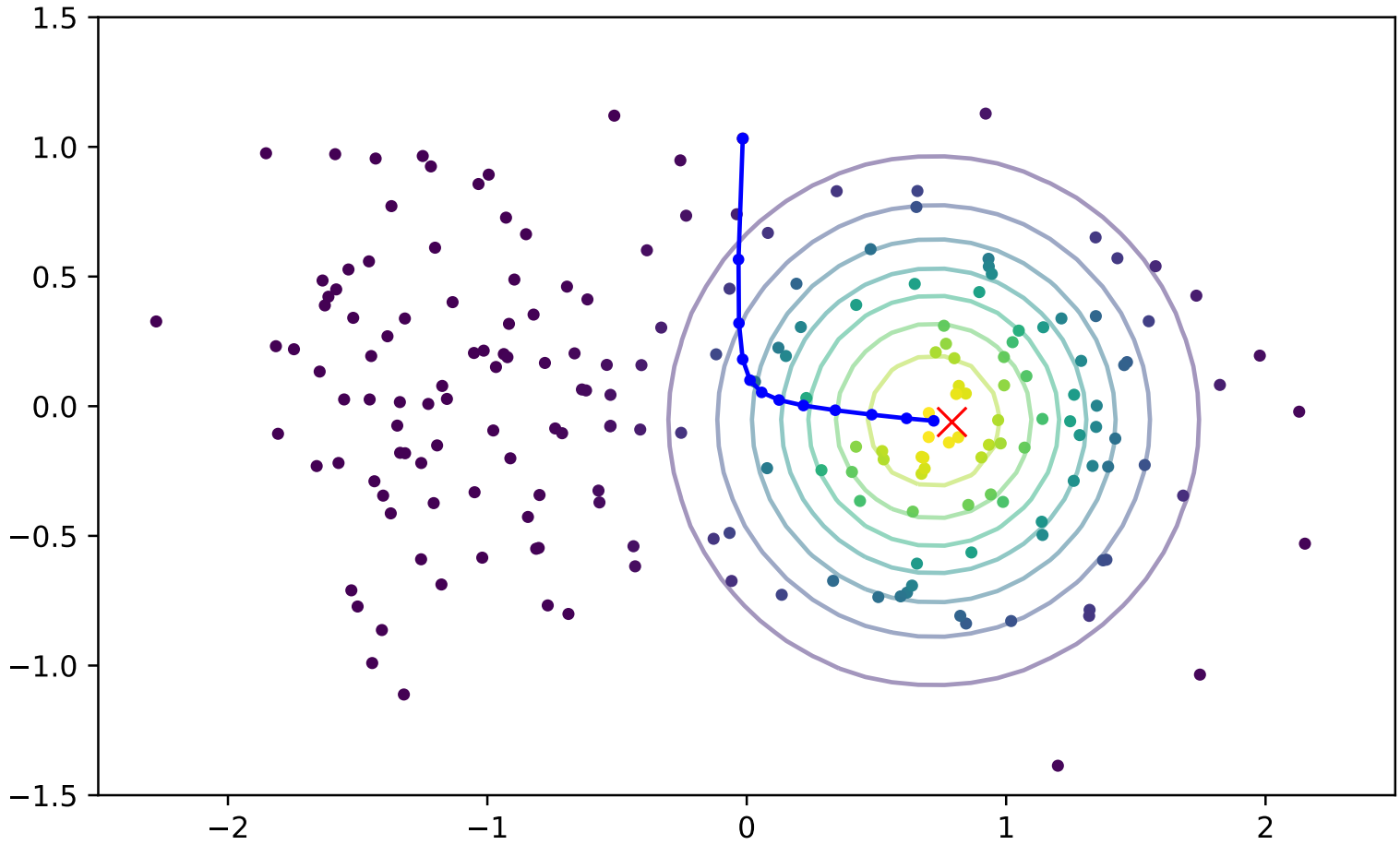
Convergence – Gaussian kernel



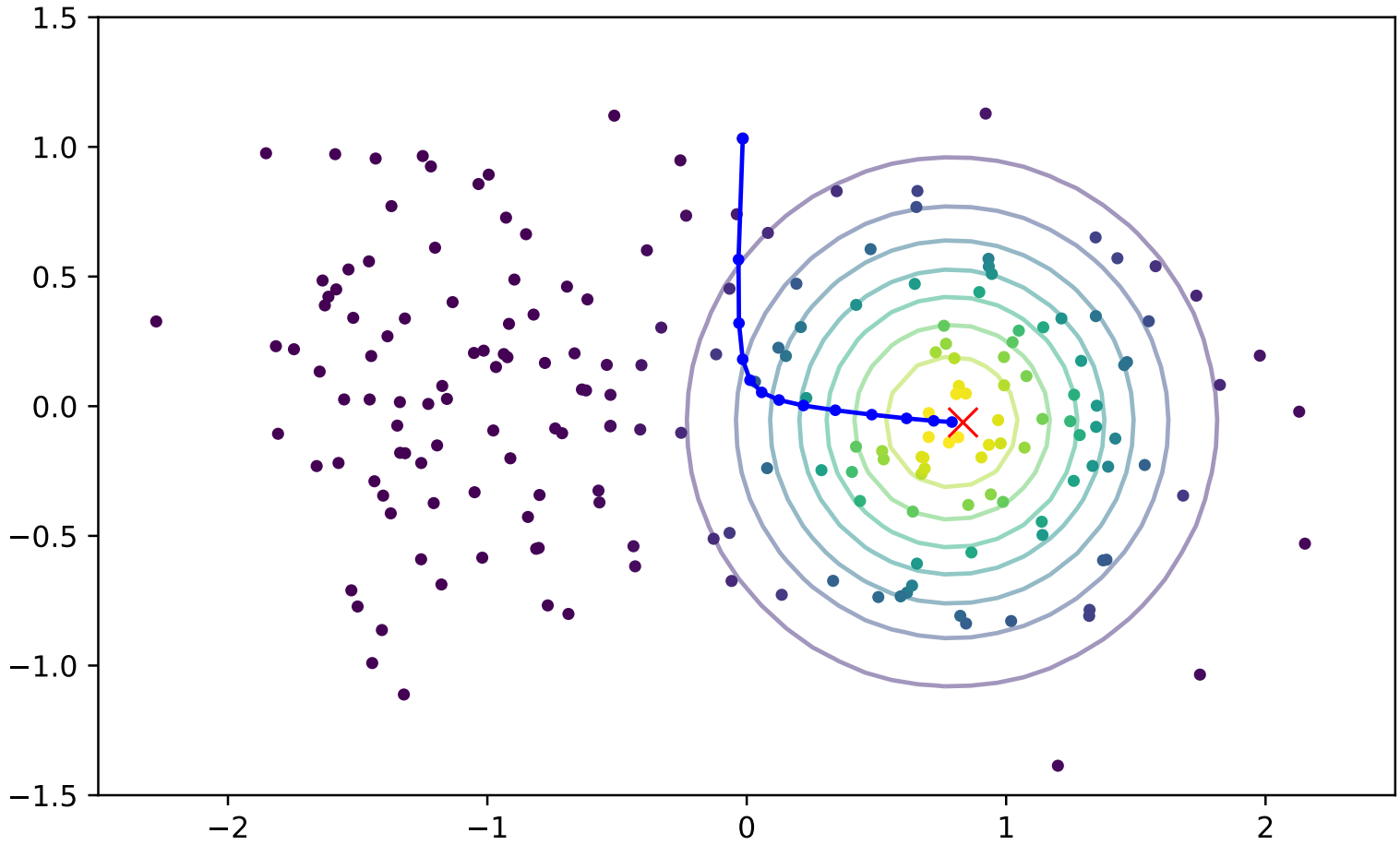
Convergence – Gaussian kernel



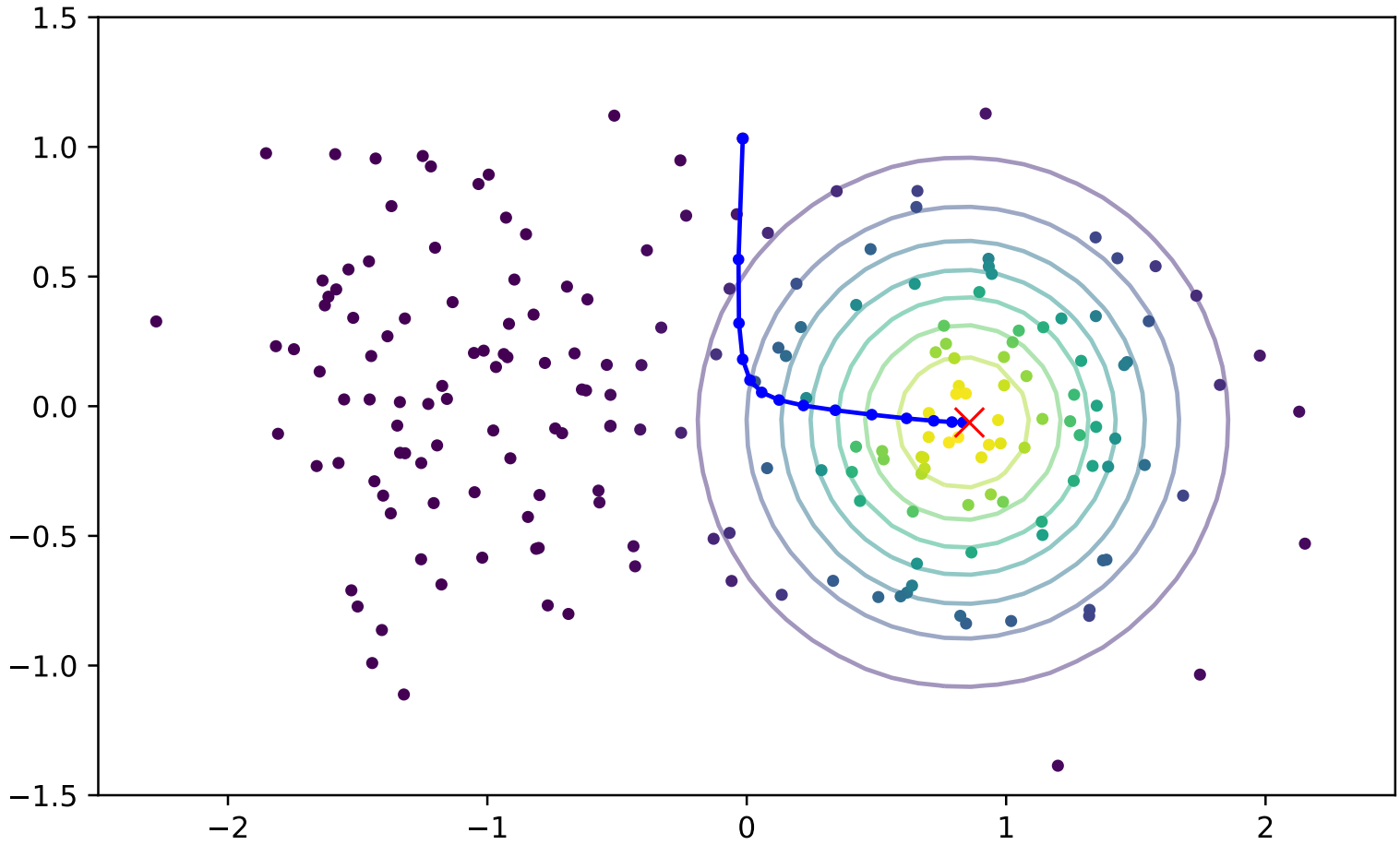
Convergence – Gaussian kernel



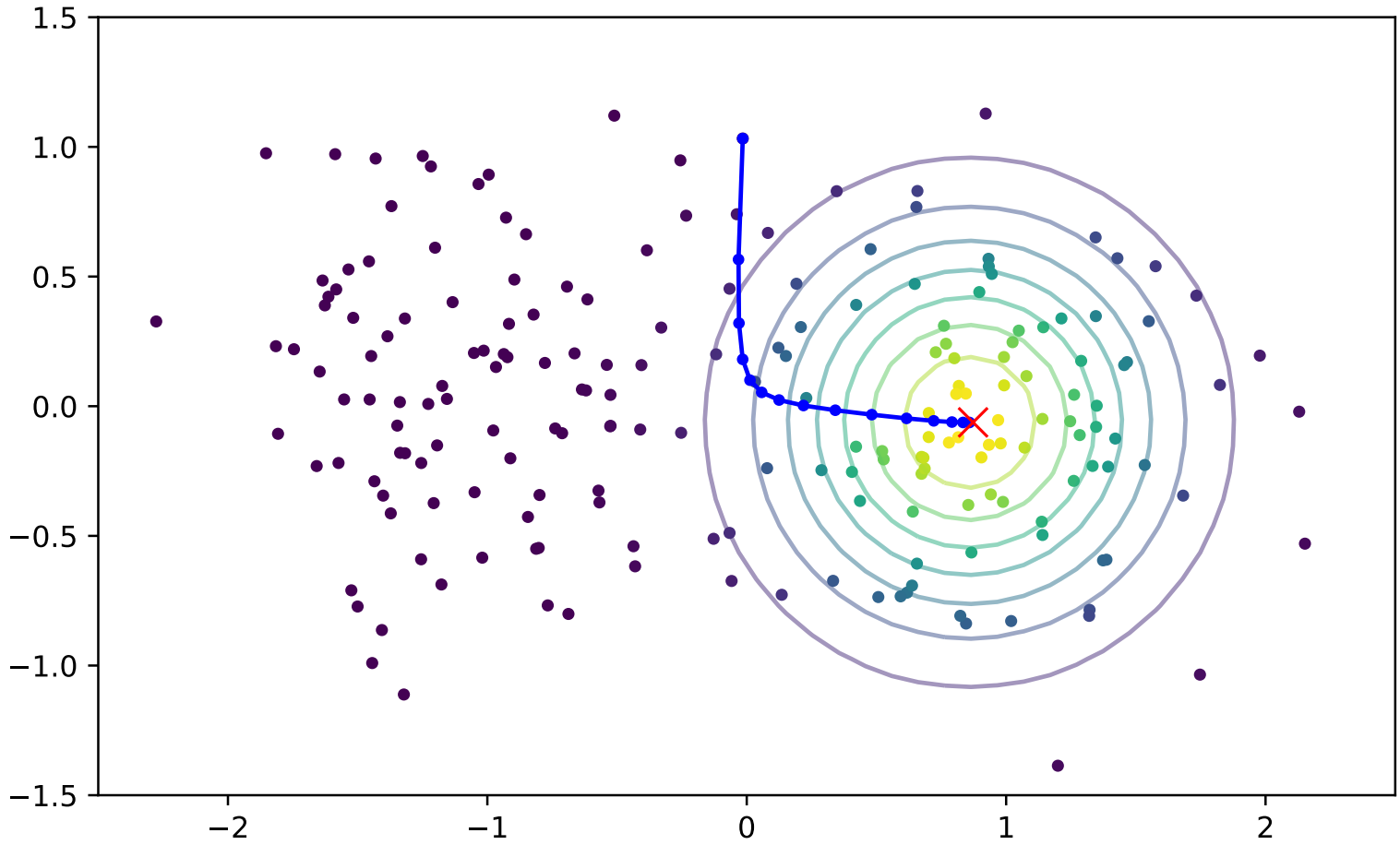
Convergence – Gaussian kernel



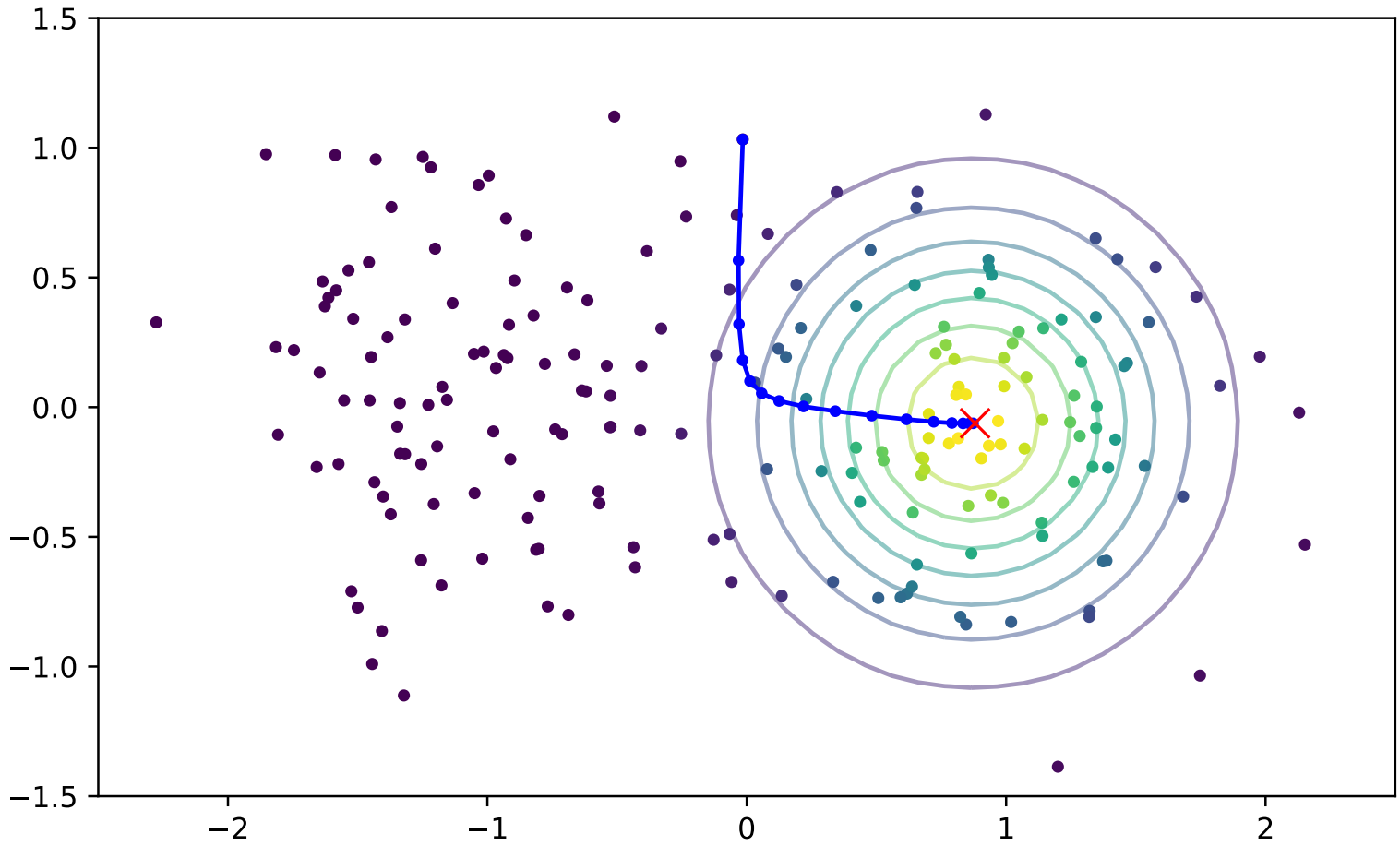
Convergence – Gaussian kernel



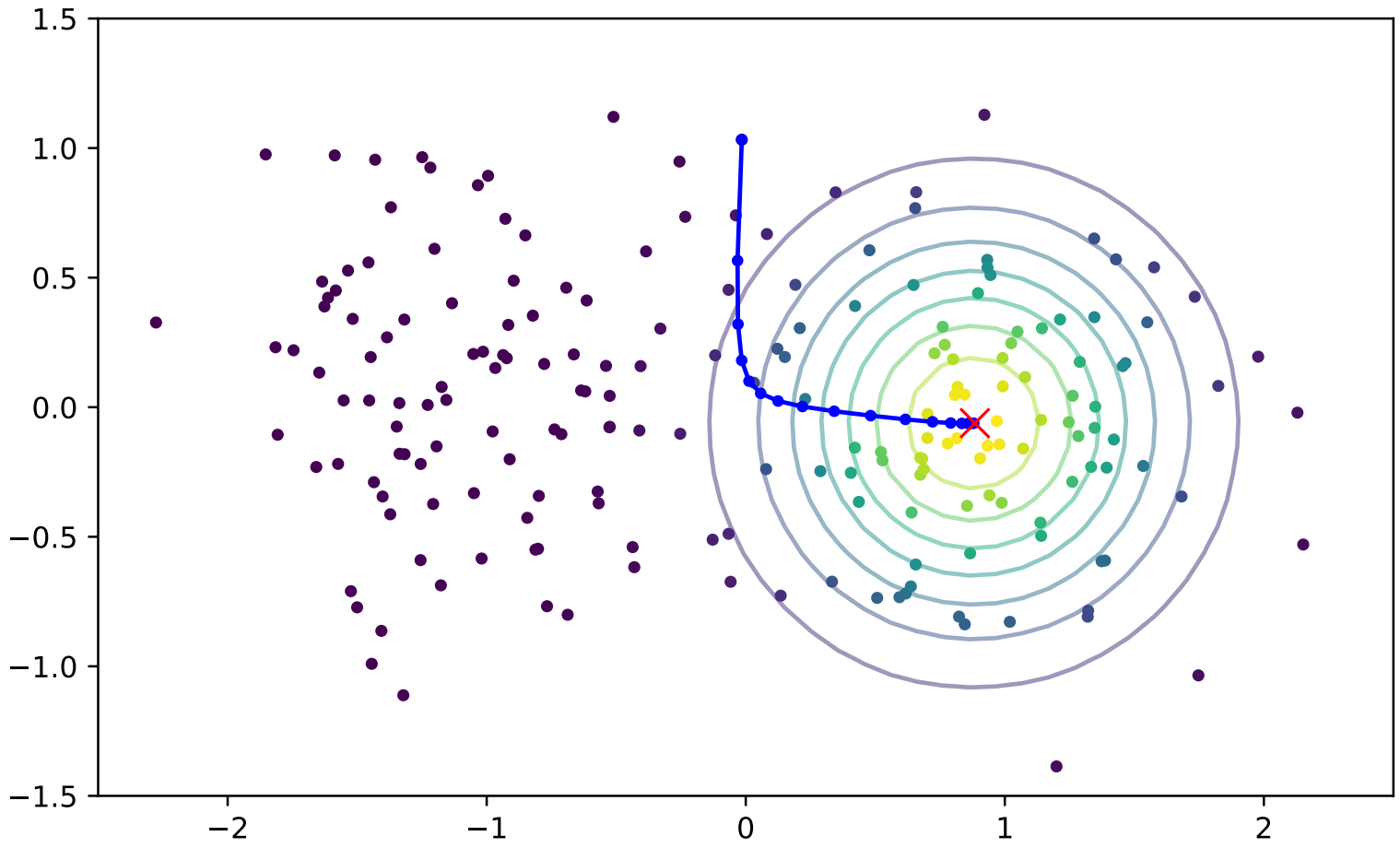
Convergence – Gaussian kernel



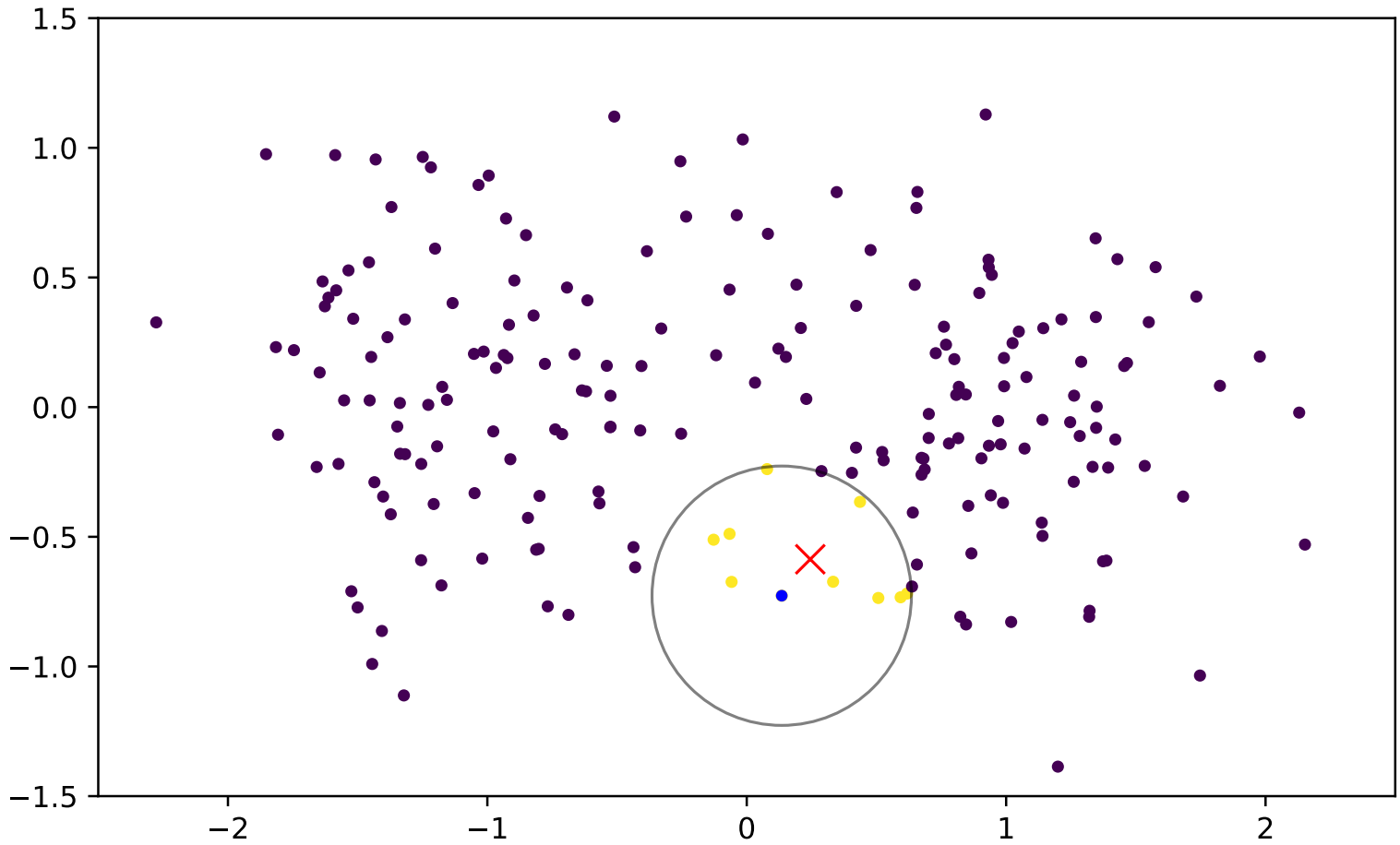
Convergence – Gaussian kernel



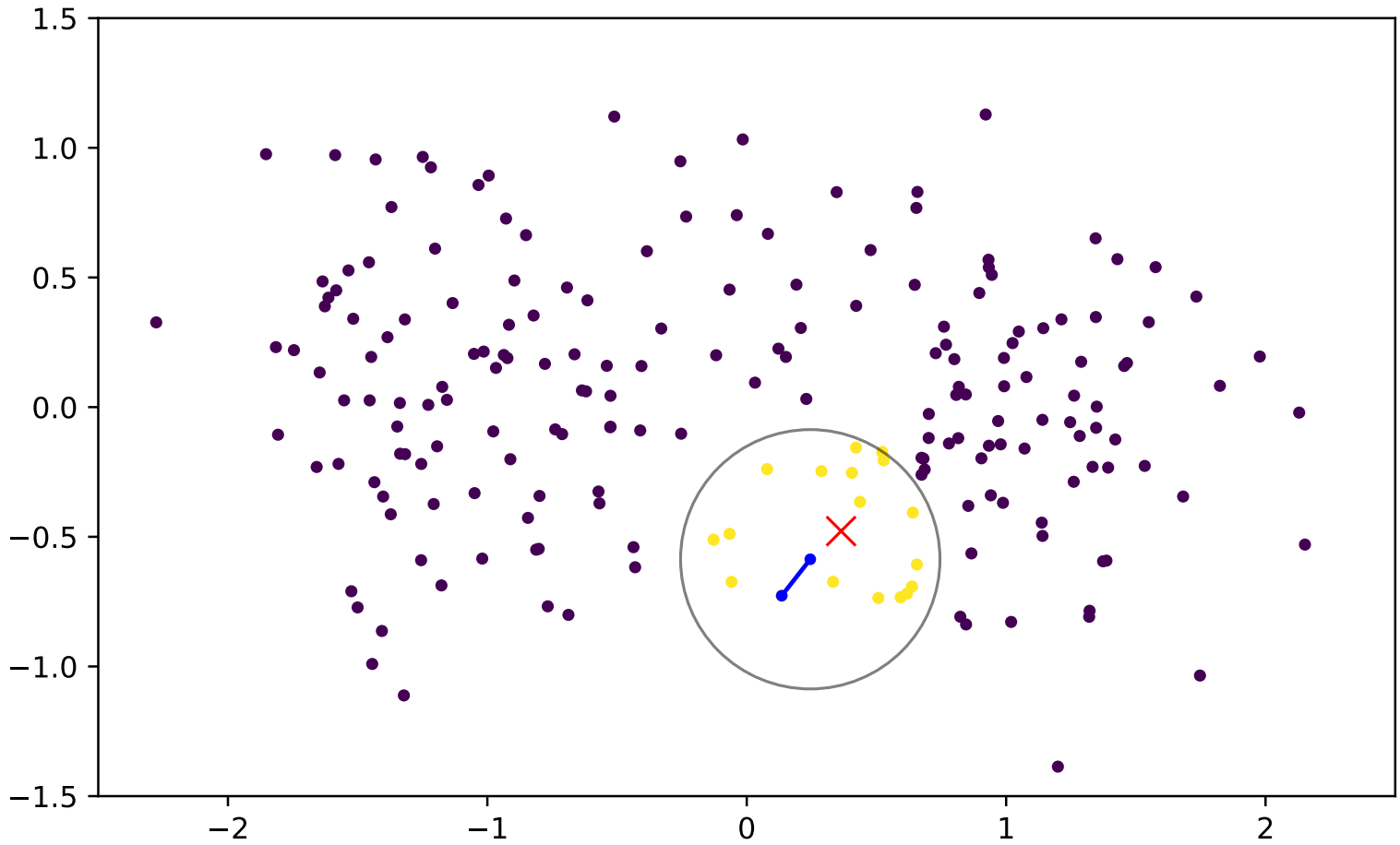
