Kernel Density Estimates and Mean-Shift Clustering

Jonas Spinner*– 1927895
Analytics and Statistics
KIT – Karlsruhe Institute of Technologie

January 10, 2019

*jonas.spinner@student.kit.edu

Contents

Abstract

Kernel density estimation is widely used for nonparametric data analysis and one of the main ideas behind the mean-shift algorithm. The mean-shift algorithm is a clustering algorithm with

1 Introduction

In this seminar paper I am going to discuss the method of kernel density estimation (KDE) and it's use in the mean-shift clustering algorithm (MSC). Clustering is one of the main tasks in Machine-Learning and MSC has many applications, mainly in image segmentation. The mean-shift algorithm is an non-parametric approach which allows a wide range of data distributions without imposing prior knowledge.

2 Kernel Density Estimation

The goal of density estimation is estimating the probality density of p, given samples $\{\boldsymbol{x}_i\}_{i=1}^n$ in $\mathbb{R}^d, \boldsymbol{x} \sim p(\boldsymbol{x})$.

Histograms estimate the probability distribution by bucketing the samples and counting the proportion of samples which fall in each bucket.

The *kernel density estimate* is defined as

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

2.1 Popular Kernels

A general Kernel is a function $K : \mathbb{R}^d \to \mathbb{R}$ satisfying the following properties. See (Comaniciu & Meer 2002) and (Wand & Jones 1995, p. 95).

$$\int_{\mathbb{R}^d} K(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = 1$$

Name	Formula	Support
Uniform	$K(\boldsymbol{x}) = \frac{1}{2}$	$\|\boldsymbol{x}\| \leq 1$
Triangluar	$K(\boldsymbol{x}) = 1 - \|\boldsymbol{x}\|$	$\|\boldsymbol{x}\| \leq 1$
Biweight		
Triweight		
Epanechnikov		

$$\int_{\mathbb{R}^d} \boldsymbol{x} K(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = \boldsymbol{0}$$

$$\lim_{\|\boldsymbol{x}\| \to \infty} \|\boldsymbol{x}\|^d K(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = 0$$

$$\int_{\mathbb{R}^d} \boldsymbol{x}\boldsymbol{x}^T K(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = c_K \boldsymbol{I}$$

$$K^P(\boldsymbol{x}) = \prod_{i=1}^{n} K_1(\boldsymbol{x}_i)$$

$$K^S(\boldsymbol{x}) = a_{k,d} K_1(\|\boldsymbol{x}\|)$$

2.2 Bandwidth Selection

One of the main challenges in using the kernel density estimator in practice is the choice of the bandwidth.

3 Mean Shift Clustering

Clustering is one of the main machine learning tasks, concerned with grouping objects $\{x_i\}_{i=1}^n$ into clusters C_j resulting in a clustering $\{C_j\}_{j=1}^K$.

3.1 Different clustering notions

There are many different notions of what a good clustering is and what a form a cluster takes. These different notions result in different clustering algorithmns. One class of clustering algorithms is centroid based clustering. Each cluster is represented by a representative called centroid. For example the representative of a K-Means clustering is the mean of the cluster $m_j := \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$.

Another way clustering algorithms can be differentiated is wheter of not the amount of clusters is given as an input to the algorithm or not. For example the K-Means algorithm searches for exactly K clusters. But there are also other notions of a cluster center or representative. One is the added restriction, that the center is itself a data object or atleast "looks like" one. The latter is covered by using representatives which are likely also a data object. That's the core of the mean shift clustering algorithm. It estimates the underlying density of the data objects and searches for points which high relative probability. Or in other words it identifies the modes of the estimated density.

That leaves open on how to estimate the density.

