Fast Clustering of Flow Cytometry Data via Adaptive Mean Shift

Suchismit Mahapatra, Jason Zhu, MengXiang Tang

BD Biosciences, San Jose, CA, USA

Abstract

Clustering of flow cytometry data in high-dimensional space is a major challenge in automated data analysis for high-throughput, multicolor flow cytometry. Various parametric and non-parametric clustering algorithms have been applied to achieve this. Parametric approaches rely upon a priori knowledge of the number of clusters present as well as make assumptions regarding the shape of the clusters. Non-parametric techniques on the other hand, make no such assumptions, however they tend to be computationally expensive. Mean shift technique, belonging to the latter category, had been introduced earlier to the field for clustering of flow cytometry data. In this work, the authors use Locality Sensitive Hashing (LSH) for fast and efficient Nearest Neighbor searches, thereby reducing the computational costs of the adaptive Mean Shift procedure. Further, Multithreading is employed to achieve up to ≥15x speed ups in execution times.

Context

The Mean Shift algorithm, belongs to the family of non-parametric density estimation based approaches in which the feature space is regarded as the empirical probability density function (pdf) of the represented parameters. Dense regions in the feature space correspond to local maxima of the pdf, i.e. the modes of the unknown density distribution. The algorithm has been tested with clustering subset lymphocyte cell populations in typical multicolor immunophenotyping assays. The clustering results demonstrated a degree of high agreement with those obtained with a previously reported gating approach, based on a low dimensional mixture modeling approach, for well-studied subset populations as CD4, CD8, etc. The promising results call for further investigation of this highly capable clustering algorithm. It is noteworthy that the modes found by the mean-shift procedure can provide biologists with new insight into high-dimensional flow cytometry data, as well as enabling development of new methods for automated flow cytometry data analysis.

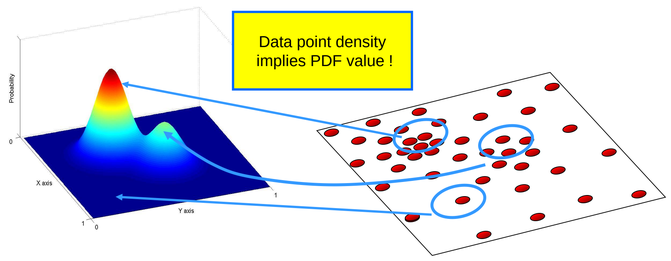

Figure 1. Density estimation based approaches assume that the data points are samples from an underlying PDF.

1 Mean shift clustering

Given N data points $\{x_i\}$ where $i \in \{1, \dots, N\}$ in $\mathbb{R}^D$ space, the sample point estimate obtained with kernel $\mathcal{K}(x)$ and bandwidth $\hbar$ is given by,

$$\hat{f}_{\mathcal{K}}(x) = \frac{1}{N\hbar^D} \sum_{i=1}^{N} \hbar\left(\left\|\frac{x - x_i}{\hbar}\right\|^2\right)$$

based on a spherically symmetric kernel $\mathcal{K}$ with bounded support satisfying

$$\mathcal{K}(x) = c_{\hbar,D}\hbar\left(\|x\|^2\right) > 0, \qquad \|x\| < 1$$

is a non-parametric estimator of the density at the location x in the feature space. The function $\hbar(x), 0 \leq x \leq 1$ is called the profile of the kernel and $c_{\hbar,D}$ is a normalization constant which assures $\int_{\mathbb{R}^D} \mathcal{K}(x)\, dx = 1$. The modes of the density function are located at $\{x | \nabla f(x) = 0\}$.

The gradient of the density estimator is given by,

$$\nabla \hat{f}_{\mathcal{K}}(x) = \frac{2c_{\hbar,D}}{N\hbar^{D+2}} \sum_{i=1}^{N} (x - x_i) g\left(\left\|\frac{x - x_i}{\hbar}\right\|^2\right)$$

$$= \frac{2c_{\hbar,D}}{N\hbar^{D+2}} \left[\sum_{i=1}^{N} g\left(\left\|\frac{x - x_i}{\hbar}\right\|^2\right)\right]\left[\frac{\sum_{i=1}^{N} x_i g\left(\left\|\frac{x - x_i}{\hbar}\right\|^2\right)}{\sum_{i=1}^{N} g\left(\left\|\frac{x - x_i}{\hbar}\right\|^2\right)} - x\right]$$

where $g(x) = -\hbar'(x)$

The first part is proportional to the density estimate at location x computed with the kernel

$$G(x) = c_{g,D}\, g\left(\|x\|^2\right)$$

and the second part

$$m_G(x) = \frac{\sum_{i=1}^{N} x_i g\left(\left\|\frac{x - x_i}{\hbar}\right\|^2\right)}{\sum_{i=1}^{N} g\left(\left\|\frac{x - x_i}{\hbar}\right\|^2\right)}$$

is called the mean-shift vector. At location x, the weighted mean of the data points selected with kernel G is proportional to the density gradient estimate obtained with kernel $\mathcal{K}$. Thus the mean shift vector always points towards the direction of the maximum increase in density.