Convergence – Gaussian kernel



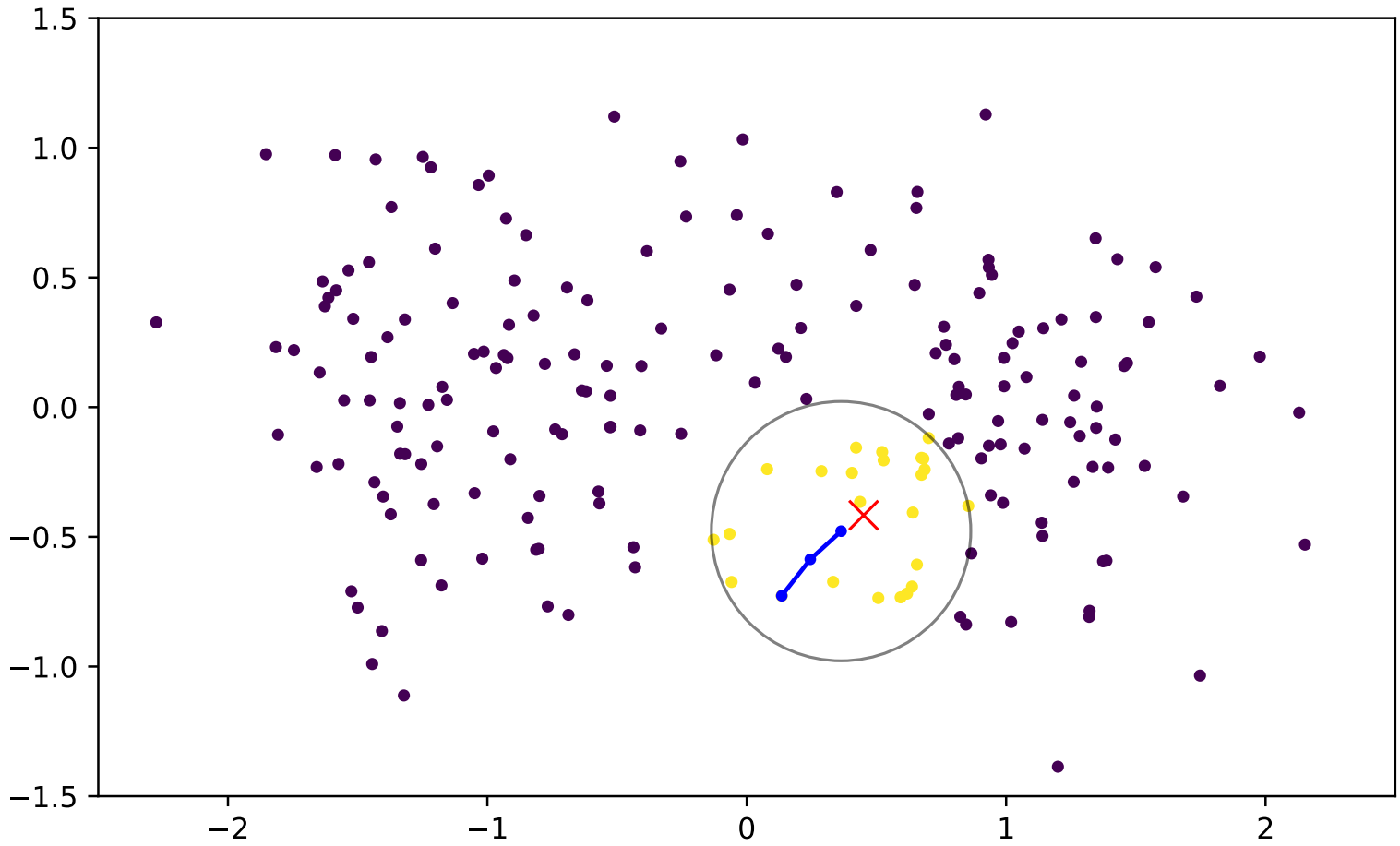
Convergence – Uniform kernel



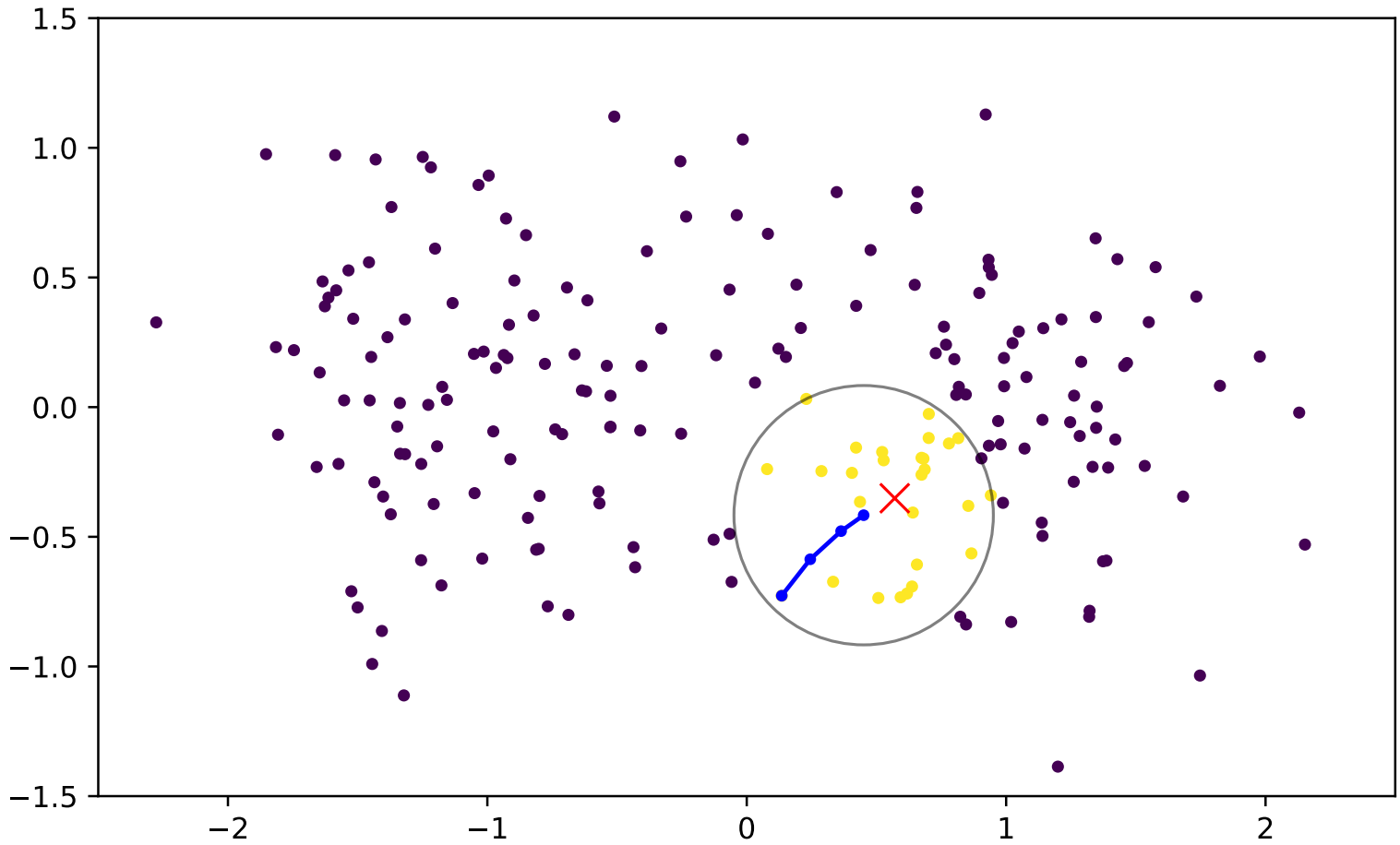
Convergence – Uniform kernel



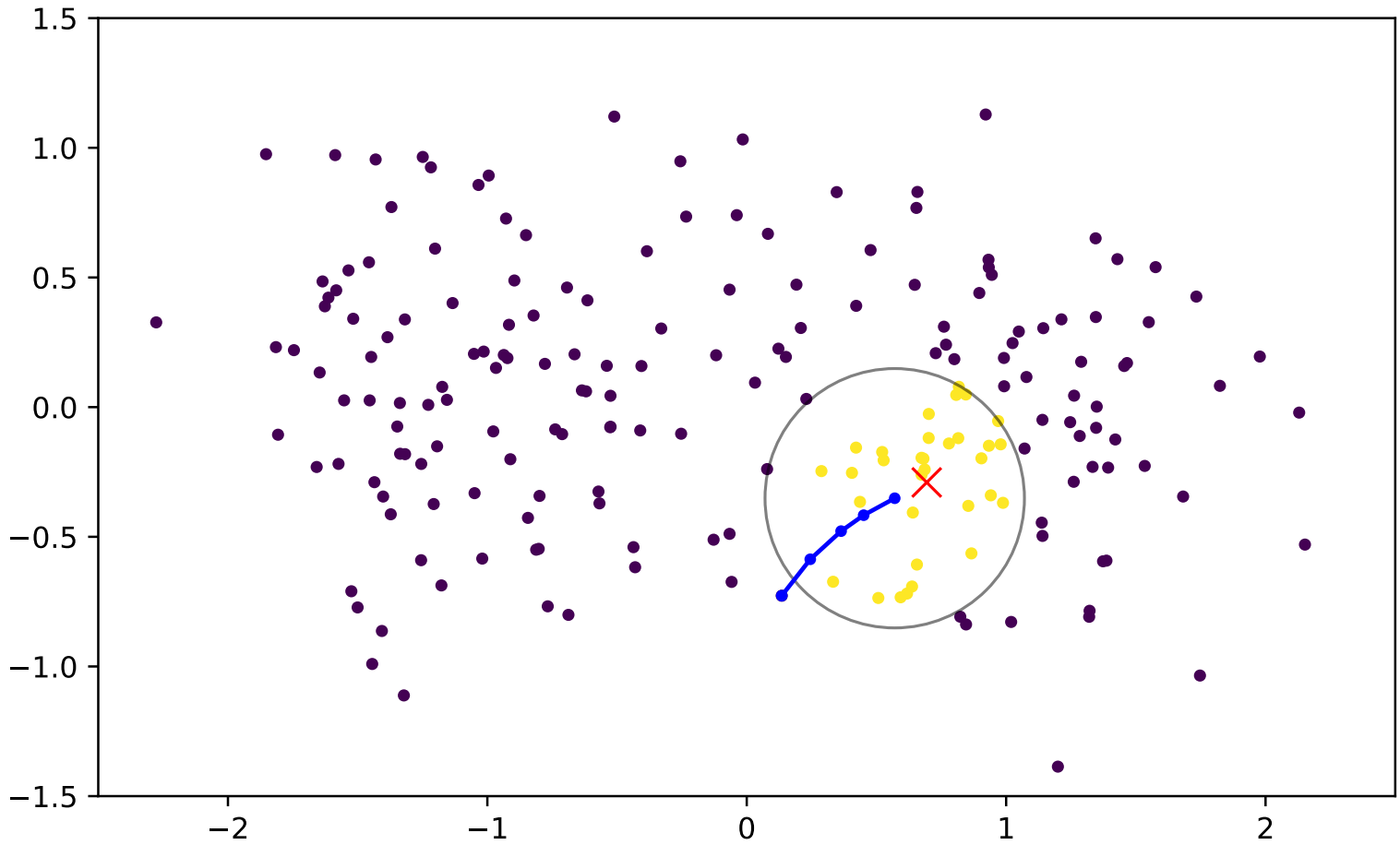
Convergence – Uniform kernel



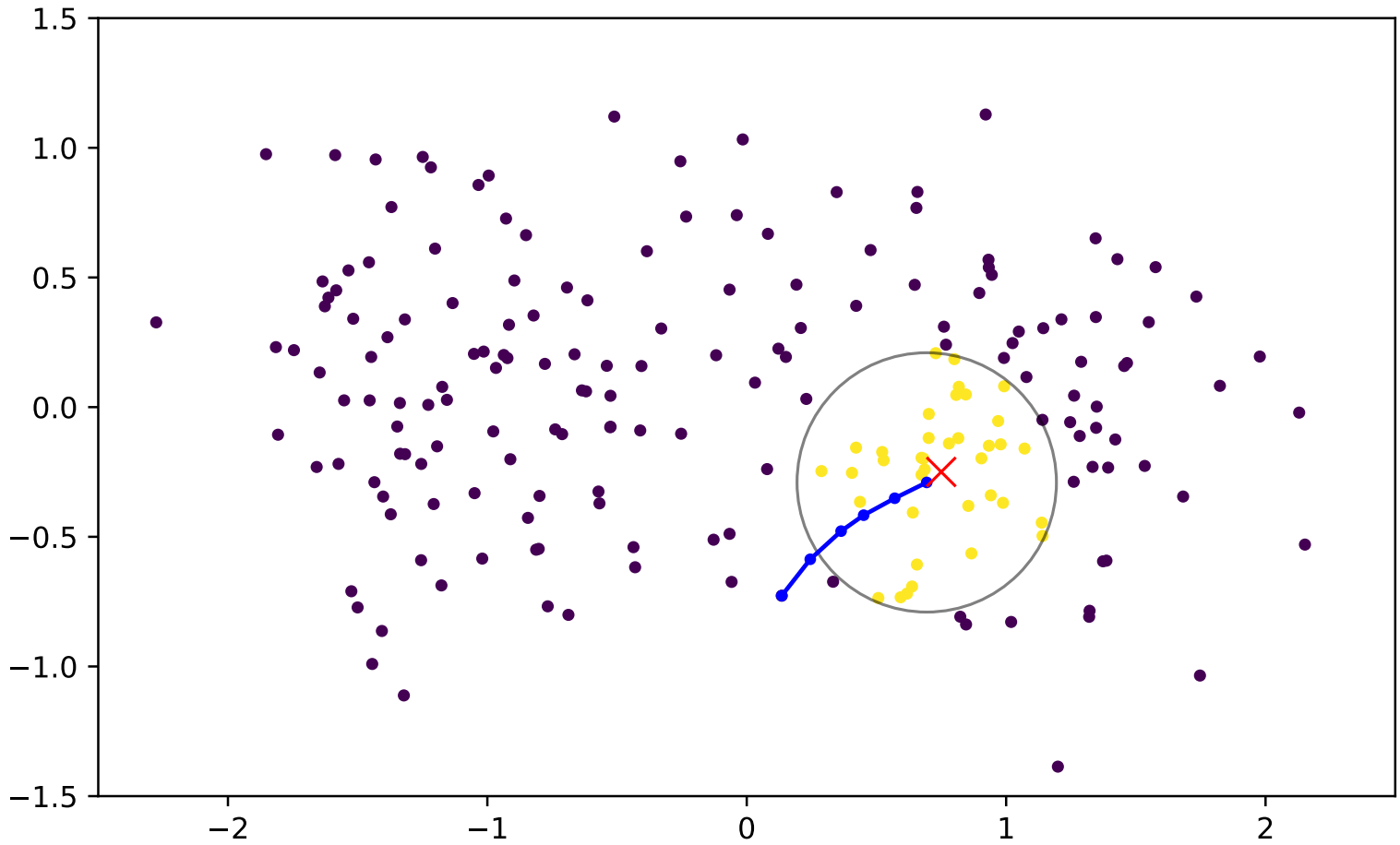
Convergence – Uniform kernel



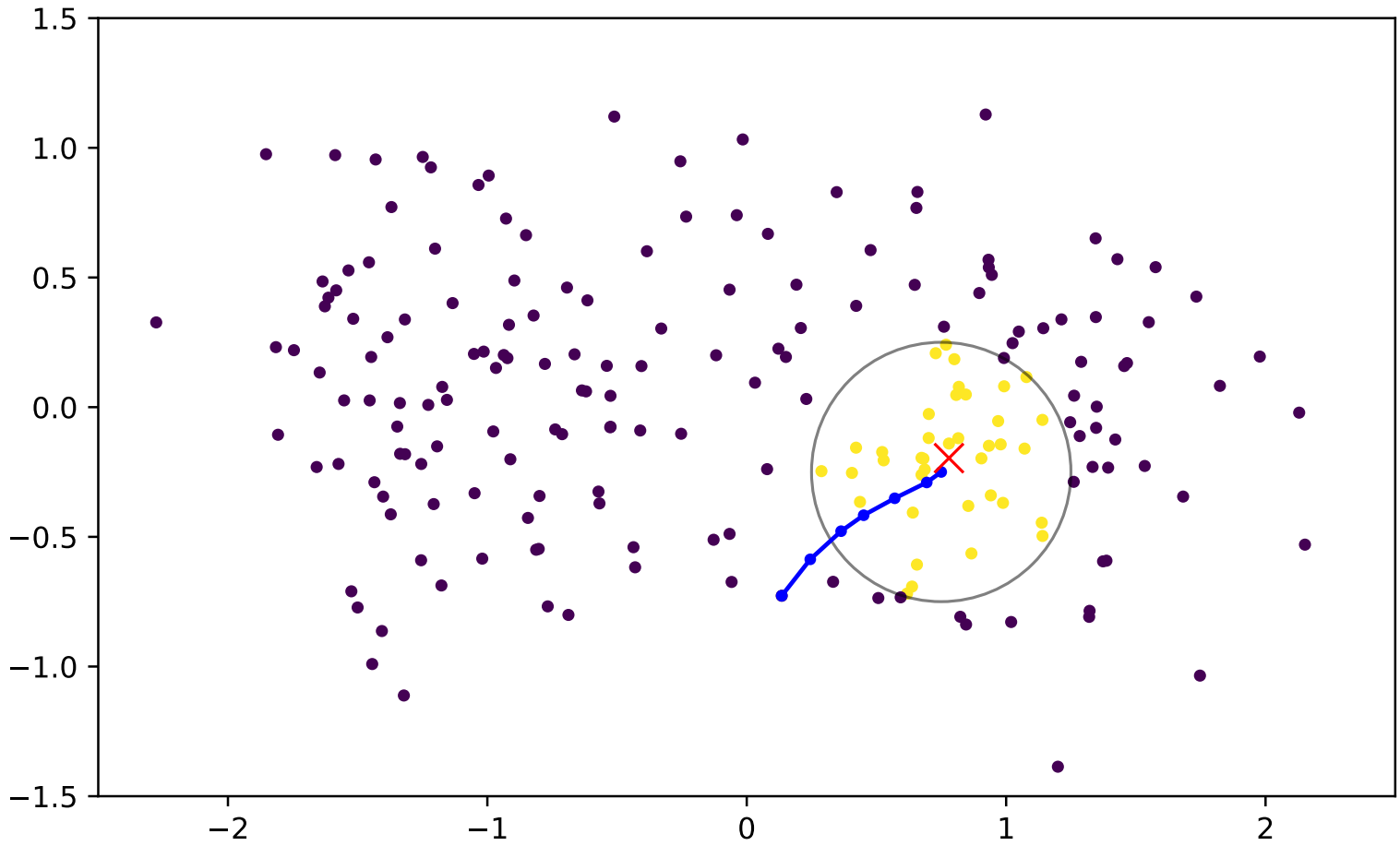
Convergence – Uniform kernel



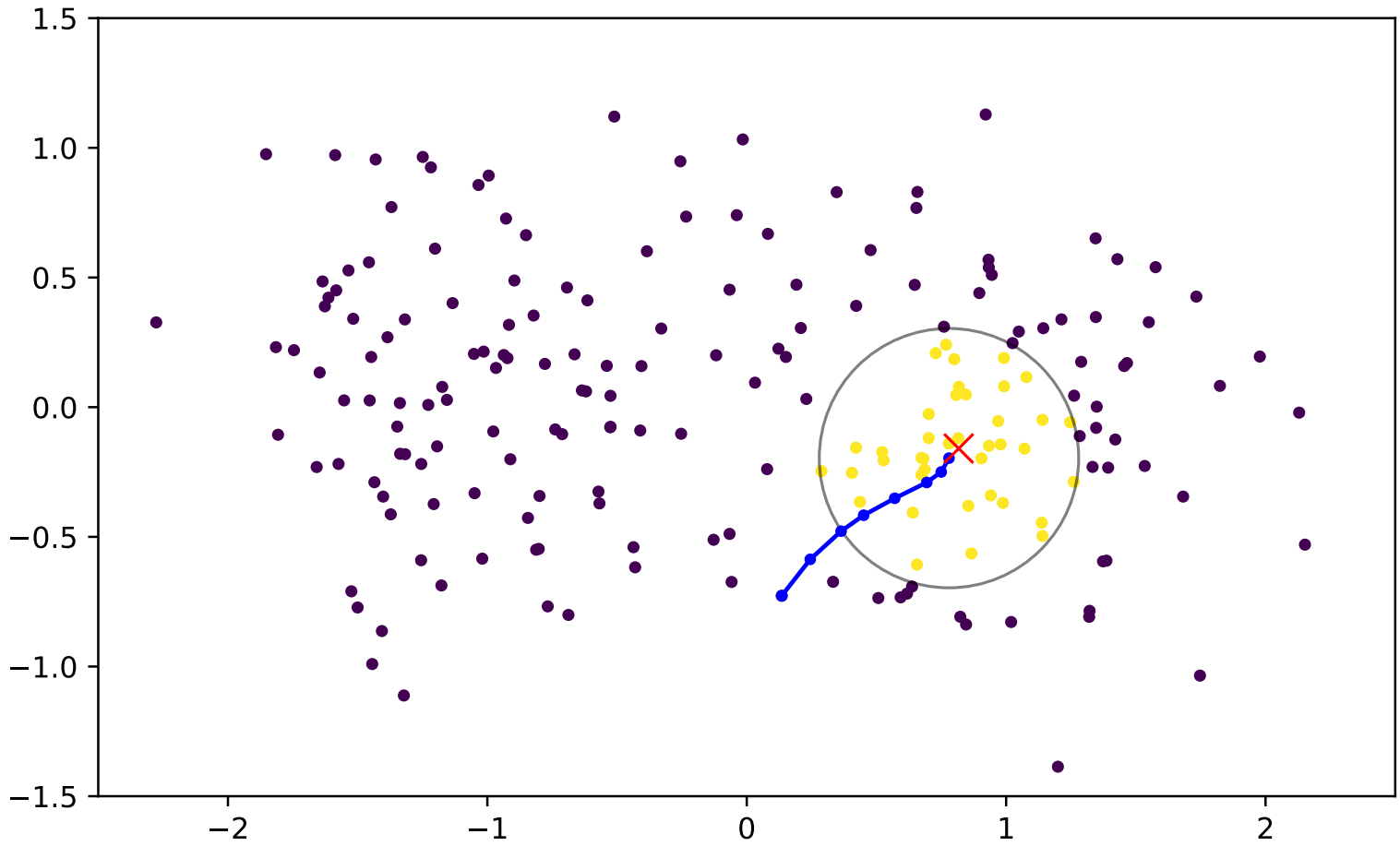
Convergence – Uniform kernel



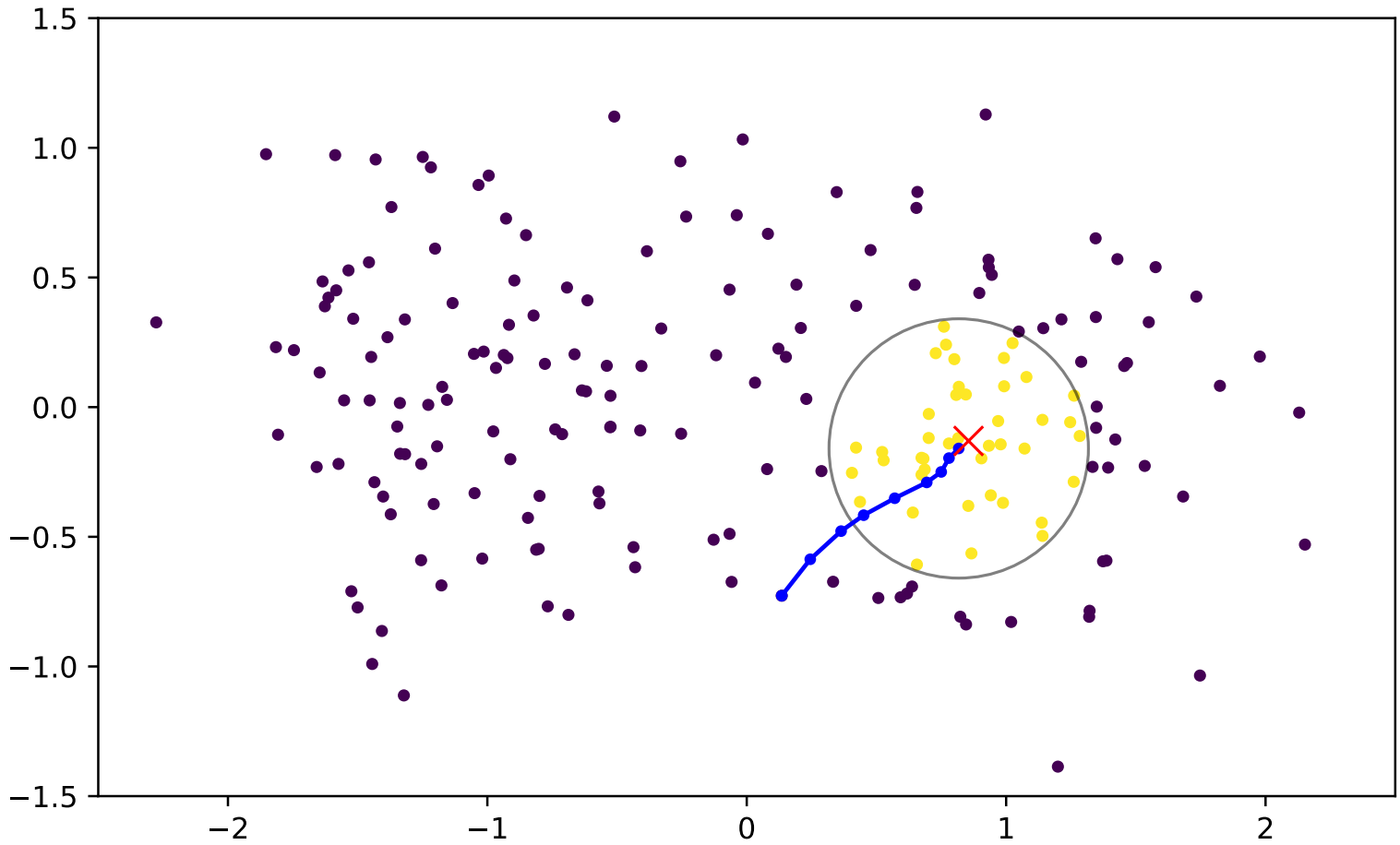
Convergence – Uniform kernel



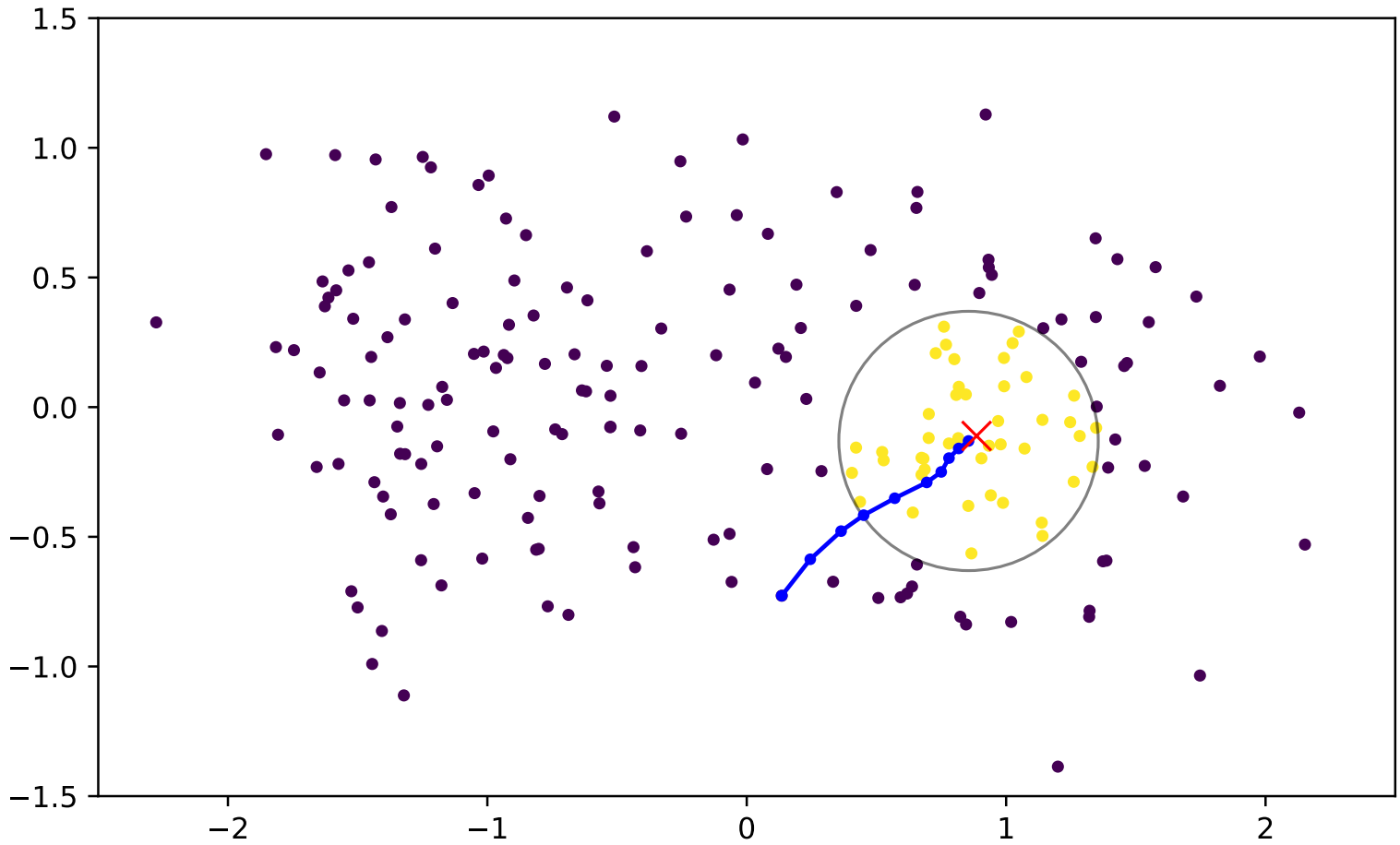
