

# Mean shift

**Mean shift** is a non-parametric feature-space analysis technique for locating the maxima of a density function, a so-called mode-seeking algorithm.<sup>[1]</sup> Application domains include cluster analysis in computer vision and image processing.<sup>[2]</sup>

## Contents

**History**

**Overview**

**Details**

**Types of kernels**

**Applications**

Clustering

Tracking

Smoothing

**Strengths**

**Weaknesses**

**Availability**

**See also**

**References**

## History

The mean shift procedure was originally presented in 1975 by Fukunaga and Hostetler.<sup>[3]</sup>

## Overview

Mean shift is a procedure for locating the maxima—the modes—of a density function given discrete data sampled from that function.<sup>[1]</sup> This is an iterative method, and we start with an initial estimate ***x***. Let a kernel function ***K***(***x<sub>i</sub>*** − ***x***) be given. This function determines the weight of nearby points for re-estimation of the mean. Typically a Gaussian kernel on the distance to the current estimate is used, ***K***(***x<sub>i</sub>*** − ***x***) = *e*<sup>−*c*||***x<sub>i</sub>*** − ***x***||<sup>2</sup></sup>. The weighted mean of the density in the window determined by ***K*** is

$$\mathbf{m}(\mathbf{x}) = \frac{\sum_{\mathbf{x}_i \in N(\mathbf{x})} K(\mathbf{x}_i - \mathbf{x}) \mathbf{x}_i}{\sum_{\mathbf{x}_i \in N(\mathbf{x})} K(\mathbf{x}_i - \mathbf{x})}$$

where ***N***(***x***) is the neighborhood of ***x***, a set of points for which ***K***(***x<sub>i</sub>***) ≠ 0.

The difference ***m***(***x***) − ***x*** is called *mean shift* in Fukunaga and Hostetler.<sup>[3]</sup> The *mean-shift algorithm* now sets ***x*** ← ***m***(***x***), and repeats the estimation until ***m***(***x***) converges.

Although the mean shift algorithm has been widely used in many applications, a rigid proof for the convergence of the algorithm using a general kernel in a high dimensional space is still not known.<sup>[4]</sup> Aliyari Ghassabeh showed the convergence of the mean shift algorithm in one-dimension with a differentiable, convex, and strictly decreasing profile function.<sup>[5]</sup> However, the one-

dimensional case has limited real world applications. Also, the convergence of the algorithm in higher dimensions with a finite number of the (or isolated) stationary points has been proved.<sup>[4][6]</sup> However, sufficient conditions for a general kernel function to have finite (or isolated) stationary points have not been provided.

## Details

---

Let data be a finite set  $\mathcal{S}$  embedded in the n-dimensional Euclidean space,  $X$ . Let  $K$  be a flat kernel that is the characteristic function of the  $\lambda$ -ball in  $X$ ,

$$K(\mathbf{x}) = \begin{cases} 1 & \text{if } \|\mathbf{x}\| \leq \lambda \\ 0 & \text{if } \|\mathbf{x}\| > \lambda \end{cases}$$

In each iteration of the algorithm,  $\mathbf{s} \leftarrow \mathbf{m}(\mathbf{s})$  is performed for all  $\mathbf{s} \in \mathcal{S}$  simultaneously. The first question, then, is how to estimate the density function given a sparse set of samples. One of the simplest approaches is to just smooth the data, e.g., by convolving it with a fixed kernel of width  $h$ ,

$$f(\mathbf{x}) = \sum_i K(\mathbf{x} - \mathbf{x}_i) = \sum_i k\left(\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{h^2}\right)$$

where  $\mathbf{x}_i$  are the input samples and  $k(r)$  is the kernel function (or *Parzen window*).  $h$  is the only parameter in the algorithm and is called the bandwidth. This approach is known as *kernel density estimation* or the Parzen window technique. Once we have computed  $f(\mathbf{x})$  from equation above, we can find its local maxima using gradient ascent or some other optimization technique. The problem with this "brute force" approach is that, for higher dimensions, it becomes computationally prohibitive to evaluate  $f(\mathbf{x})$  over the complete search space. Instead, mean shift uses a variant of what is known in the optimization literature as *multiple restart gradient descent*. Starting at some guess for a local maximum,  $\mathbf{y}_k$ , which can be a random input data point  $\mathbf{x}_1$ , mean shift computes the gradient of the density estimate  $f(\mathbf{x})$  at  $\mathbf{y}_k$  and takes an uphill step in that direction.<sup>[7]</sup>

