Fast Data Reduction via KDE Approximation

Daniel Freedman and Pavel Kisilev
Hewlett-Packard Laboratories, Haifa, Israel
{daniel.freedman, pavel.kisilev}@hp.com

Introduction Many of today's real world applications need to handle and analyze continually growing amounts of data, while the cost of collecting data decreases. As a result, the main technological hurdle is that the data is acquired faster than it can be processed. Data reduction methods are thus increasingly important, as they allow one to extract the most relevant and important information from giant data sets. We present one such method, based on compressing the description length of an estimate of the probability distribution of a set points.

KDE Approximation Given a set of points $\{x_i\}_{i=1}^n$ in $\mathbb{R}^d$, the Kernel Density Estimate (KDE) of the data is defined as $f(x) = \frac{1}{n}\sum_{i=1}^n K_{\mathbf{H}}(x - x_i)$, where K is a "bump" of covariance $\mathbf{H}$ centered at x_i. If the number of points n is large, the KDE has a large space complexity, and the Mean Shift algorithm (see below) has a large time complexity. Thus, we wish to approximate the KDE by a much smaller KDE which is constructed from only $m \ll n$ points. Our main result is:

Theorem: *Let f be KDE with n points, and let $\hat{f}$ be a KDE constructed by sampling m times from f, and assume a diagonal bandwidth matrix $\hat{\mathbf{H}} = \hat{h}^2\mathbf{I}$. Let the expected squared L_2 distance between the two densities be given by $J = E[\int (f(x) - \hat{f}(x))^2 dx]$. Then $J \leq 4A\hat{h} + A^2\hat{h}^2V + \frac{B}{m\hat{h}^d} + \frac{ABV}{m\hat{h}^{d-1}}$ where A, B, V are constants independent of $\hat{h}$ or m.*

That is, the two KDEs f and $\hat{f}$ will be close in expectation if m is large enough, and if the bandwidth $\hat{h}$ is chosen properly as a function of m.

Fast Data Reduction The Mean Shift algorithm clusters the data $\{x_i\}_{i=1}^n$ using the KDE; each point x_i is mapped to the local maximum of the KDE in whose basin of attraction it lies. Let M be the mapping from data points to local maxima using the original KDE f, and $\hat{M}$ using the compressed KDE $\hat{f}$. We propose the following fast data reduction algorithm:

1. **Sampling:** Take m samples of the density f to yield $\{\hat{x}_j\}_{j=1}^m$. Form the new density $\hat{f}(x) = \sum_{j=1}^m K_h(x, \hat{x}_j)$.
2. **Mean Shift:** Perform Mean Shift on each of the m samples: $\hat{x}_j \rightarrow \hat{M}(\hat{x}_j)$.
3. **Map Backwards:** For each x_i, find the closest new sample $\hat{x}_{j^*}$. Then $x_i \rightarrow \hat{M}(\hat{x}_{j^*})$.

Complexity The key speed-up occurs in the second step; instead of using all n samples to compute the Mean Shift, we can use the reduced set of m samples. With a naive nearest neighbour data structure, the complexity decreases from $O(n^2)$ to $O(mn)$. It can be shown to decrease with more complex data structures as well. In practical clustering tasks, such as image segmentation, the speed-up has been observed to be between 500 and 1000 without significant degradation of the clustering.