Convergence – Uniform kernel



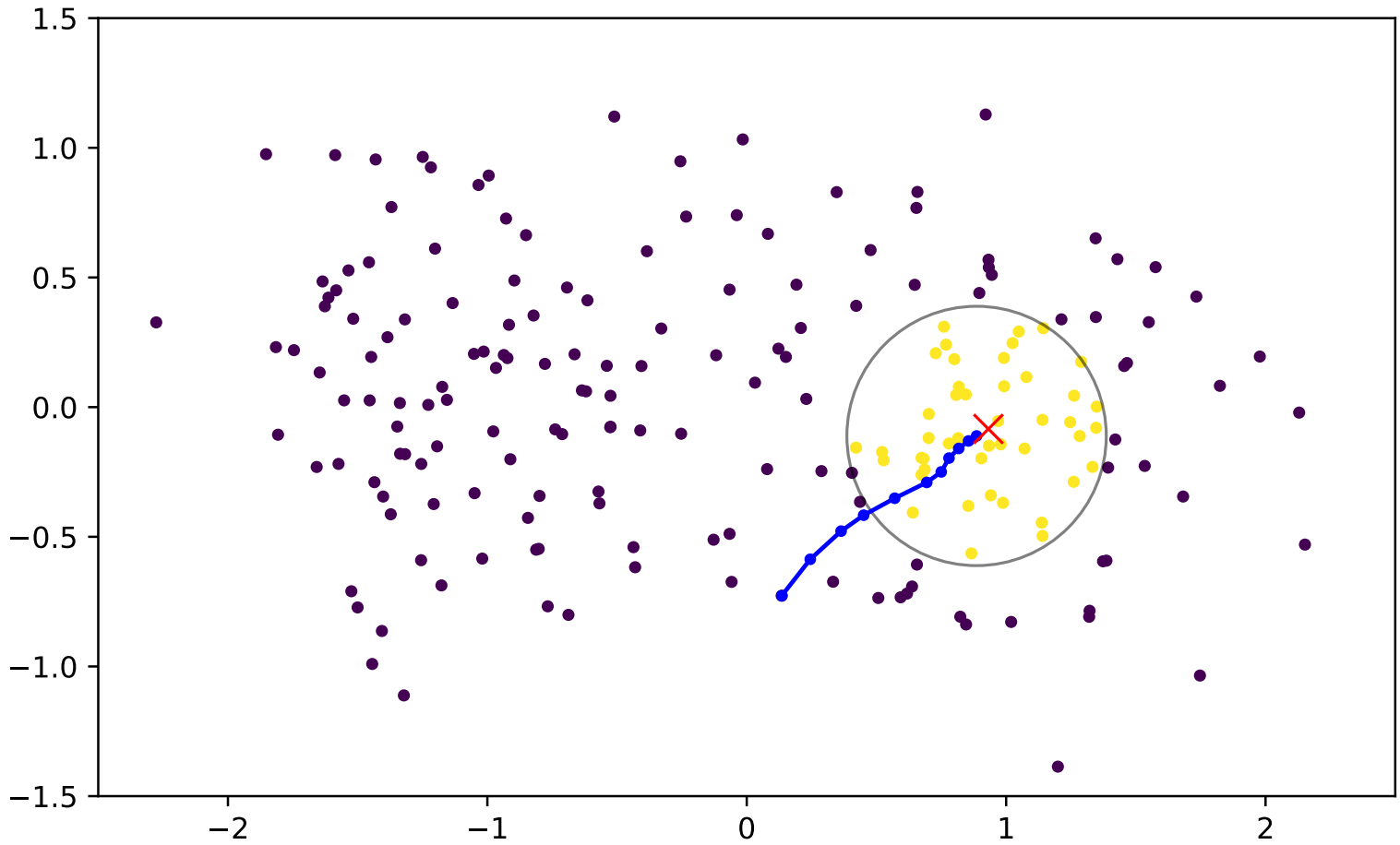
Convergence – Uniform kernel



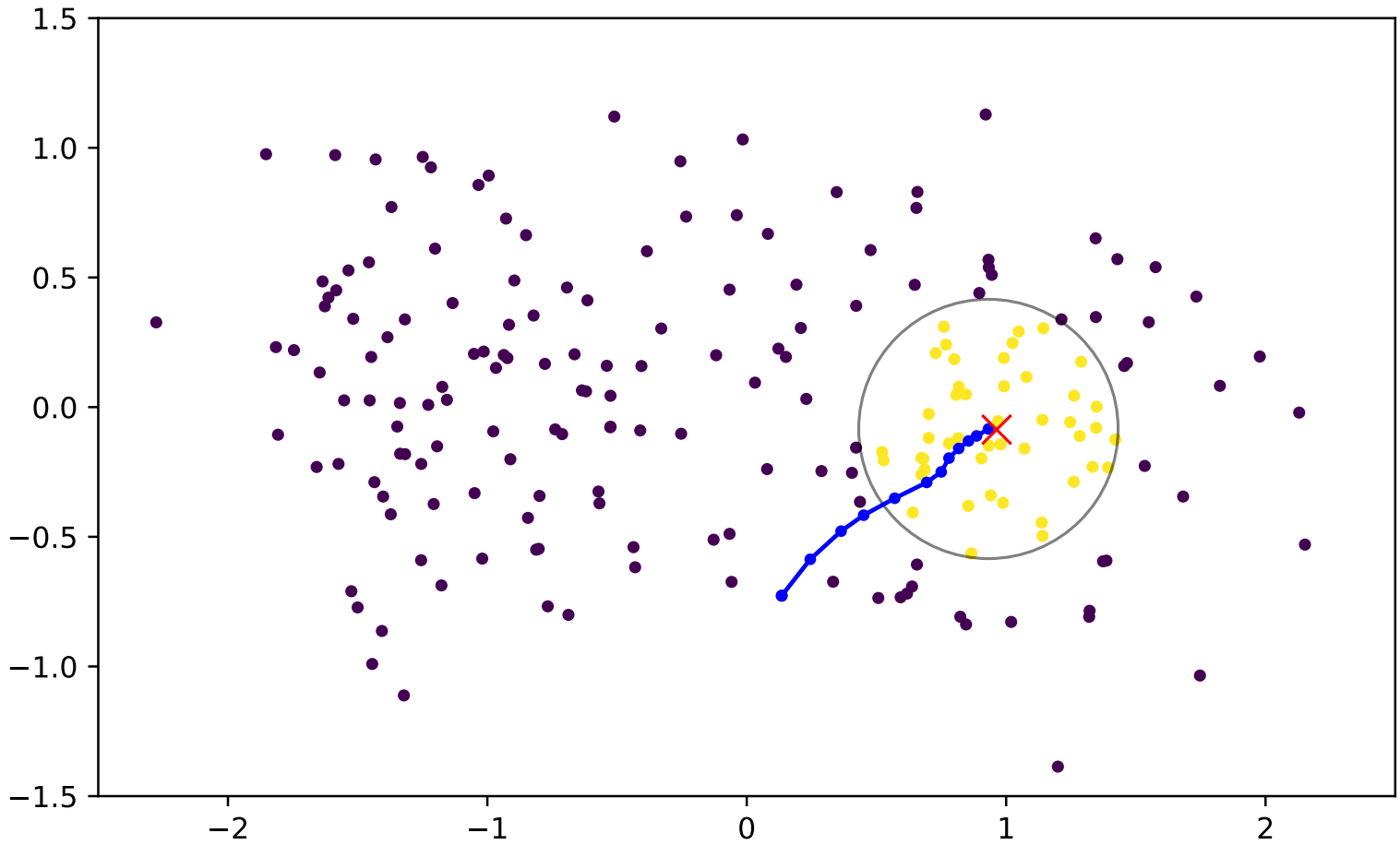
Convergence – Uniform kernel



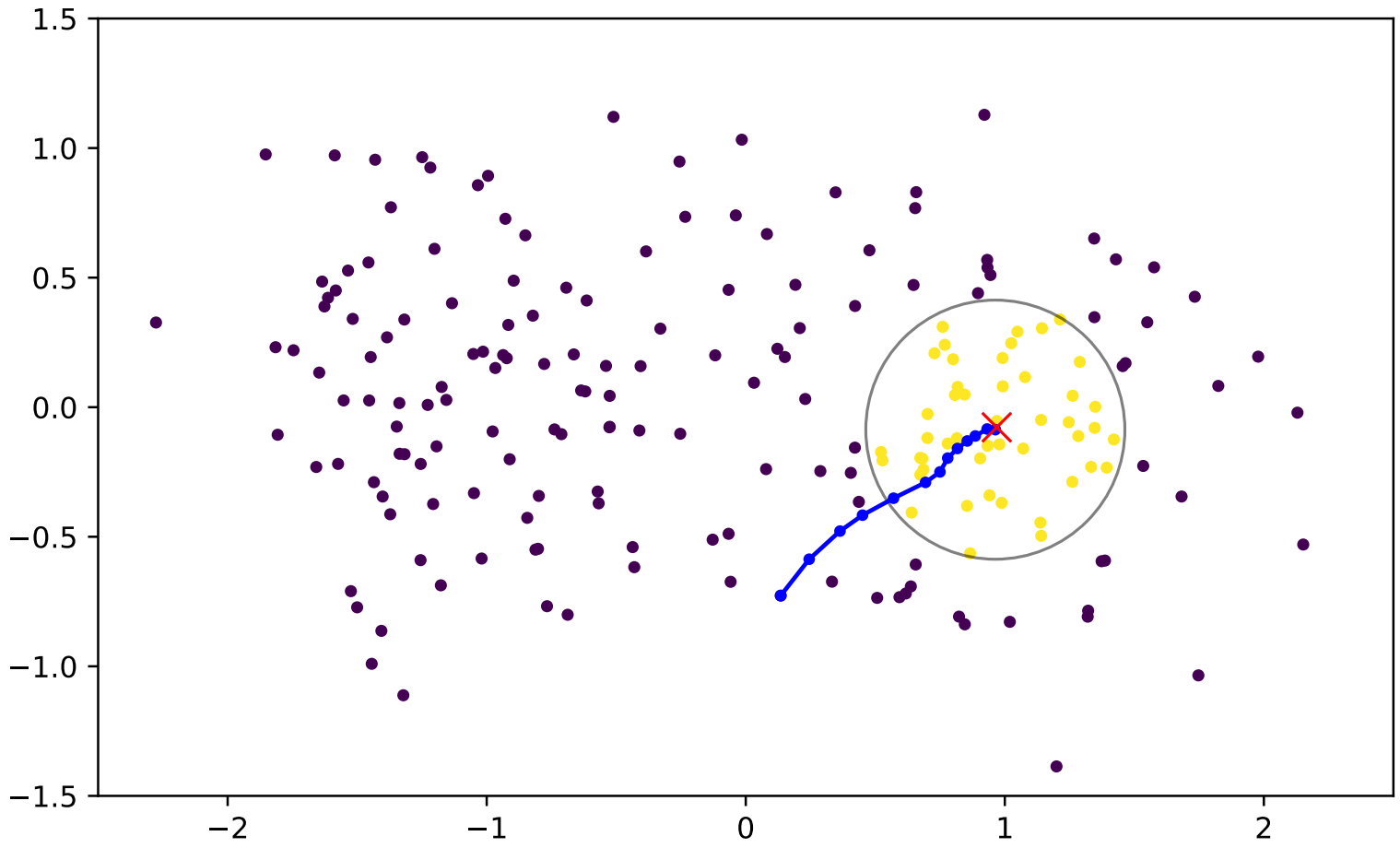
Convergence – Uniform kernel



Convergence – Uniform kernel



Convergence – Uniform kernel



Connecting the mean shift vector and kernel density estimation

- **Result**: the mean shift vector $\boldsymbol{m}(\boldsymbol{x})$ points into the gradient direction of a kernel density estimate.

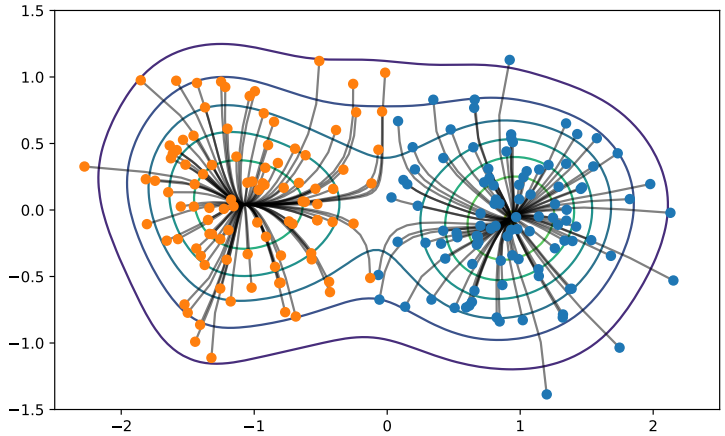