## Types of kernels

---

**Kernel definition:** Let  $X$  be the n-dimensional Euclidean space,  $\mathbf{R}^n$ . Denote the  $i$ th component of  $\mathbf{x}$  by  $\mathbf{x}_i$ . The norm of  $\mathbf{x}$  is a non-negative number,  $\|\mathbf{x}\|^2 = \mathbf{x}^T \mathbf{x}$ . A function  $K: \mathbf{X} \rightarrow \mathbf{R}$  is said to be a kernel if there exists a *profile*,  $k: [0, \infty] \rightarrow \mathbf{R}$ , such that

$$K(\mathbf{x}) = k(\|\mathbf{x}\|^2) \text{ and}$$

- $k$  is non-negative.
- $k$  is nonincreasing:  $k(a) \geq k(b)$  if  $a < b$ .
- $k$  is piecewise continuous and  $\int_0^\infty k(r) dr < \infty$

The two frequently used kernel profiles for mean shift are:

### Flat kernel

$$k(x) = \begin{cases} 1 & \text{if } x \leq \lambda \\ 0 & \text{if } x > \lambda \end{cases}$$

### Gaussian kernel

$$k(x) = e^{-\frac{x}{2\sigma^2}},$$

where the standard deviation parameter  $\sigma$  works as the bandwidth parameter,  $h$ .

# Applications

---

## Clustering

Consider a set of points in two-dimensional space. Assume a circular window centered at  $C$  and having radius  $r$  as the kernel. Mean shift is a hill climbing algorithm which involves shifting this kernel iteratively to a higher density region until convergence. Every shift is defined by a mean shift vector. The mean shift vector always points toward the direction of the maximum increase in the density. At every iteration the kernel is shifted to the centroid or the mean of the points within it. The method of calculating this mean depends on the choice of the kernel. In this case if a Gaussian kernel is chosen instead of a flat kernel, then every point will first be assigned a weight which will decay exponentially as the distance from the kernel's center increases. At convergence, there will be no direction at which a shift can accommodate more points inside the kernel.

## Tracking

The mean shift algorithm can be used for visual tracking. The simplest such algorithm would create a confidence map in the new image based on the color histogram of the object in the previous image, and use mean shift to find the peak of a confidence map near the object's old position. The confidence map is a probability density function on the new image, assigning each pixel of the new image a probability, which is the probability of the pixel color occurring in the object in the previous image. A few algorithms, such as kernel-based object tracking,<sup>[8]</sup> ensemble tracking,<sup>[9]</sup> CAMshift<sup>[10][11]</sup> expand on this idea.

## Smoothing

Let  $\mathbf{x}_i$  and  $\mathbf{z}_i, i = 1, \dots, n$ , be the  $d$ -dimensional input and filtered image pixels in the joint spatial-range domain. For each pixel,

- Initialize  $j = 1$  and  $\mathbf{y}_{i,1} = \mathbf{x}_i$
- Compute  $\mathbf{y}_{i,j+1}$  according to  $\mathbf{m}(\cdot)$  until convergence,  $\mathbf{y} = \mathbf{y}_{i,c}$ .
- Assign  $\mathbf{z}_i = (\mathbf{x}_i^s, \mathbf{y}_{i,c}^r)$ . The superscripts  $s$  and  $r$  denote the spatial and range components of a vector, respectively. The assignment specifies that the filtered data at the spatial location axis will have the range component of the point of convergence  $\mathbf{y}_{i,c}^r$ .

## Strengths

---

1. Mean shift is an application-independent tool suitable for real data analysis.
2. Does not assume any predefined shape on data clusters.
3. It is capable of handling arbitrary feature spaces.
4. The procedure relies on choice of a single parameter: bandwidth.
5. The bandwidth/window size 'h' has a physical meaning, unlike k-means.

## Weaknesses

---

1. The selection of a window size is not trivial.
2. Inappropriate window size can cause modes to be merged, or generate additional "shallow" modes.
3. Often requires using adaptive window size.