The Mean Shift procedure successively iterates over

- Computation of the mean-shift vector $m_G(x_t)$
- Translation of the bandwidth window i.e. $x^{t+1} = x^t + m_G(x_t), t \in \{1, 2 \dots\}$

It employs a hill climbing strategy till it converges to a location, called mode, where the density gradient vanishes. See Figure 2 for an illustration.

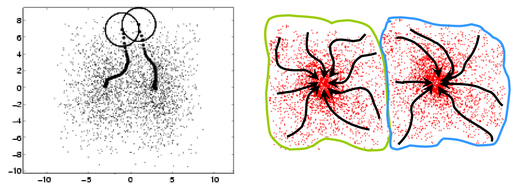

Figure 2. (Left) Illustration of Mean Shift procedure as it reaches convergence to a density gradient vanishing mode. (Right) Illustrations of the tessellations of the feature space, containing the basins of attraction, which are the regions for which all trajectories lead to the same mode.

2 Adaptive mean-shift clustering

In the basic version of Mean Shift, a single global bandwidth value $\hbar$ is employed for all the data points, whereas for the adaptive procedure, each data point $x_i$has its own bandwidth value $\hbar_i$, depending on the local distribution of points only.

The bandwidth values associated with the data points have been defined using various methods in the statistical literature. Most techniques employ a pilot density estimate. The simplest way to do it is via the nearest neighbors. Let $x_{i,k}$ be the kth nearest neighbor of the data point x_i. Thus,

$$\hbar_i = \left\|x_i - x_{i,k}\right\|^p$$

The number of neighbors k should be chosen large enough to assure that there is an increase in density within the support of most kernels having bandwidths $\hbar_i$.

3 Locality Sensitive Hashing

Locality Sensitive Hashing allows us to compute fast and efficient Nearest Neighbor searches. It belongs to a novel and interesting class of algorithms that are known as randomized algorithms. A randomized algorithm does not guarantee an exact answer but instead provides a high probability guarantee that it will return the correct answer or one close to it. By investing additional computational effort, the probability can be pushed as high as desired.

- Random hyper-planes $\hbar_1 \dots \hbar_{\mathcal{K}}$
  - Feature space sliced into $2^{\mathcal{K}}$ partitions.
  - Compare query with only $\mathbb{E}\left(\mathcal{N}/2^{\mathcal{K}}\right)$ points.
- Inexact: missed nearest neighbors ?
  - Repeat with $\mathcal{L}$ sets of $\hbar_1 \dots \hbar_{\mathcal{K}}$
- Intuitively maps $\mathbb{R}^D \implies \mathbb{R}^{\mathcal{K}}$
- Higher $\mathcal{K}$ means a more faithful representation.
- $\mathcal{L}$ introduces redundancy to cover for inexactness.

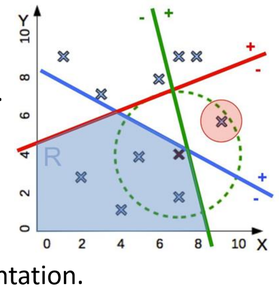

Figure 3. Illustration of how Locality Sensitive Hashing was implemented via random hyperplanes. In the actual implementation, random hyperplanes parallel to the different co-ordinate axes were used since there was no need for $2^{\mathcal{K}}$ partitions in general. Smaller values of $\mathcal{K}$ meant many more data points in a single partition which resulted in more work per thread and reduced speed up. Increased values of $\mathcal{L}$ which introduces redundancy to cover for inexactness also increased work load.

4 Fast Adaptive Mean Shift clustering

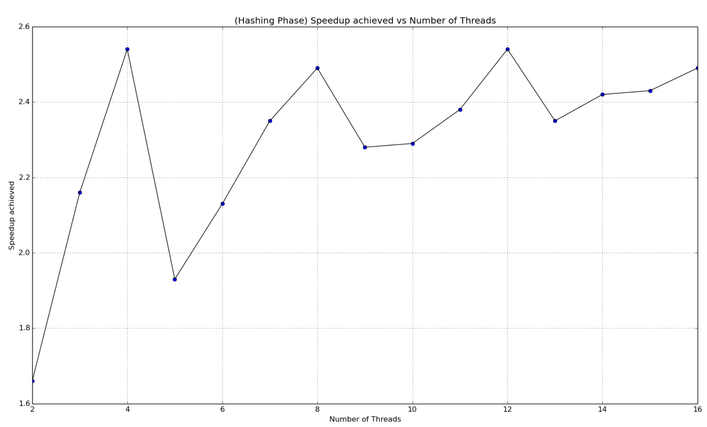

Figure 4. In the hashing phase, the high dimensional data points are mapped into a $\mathbb{R}^{\mathcal{K}}$ tessellated feature space. The hashed values are stored in a map with the $\mathcal{K}$ dimensional representation as the key and the high dimensional data points which have the same representation as the values. Experimentation using different number of threads was done to compare speedups achieved in all the different phases.