$$\boldsymbol{m}(\boldsymbol{x}) = \boldsymbol{\mu}^*(\boldsymbol{x}) - \boldsymbol{x} = \frac{h^2 c_{g,d}}{2 c_{k,d}} \frac{\nabla \hat{f}_{h,K}(\boldsymbol{x})}{\hat{f}_{h,G}(\boldsymbol{x})}$$

Connecting the mean shift vector and kernel density estimation

$$
\begin{aligned}
\nabla \hat{f}_{h,K}(\boldsymbol{x}) &= \nabla \frac{c_{k,d}}{nh^d} \sum_{i=1}^{n} k\left(\left\|\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h}\right\|^2\right) \\
&= \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^{n} (\boldsymbol{x}-\boldsymbol{x}_i) k'\left(\left\|\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h}\right\|^2\right) \\
&= \frac{2c_{k,d}}{nh^{d+2}} \left[\sum_{i=1}^{n} g\left(\left\|\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h}\right\|^2\right)\right] \left[\frac{\sum_{i=1}^{n} \boldsymbol{x}_i g\left(\left\|\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h}\right\|^2\right)} - \boldsymbol{x}\right] \\
&= \frac{2c_{k,d}}{h^2 c_{g,d}} \hat{f}_{h,G}(\boldsymbol{x}) \boldsymbol{m}(\boldsymbol{x})
\end{aligned}
$$

With $g(u) = -k'(u)$ and $G(\boldsymbol{x}) = c_{g,d} g(\|\boldsymbol{x}\|^2)$.