## Availability

---

Variants of the algorithm can be found in machine learning and image processing packages:

- [ELKI](#). Java data mining tool with many clustering algorithms.
- [ImageJ](#). Image filtering using the mean shift filter.
- [OpenCV](#) contains mean-shift implementation via `cvMeanShift` Method
- [Orfeo toolbox](#). A C++ implementation.
- [Scikit-learn](#) Numpy/Python implementation uses ball tree for efficient neighboring points lookup

## See also

---

- [DBSCAN](#)
- [OPTICS algorithm](#)
- [Kernel density estimation \(KDE\)](#)
- [Kernel \(statistics\)](#)

## References

---

1. Cheng, Yizong (August 1995). "Mean Shift, Mode Seeking, and Clustering". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **17** (8): 790–799. [CiteSeerX 10.1.1.510.1222](#) (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.510.1222>). [doi:10.1109/34.400568](#) (<https://doi.org/10.1109%2F34.400568>).
2. Comaniciu, Dorin; Peter Meer (May 2002). "Mean Shift: A Robust Approach Toward Feature Space Analysis". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **24** (5): 603–619. [CiteSeerX 10.1.1.160.3832](#) (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.160.3832>). [doi:10.1109/34.1000236](#) (<https://doi.org/10.1109%2F34.1000236>).
3. Fukunaga, Keinosuke; Larry D. Hostetler (January 1975). "The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition". *IEEE Transactions on Information Theory*. **21** (1): 32–40. [doi:10.1109/TIT.1975.1055330](#) (<https://doi.org/10.1109%2FTIT.1975.1055330>).
4. Aliyari Ghassabeh, Youness (2015-03-01). "A sufficient condition for the convergence of the mean shift algorithm with Gaussian kernel". *Journal of Multivariate Analysis*. **135**: 1–10. [doi:10.1016/j.jmva.2014.11.009](#) (<https://doi.org/10.1016%2Fj.jmva.2014.11.009>).
5. Aliyari Ghassabeh, Youness (2013-09-01). "On the convergence of the mean shift algorithm in the one-dimensional space". *Pattern Recognition Letters*. **34** (12): 1423–1427. [arXiv:1407.2961](#) (<https://arxiv.org/abs/1407.2961>). [doi:10.1016/j.patrec.2013.05.004](#) (<https://doi.org/10.1016%2Fj.patrec.2013.05.004>).
6. Li, Xiangru; Hu, Zhanyi; Wu, Fuchao (2007-06-01). "A note on the convergence of the mean shift". *Pattern Recognition*. **40** (6): 1756–1762. [doi:10.1016/j.patcog.2006.10.016](#) (<https://doi.org/10.1016%2Fj.patcog.2006.10.016>).
7. Richard Szeliski, *Computer Vision, Algorithms and Applications*, Springer, 2011
8. Comaniciu, Dorin; Visvanathan Ramesh; Peter Meer (May 2003). "Kernel-based Object Tracking". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **25** (5): 564–575. [CiteSeerX 10.1.1.8.7474](#) (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.8.7474>). [doi:10.1109/tpami.2003.1195991](#) (<https://doi.org/10.1109%2Ftpami.2003.1195991>).
9. Avidan, Shai (2005). *Ensemble Tracking*. 2005 *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. **2**. San Diego, California: IEEE. pp. 494–501. [doi:10.1109/CVPR.2005.144](#) (<https://doi.org/10.1109%2FCVPR.2005.144>). ISBN 978-0-7695-2372-9.
10. Gary Bradski (1998) *Computer Vision Face Tracking For Use in a Perceptual User Interface* (<http://download.intel.com/technology/itj/q21998/pdf/camshift.pdf>) Archived (<https://web.archive.org/web/20120417121810/http://download.intel.com/technology/itj/q21998/pdf/camshift.pdf>) 2012-04-17 at the [Wayback Machine](#), Intel Technology Journal, No. Q2.
11. Emami, Ebrahim (2013). *Online failure detection and correction for CAMShift tracking algorithm* (<https://ieeexplore.ieee.org/abstract/document/6779974/>). 2013 *Iranian Conference on Machine Vision and Image Processing (MVIP)*. **2**. IEEE. pp. 180–183. [doi:10.1109/IranianMVIP.2013.6779974](#) (<https://doi.org/10.1109%2FIranianMVIP.2013.6779974>). ISBN 978-1-4673-6184-2.

---

Retrieved from "[https://en.wikipedia.org/w/index.php?title=Mean\\_shift&oldid=912471903](https://en.wikipedia.org/w/index.php?title=Mean_shift&oldid=912471903)"

---

**This page was last edited on 25 August 2019, at 20:48 (UTC).**

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.