3.2 The algorithm

function MeanShift($\boldsymbol{x}_1, ..., \boldsymbol{x}_n : \mathbb{R}^d$)
  $\boldsymbol{Z} = \boldsymbol{X} : \mathbb{R}^{d \times n}$
  repeat
    $\boldsymbol{W} = (\exp(-\frac{1}{2}\|(\boldsymbol{x}_i - \boldsymbol{x}_j)/\sigma\|^2))_{i,j=1..n}$
    $\boldsymbol{D} = \operatorname{diag}(\sum_i \boldsymbol{W}_{ij})$
    $\boldsymbol{Q} = \boldsymbol{W}\boldsymbol{D}^{-1} : \mathbb{R}^{n \times n}$
    $\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{Q} : \mathbb{R}^{d \times n}$
  until stop
  return ConnectedComponents($\{\boldsymbol{z}_i\}_{i=1}^n, \varepsilon$)

function MeanShift($\boldsymbol{x}_1, ..., \boldsymbol{x}_n : \mathbb{R}^d$)
  for $i = 1..n$ **do**
    $\boldsymbol{x} = \boldsymbol{x}_i$
    repeat
      $\forall n : p(i \mid \boldsymbol{x}) = \frac{\exp(-\frac{1}{2}\|(\boldsymbol{x}-\boldsymbol{x}_i)/\sigma\|^2)}{\sum_{j=1}^n \exp(-\frac{1}{2}\|(\boldsymbol{x}-\boldsymbol{x}_j)/\sigma\|^2)}$
      $\boldsymbol{x} = \sum_{i=1}^n p(i \mid \boldsymbol{x})\boldsymbol{x}_i$
    until stop
    $\boldsymbol{z}_i = \boldsymbol{x}$
  return ConnectedComponents($\{\boldsymbol{z}_i\}_{i=1}^n, \varepsilon$)

3.3 Problems

4 Application

The main application for the mean shift clustering is image segmentation. Although the raw image data in the RGB-colorspace (red, green, blue) can be used, it is often not disereable. A human percieves distances of colors differently than the RGB-colorspace indicates. For that reason the image is often transformed in a more suitable colorspace, for example the L*u*v*-colorspace.

$$L*$$

Another aspect is the spatial coherence of the clusters in the image. Two pixels with the same color, but in widely different parts of the image would be put in the same cluster. That can be prevented by adding imagespace information to the pixels.

4.1 Experiments

4.2 Evaluation and Comparison

5 Summary

References

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.

Wand, M. P. & Jones, M. C. (1995), *Kernel Smoothing*, Vol. 60 of *Monographs on Statistics and Applied Probability*, Springer US, Boston, MA and s.l.
URL: *http://dx.doi.org/10.1007/978-1-4899-4493-1*

A Code

```
function [A, C] = mean_shift(X, kernel, epsilon)
    tol = 1e-3; max_iter = 1000;

    Z = X;
    for t = 1:max_iter
        W = apply_kernel(X, Z, kernel);
        D = diag(sum(W, 1));
        Q = W * D^(-1);
        Z_next = X * Q;

        % stop criteria
        if (max(abs(Z_next - Z), [], 'all') < tol)
            break
        end
    Z = Z_next;
    end

    [A, C] = connected_component(Z, epsilon);
end
```

B Notation

Data

$$n \quad \text{vs.} \quad N$$

$$x \quad \text{vs.} \quad \boldsymbol{x}$$

$$x_i \quad \text{vs.} \quad x^{(i)}$$

Kernels

$$\hat{f}(x) \quad \text{vs.} \quad \hat{p}(x)$$

$$\hat{f}_h(x) = \frac{1}{n} \sum_{i=1}^{n} K_h(x - x_i)$$

$$\hat{f}_h(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \qquad \text{vs.} \qquad \hat{f}_h(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

Radial Kernel

$$\hat{f}_h(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} K_h(\|\boldsymbol{x} - \boldsymbol{x}_i\|)$$

Product Kernel

$$\hat{f}_h(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n}\prod_{j=1}^{d} K_{h_j}(\boldsymbol{x}_j - \boldsymbol{x}_{ij}) \qquad \text{vs.} \qquad \hat{f}(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n}\prod_{j=1}^{d} K_{h_j}(\boldsymbol{x}_j - \boldsymbol{x}_j^{(i)})$$