Connecting the mean shift vector and kernel density estimation



$$\boldsymbol{m}(\boldsymbol{x}) = \frac{h^2 c_{g,d}}{2c_{k,d}} \frac{\nabla \hat{f}_{h,K}(\boldsymbol{x})}{\hat{f}_{h,G}(\boldsymbol{x})}$$

- For the gaussian kernel $k(u) \propto -k'(u) = g(u)$ and therefore $K = G$.

Connecting the mean shift vector and kernel density estimation

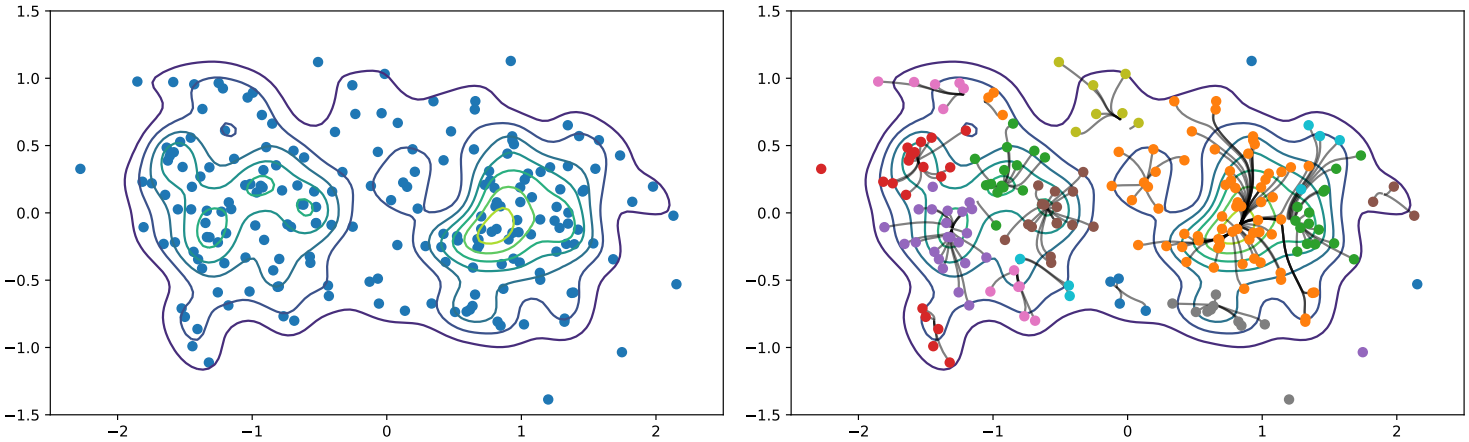
Name	$k(u)$	$-k'(u)$
Uniform	1	0
Epanechnikov	$1-u$	1
Biweight	$(1-u)^2$	$2(1-u)$
Triweight	$(1-u)^3$	$3(1-u)^2$
Gaussian	$\exp\left(-\frac{1}{2}u\right)$	$\frac{1}{2}\exp\left(-\frac{1}{2}u\right)$

- A kernel K for which $-k'(u) = g(u)$, is called a **shadow** of G.
- Calculating the mean shift vector $\boldsymbol{m}(\boldsymbol{x})$ with G does calculate the gradient direction of the kernel density estimate with K as kernel.

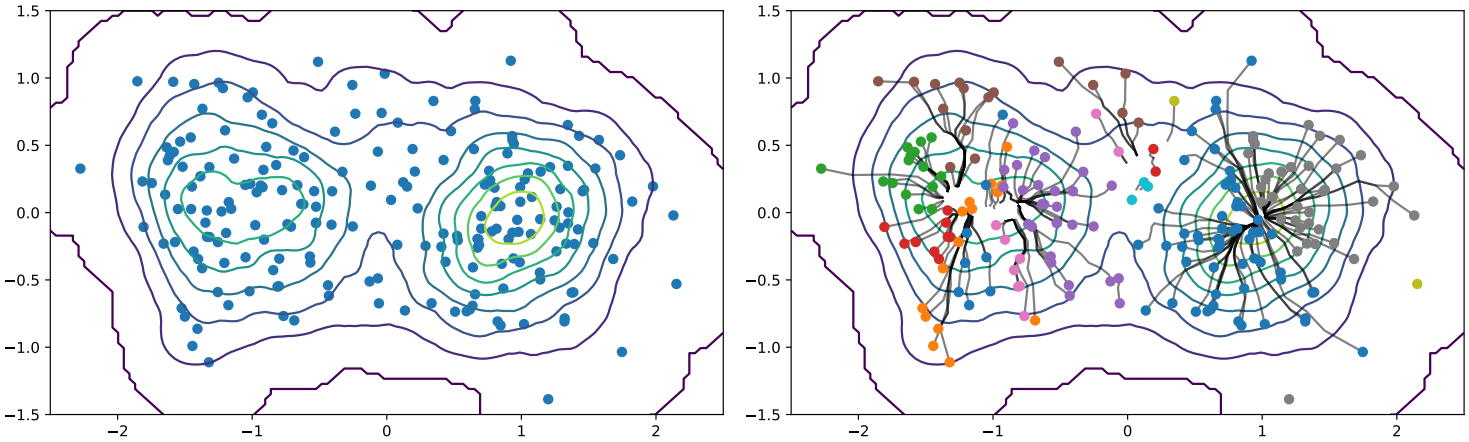
Bandwidth effects

- The choice of bandwidth influences the density estimation and therefore the clustering outcome
- Small bandwidth $\Rightarrow$ many density peaks / clusters
- Large bandwidth $\Rightarrow$ few peaks / clusters

Small bandwidth

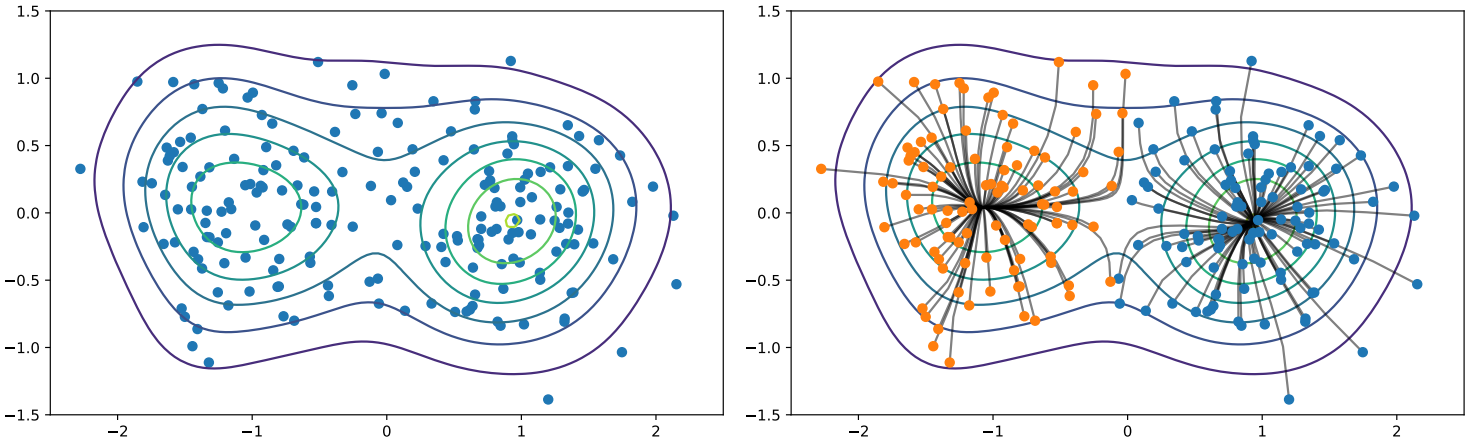


$h = 0.15$, gaussian kernel

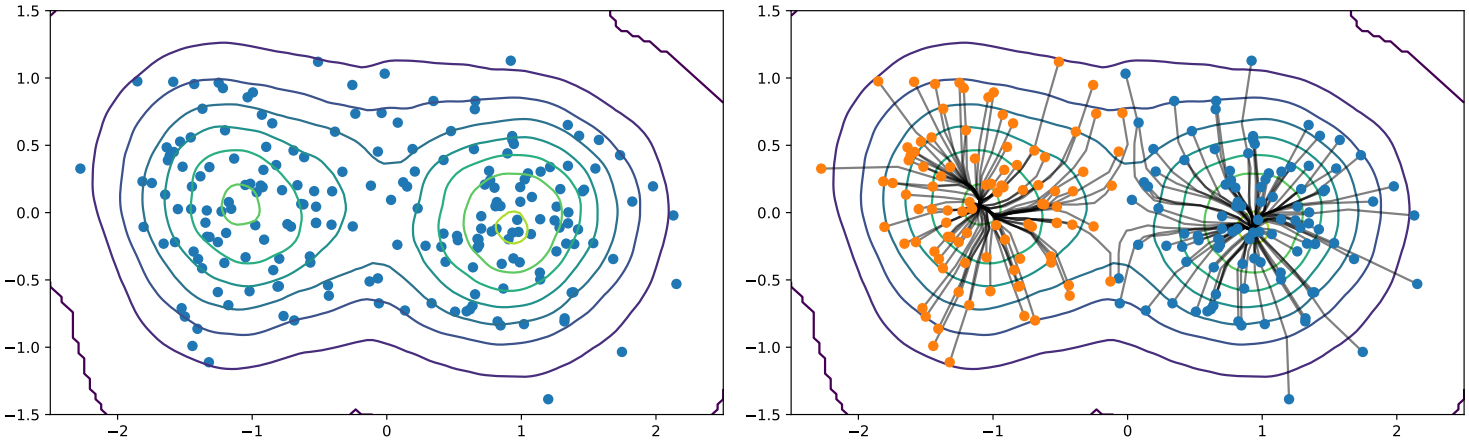


$h = 0.5$, uniform kernel

Medium bandwidth

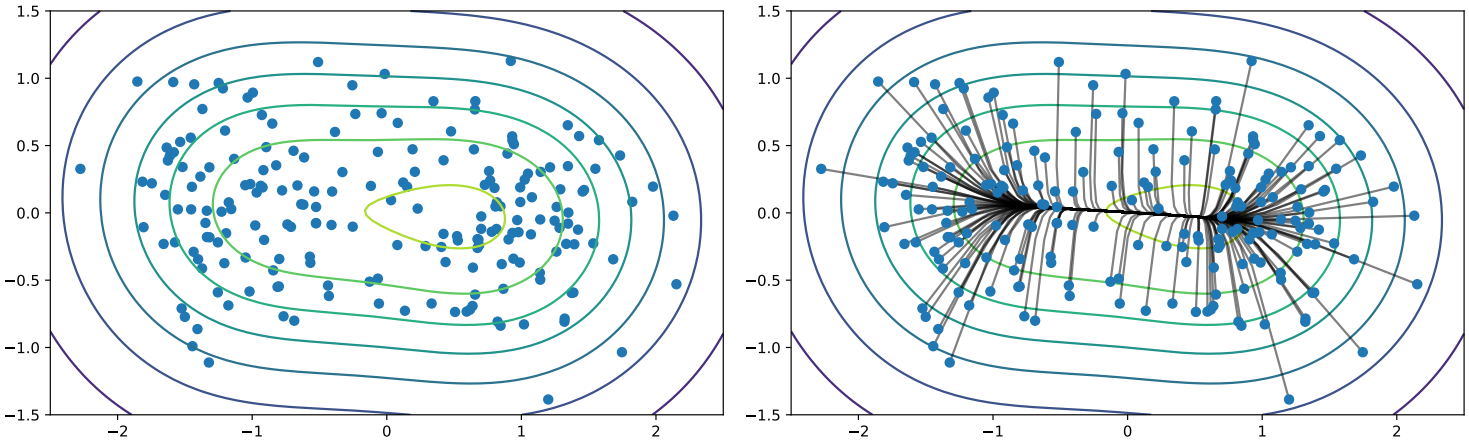


$h = 0.35$, gaussian kernel

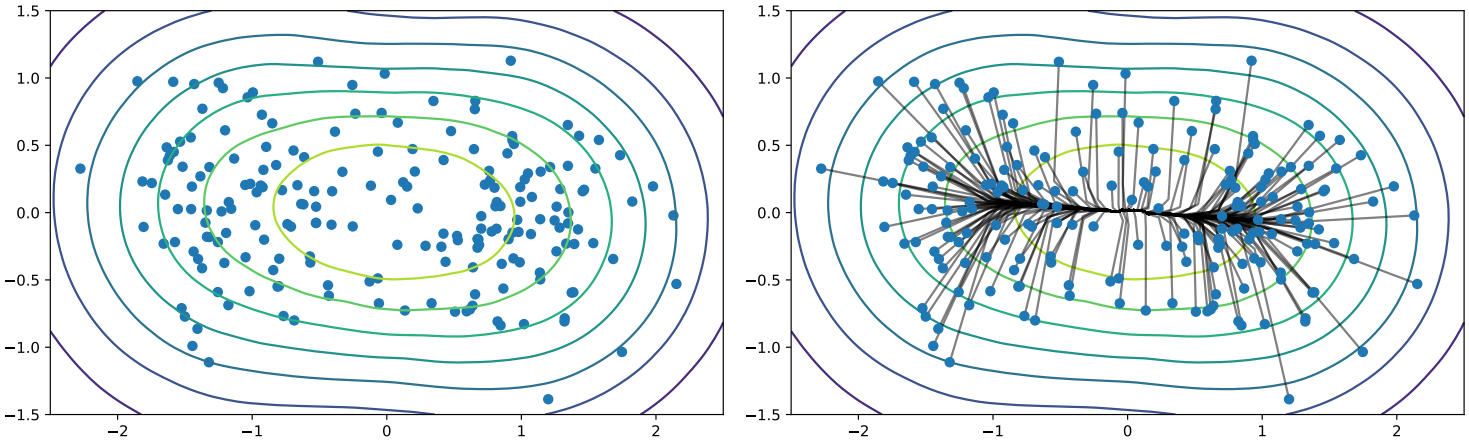


$h = 0.8$, uniform kernel

Large bandwidth



$h = 0.8$, gaussian kernel



$h = 1.7$, uniform kernel

Speedup methods

The mean shift algorithm has computational complexity $\mathcal{O}(Tn^2)$, with T being the number of iterations until convergence.

- **Parallelization** of iteration procedure. The trajectories are computational independent.
- **Sampling** of the dataset. Only perform it on a subset.
- **Bucketing** the dataset. Reduce number of possible positions.
- Use of **spatial data structures**. For kernels with compact support only the data points in the neighborhood are relevant.
- **Merging** of trajectories. When two trajectories get close to each other it is likely that they converge to the same point.

Discussion

Advantages

- No prior assumption on cluster shapes. Complex and non-convex shapes are possible.
- Only has one tuning parameter, the bandwidth.
- No restriction on number of clusters.
- Outliers do not affect the clustering.

