Robert Collins
CSE586

Advanced Clustering Methods

MeanShift, MedoidShift and QuickShift

References:

D. Comaniciu, P. Meer: Mean Shift: A Robust Approach toward Feature Space Analysis, IEEE Trans. Pattern Analysis Machine Intel., Vol. 24(5), 603-619, 2002

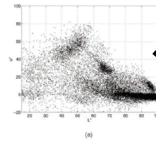
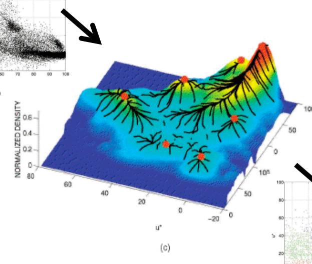
Yaser Sheikh, Erum Khan, Takeo Kanade, "Mode-seeking via Medoidshifts", IEEE International Conference on Computer Vision, 2007.

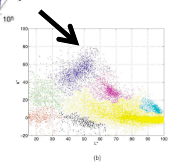
A. Vedaldi and S. Soatto, "Quick Shift and Kernel Methods for Mode Seeking," in Proceedings of the European Conference on Computer Vision (ECCV), 2008

Robert Collins
CSE586

MeanShift Clustering

nonparametric mode seeking

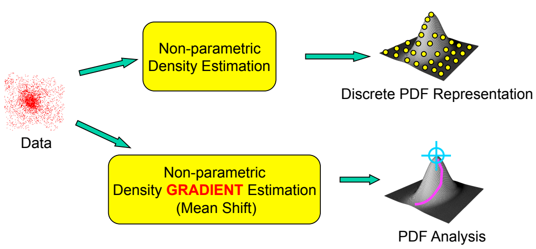


Don't need to know number of clusters in advance!

Robert Collins
CSE586

What is Mean Shift ?

A tool for:
Finding modes in a set of data samples, manifesting an underlying probability density function (PDF) in R^N

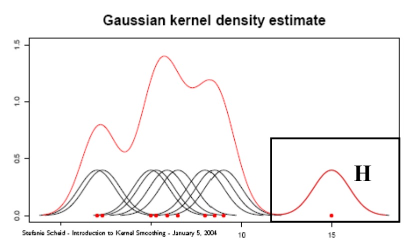


Ukrainitz&Sarel, Weizmann

Robert Collins
CSE586

Parzen Density Estimation

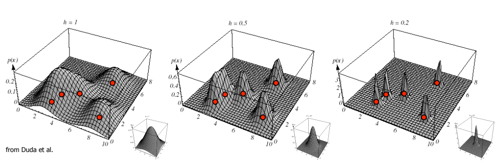
Given a set of data samples x_i; $i=1...n$
Convolve with a kernel function H to generate a smooth function $f(x)$
Equivalent to superposition of multiple kernels centered at each data point



Robert Collins
CSE586

Density Estimation at Different Scales

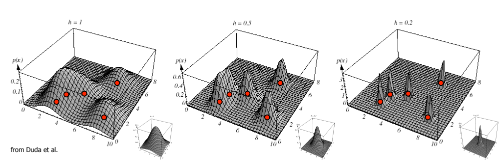
- Example: Density estimates for 5 data points with differently-scaled kernels
- Scale influences accuracy vs. generality (overfitting)



Robert Collins
CSE586

Smoothing Scale Selection

- Unresolved issue: how to pick the scale (sigma value) of the smoothing filter
- Answer for now: a user-settable parameter

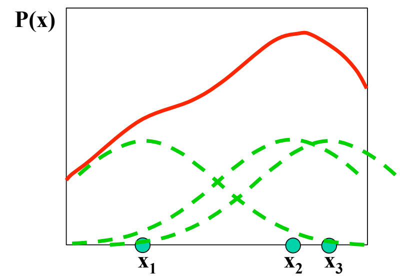


Mean-Shift

The mean-shift algorithm is a hill-climbing algorithm that seeks modes of a density without explicitly computing that density.

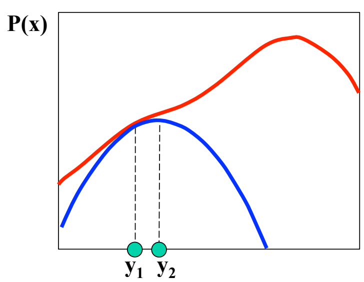
The density is implicitly represented by raw samples and a kernel function. The density is the one that would be computed if Parzen estimation was applied to the data with the given kernel.

MeanShift as Mode Seeking



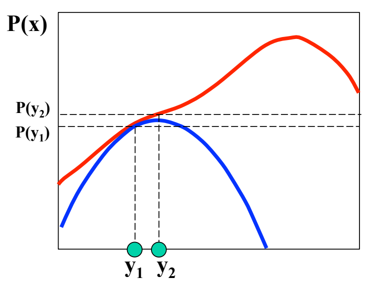
Parzen density estimation

MeanShift as Mode Seeking



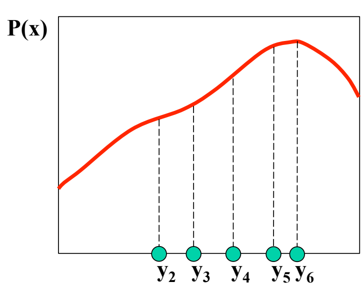
Seeking the mode from an initial point y_1
Construct a tight convex lower bound b at y_1 [$b(y_1)=P(y_1)$]
Find y2 to maximize b

MeanShift as Mode Seeking



Note, $P(y_2) >= b(y_2) > b(y1) = P(y_1)$
Therefore $P(y2) > P(y_1)$

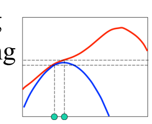
MeanShift as Mode Seeking



Move to y_2 and repeat until you reach a mode

Aside: EM Optimizes a Lower Bound

- The optimization EM is performing can also be described as constructing a lower bound, optimizing it, then repeating until convergence.