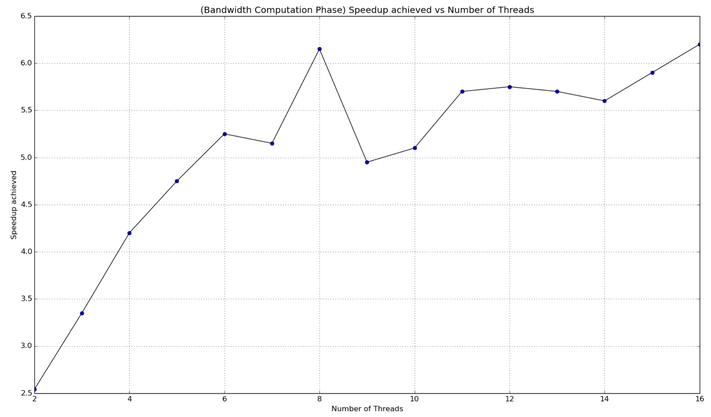

Figure 5. In the bandwidth computation phase, individual data point bandwidth was computed. The computation involved solving a kNN query to determine the bandwidth value $\hbar_i$ of the data point x_i.

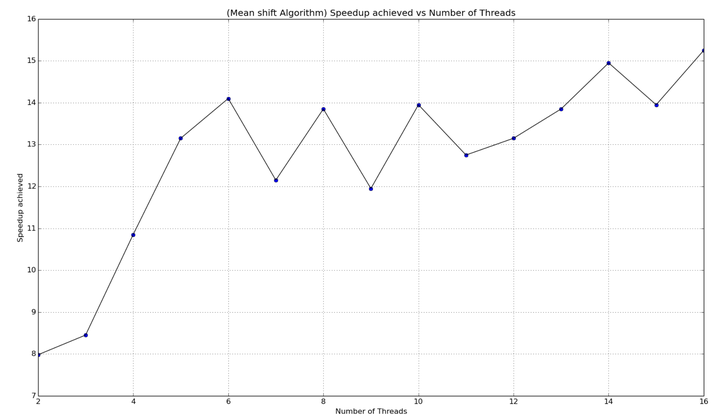

Figure 6. Speedups achieved for the Mean Shift procedure.

We experimented with multiple kernel profiles. The quality of results obtained with the Epanechnikov kernel were found to be better in general and was used in this implementation. Rather than using critical sections or message passing techniques, the use of local buffers was found to give the speedup results. This is possibly due to the heaviness of the Mutual Exclusion objects as well as the frequent Context Switching which affected performance.

5 Fast Adaptive Mean shift clustering of a 6-color immunophenotyping assay

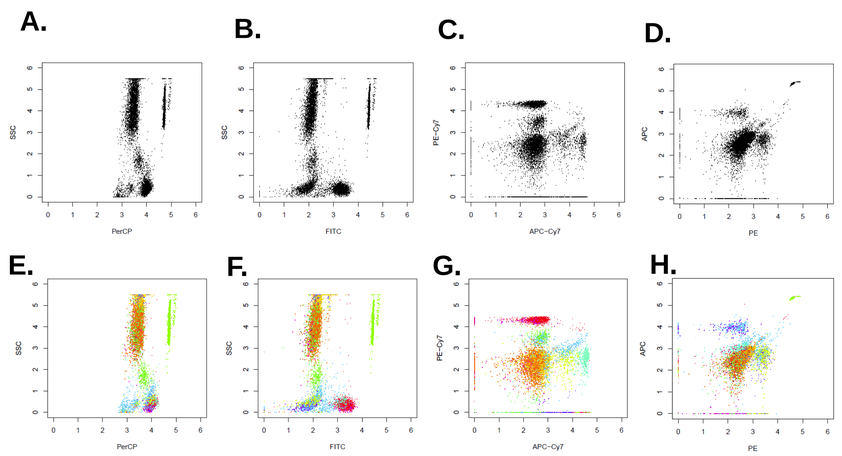

Figure 6. Adaptive mean-shift clustering was applied to clustering example data of a 6-color immunophenotyping assay with a bead counting process control. Panel A, B, C, D show the 2-D scatter plots for SSC vs PerCP, SSC vs FITC, PE-Cy7 vs APC-Cy7, and APC vs PE.

Panel E, F, G, H present the results of fast adaptive mean shift clustering using a set of 2-D scatter plots. A total of 15 major clusters (populations) was identified, and each is labeled with a distinct color.

Summary

- **The Fast Adaptive Mean Shift adaptation was able to significantly alleviate the high computational costs associated with the Mean Shift algorithm. The speedups achieved are appreciable and will allow for more efficient and faster data processing.**
- **Future work in this direction will focus on more involved procedures i.e. developing a mixture model based approach supported by the above work which will allow to accurately classify and predict "leukemia" and "healthy" cases.**