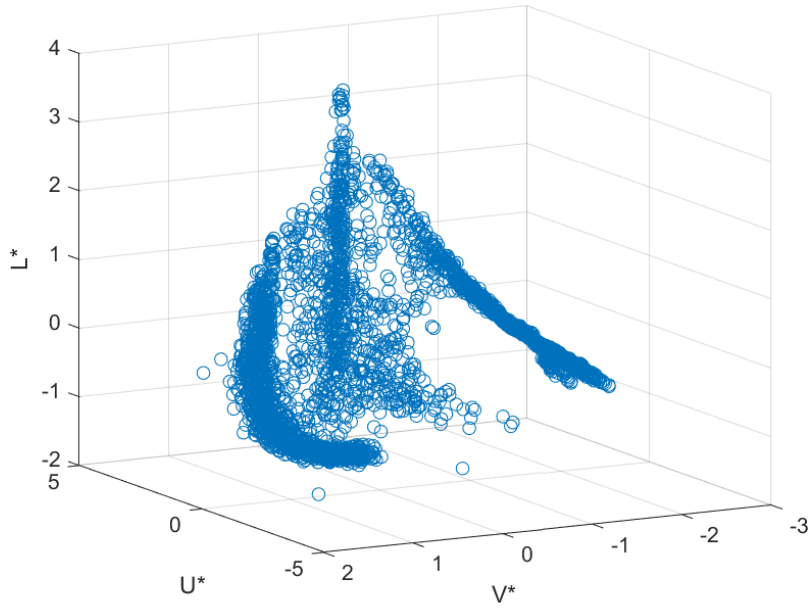
Disadvantages

- Density estimation fails for high dimensions ($\approx d > 5$).
- Bad computational complexity $\mathcal{O}(Tn^2)$.
- Finding a good bandwidth is hard.

Application

- Introduced by Fukunaga & Hostetler (1975), but popularized for computer vision tasks Comaniciu & Meer (2002) and Comaniciu et al. (2003).
- Generic clustering algorithm.
- Image segmentation, image filtering and object tracking.

Application – Image segmentation



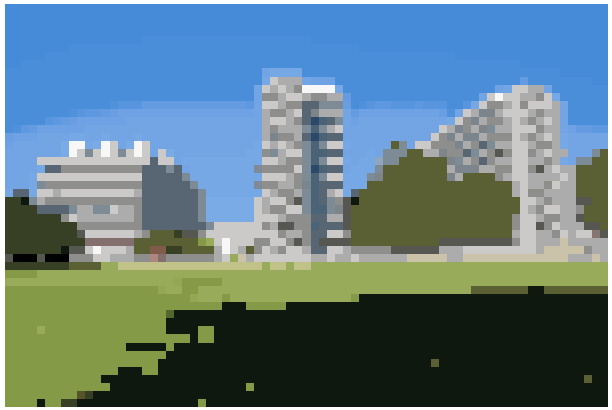
- Image data can be represented as data points. The pixels can be clustered and the clustering results in a segmentation of the original image.

Preprocessing

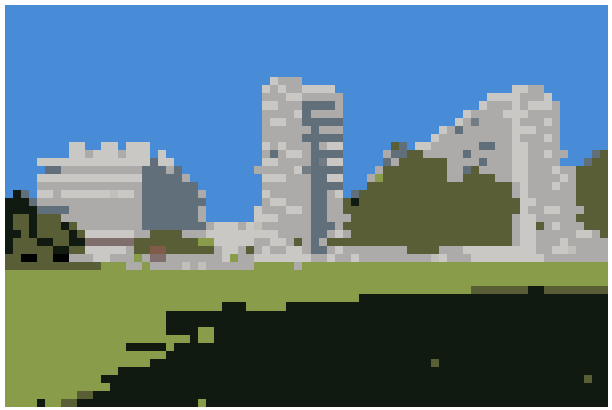
- **Image rescaling** for quicker convergence.
- **Color space transformation**. The CIE 1976 L*,u*,v* color space is built to approximate a perceptually uniform color space.
- **Feature standardization** with mean and standard deviation.
- **Adding spatial features**. Adding (x, y) pixel coordinates to pixel data.

Image segmentation – Gaussian kernel

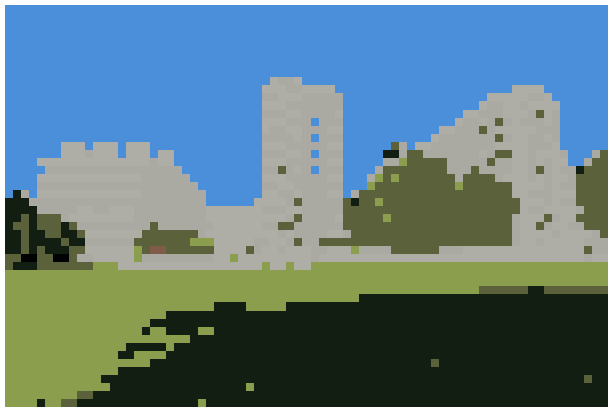
(a) $h = 0.1$



(b) $h = 0.2$

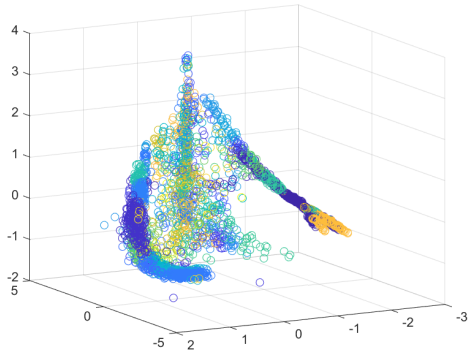


(c) $h = 0.3$

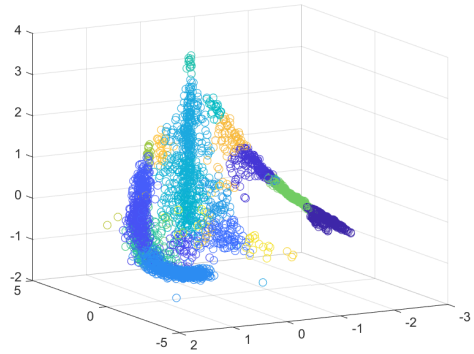


(d) $h = 0.4$

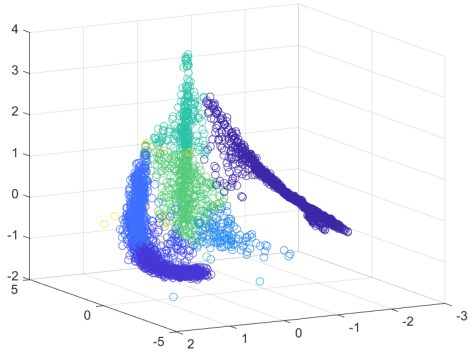
Image segmentation – Gaussian kernel



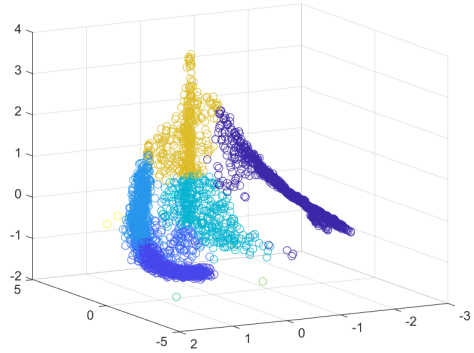
(a) $h = 0.1$



(b) $h = 0.2$



(c) $h = 0.3$

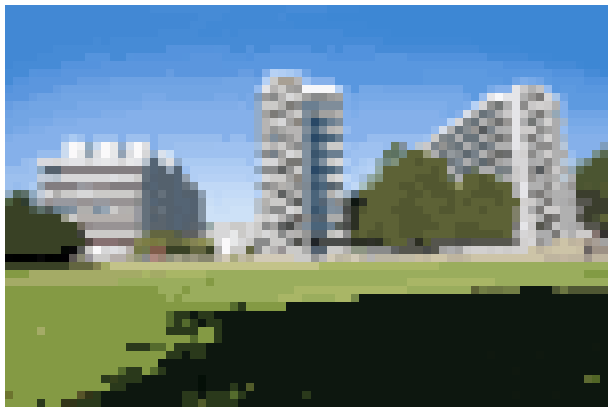


(d) $h = 0.4$

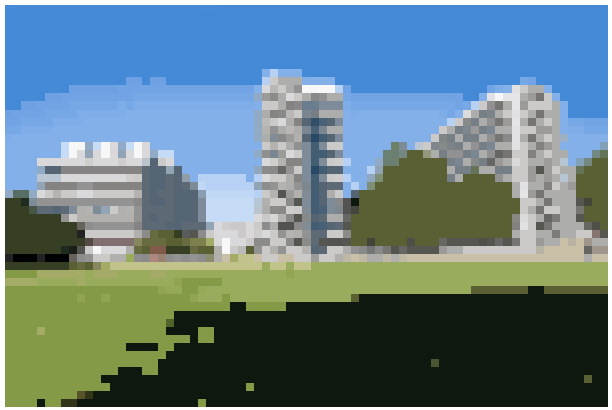
Image segmentation – Uniform kernel

(a) $h = 0.1$

(b) $h = 0.2$

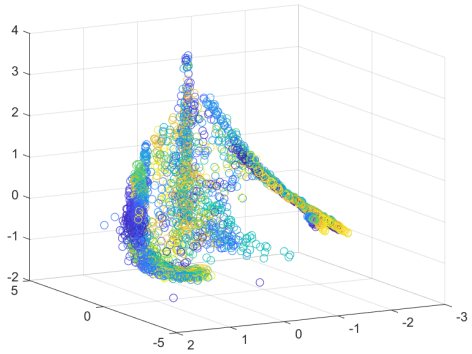


(c) $h = 0.3$

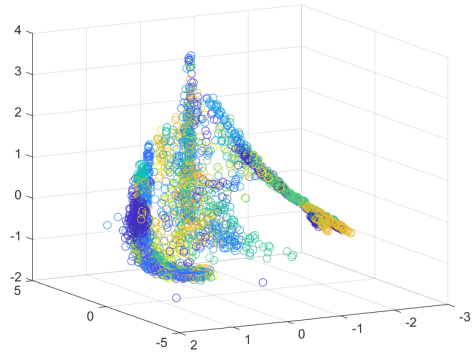


(d) $h = 0.4$

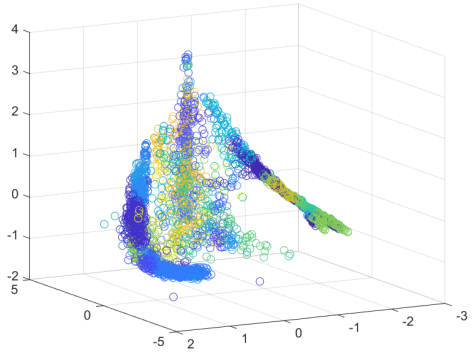
Image segmentation – Uniform kernel



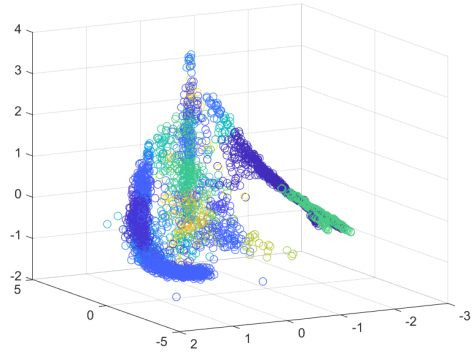
(a) $h = 0.1$



(b) $h = 0.2$

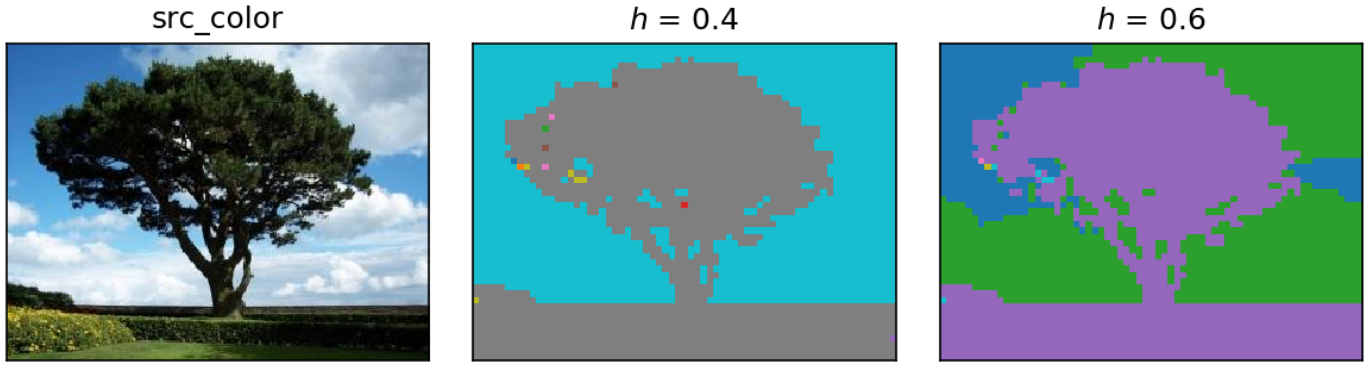


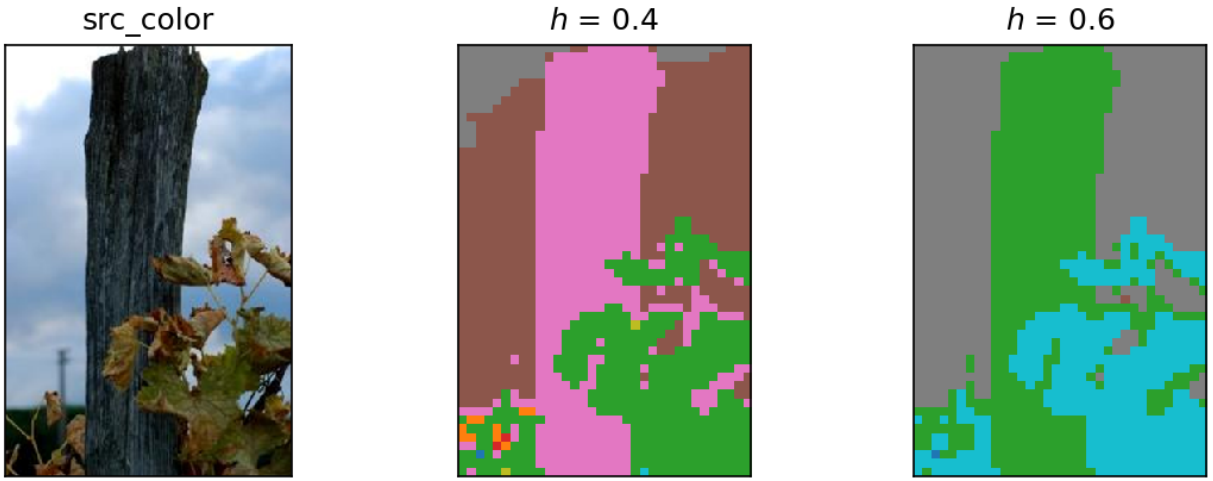
(c) $h = 0.3$



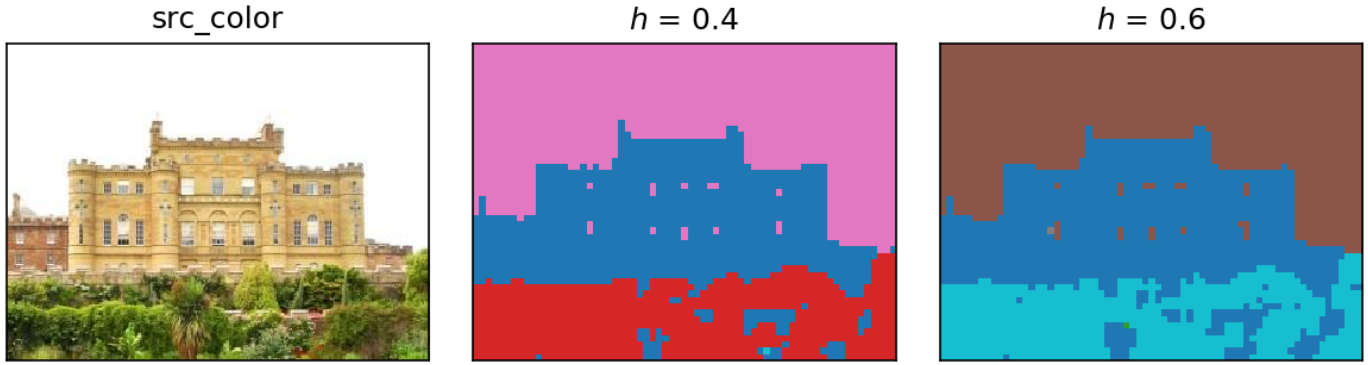
(d) $h = 0.4$

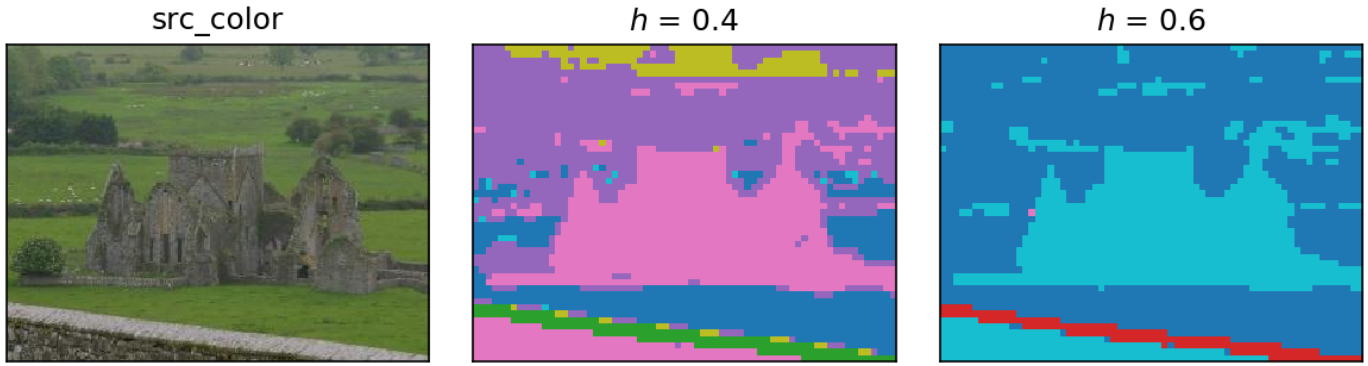
Results



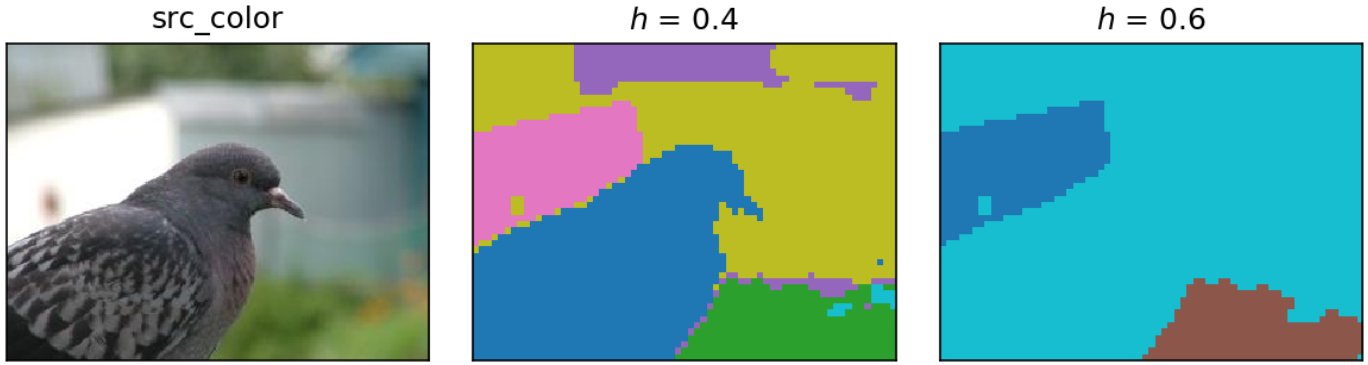


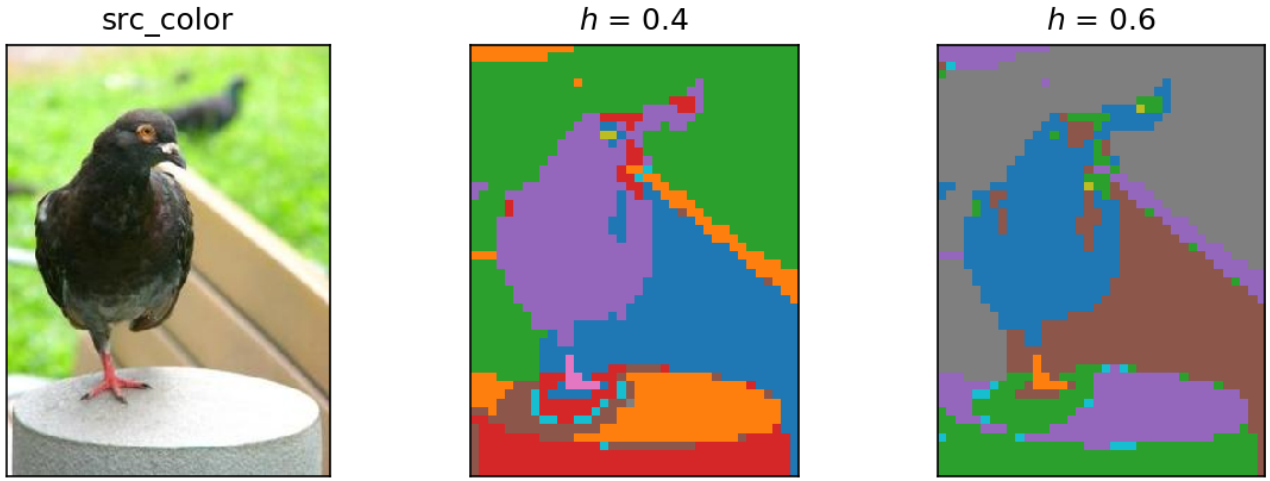
Results



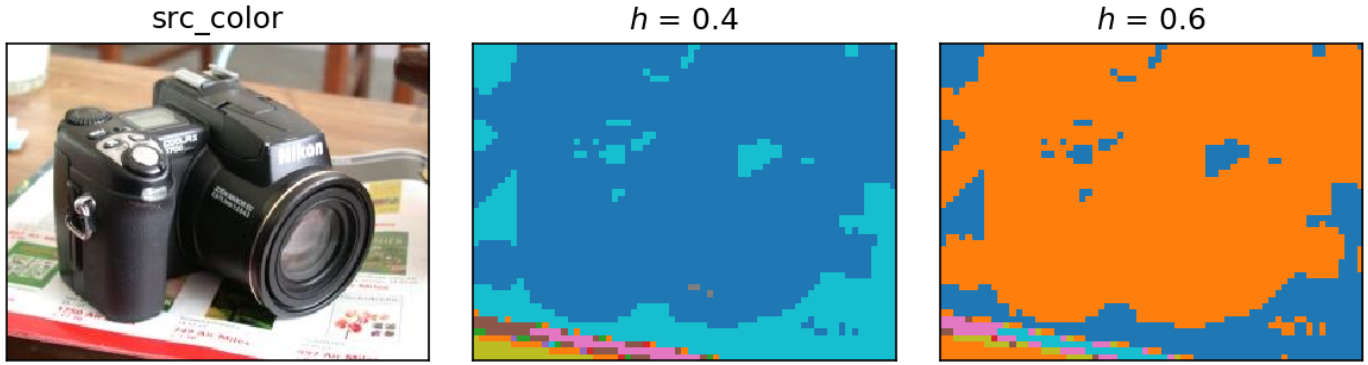


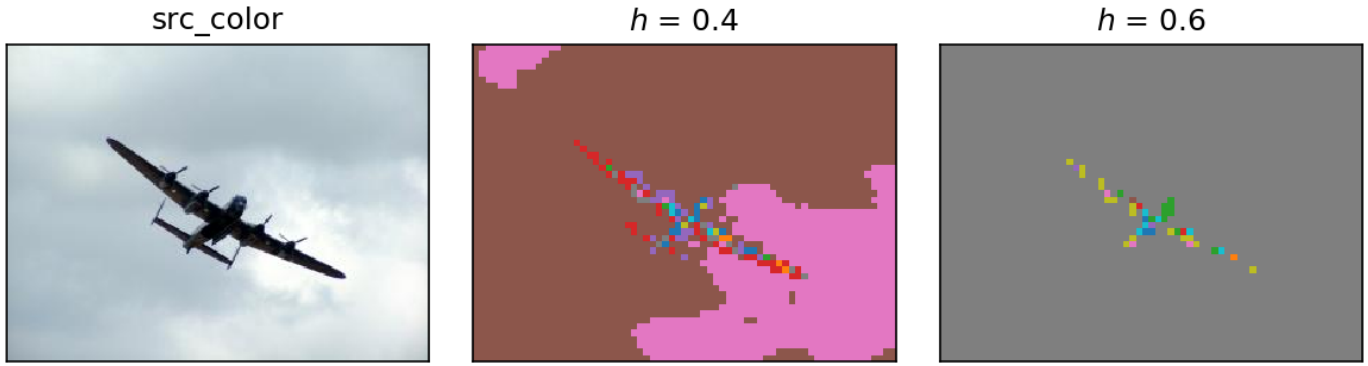
Results



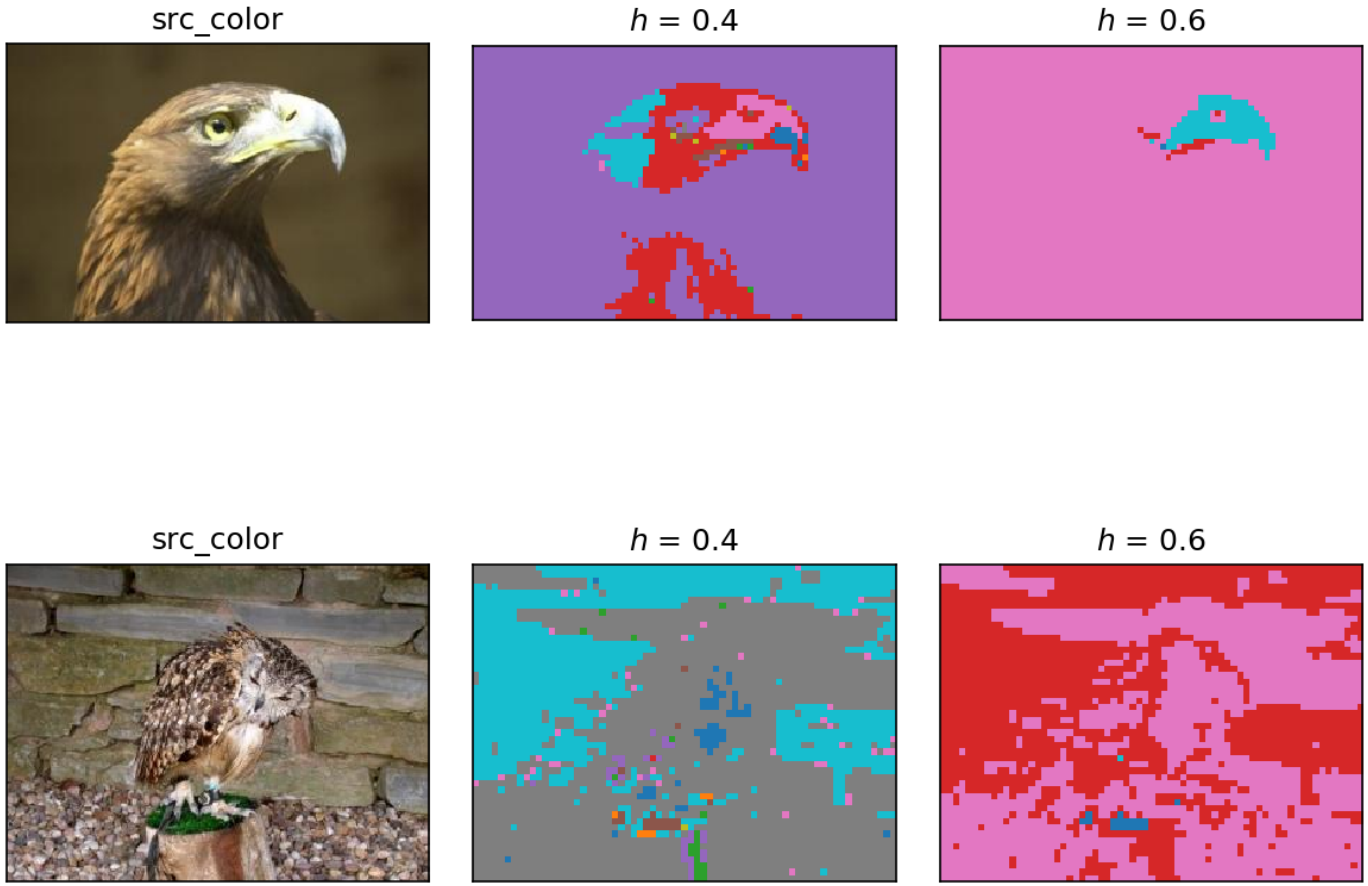


Limitations





Limitations



References I

Alpert, S., Galun, M., Brandt, A. & Basri, R. (2012), 'Image segmentation by probabilistic bottom-up aggregation and cue integration', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **34**(2), 315–326.

Comaniciu, D. & Meer, P. (2002), 'Mean shift: a robust approach toward feature space analysis', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(5), 603–619.

Comaniciu, D., Ramesh, V. & Meer, P. (2003), 'Kernel-based object tracking', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25**(5), 564–577.

Fukunaga, K. & Hostetler, L. (1975), 'The estimation of the gradient of a density function, with applications in pattern recognition', *IEEE Transactions on Information Theory* **21**(1), 32–40.

Figures

- “KIT chemical faculty building” picture. Copyright by KIT.
- Segmentation pictures. “segmentation evaluation database” from Alpert et al. (2012).
- All other illustrations were done by the author.