- In the case of EM the lower bound is constructed by Jensen's Inequality:

$$\log \frac{1}{N}\Sigma f(x_i) \geq \underbrace{\frac{1}{N}\Sigma \log f(x_i)}_{\text{Not necessarily quadratic, but solvable in closed-form}}$$

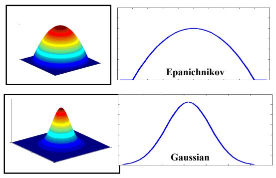
MeanShift in Equations

Parzen density estimate using $\{x_1, x_2, \ldots, x_N\}$

$$P(y) = \frac{1}{N}\sum_{j=1}^{N} \phi(\|y-x_j\|^2)$$

where $\Phi(r)$ is a convex, symmetric windowing (weighting) kernel; e.g. $\Phi(r)=\exp(-r)$ with $r = \|x_i-x\|^2$

Examples



$\Phi(r)= 1-r$

$\Phi(r)= \exp(-r)$

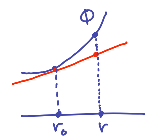
MeanShift in Equations

Parzen density estimate using $\{x_1, x_2, \ldots, x_N\}$

$$P(y) = \frac{1}{N}\sum_{j=1}^{N} \phi(\|y-x_j\|^2)$$

where $\Phi(r)$ is a convex, symmetric windowing (weighting) kernel; e.g. $\Phi(r)=\exp(-r)$

Form lower bound of each Φ_j by expanding about $r_0 = \|y_{old}-x_j\|^2$

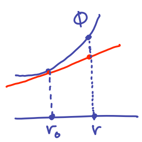


$$\phi(r) \ge \phi(r_0) + (r-r_0)\dot{\phi}(r_0)$$

note: first-order taylor series expansion
$f(r) \sim f(r_0) + (r-r_0)f'(r_0) + \text{h.o.t.}$

MeanShift in Equations

Lower bound of Φ at r_0



$$\phi(r) \ge \phi(r_0) + (r-r_0)\dot{\phi}(r_0)$$

now let $r_0 = \|y_{old} - x_j\|^2$
$r = \|y - x_j\|^2$

and substitute into Parzen equation

to get a quadratic lower bound (wrt y) on the Parzen density function

$$P(y) \ge P(y_{old}) + \frac{1}{N}\sum_{j=1}^{N} \left(\|y-x_j\|^2 - \alpha_j\right) w_j$$

where

$$w_j = \dot{\phi}(\|y_{old} - x_j\|^2)$$

$$\alpha_j = \|y_{old} - x_j\|^2$$

MeanShift in Equations

Quadratic lower bound on the Parsen density function

$$P(y) \ge P(y_{old}) + \frac{1}{N}\sum_{j=1}^{N} \left(\|y-x_j\|^2 - \alpha_j\right) w_j$$

Want to find y to maximize this. We therefore need to maximize

$$\arg\max_y \frac{1}{N}\sum_{j=1}^{N} (y-x_j)^T(y-x_j)\, w_j$$

$$= \frac{1}{N}\sum (y^Ty - 2y^Tx_j + x_j^Tx_j)\, w_j$$

Take derivative wrt y, set to 0, and solve

$$\frac{\partial}{\partial y} = \frac{1}{N}\sum w_j [2y - 2x_j] = 0$$

$$\Rightarrow \sum w_j y = \sum w_j x_j$$

$$y = \sum w_j x_j \Big/ \sum w_j$$

mean-shift update eqn

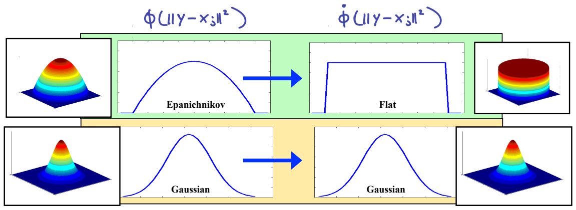
MeanShift in Equations

$$P(y) = \frac{1}{N}\sum_{j=1}^{N} \phi(\|y-x_j\|^2)$$ Density estimate

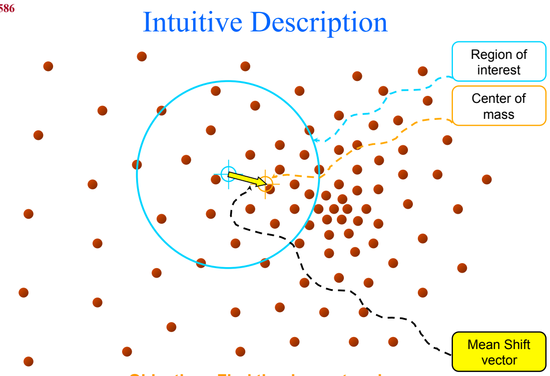
$$y = \sum w_j x_j \Big/ \sum w_j$$ Mean-shift update eqn

$$w_j = \dot{\phi}(\|y_{old} - x_j\|^2)$$ Weights at each iteration

$\phi(\|y-x_j\|^2)$ $\quad$ $\dot{\phi}(\|y-x_j\|^2)$

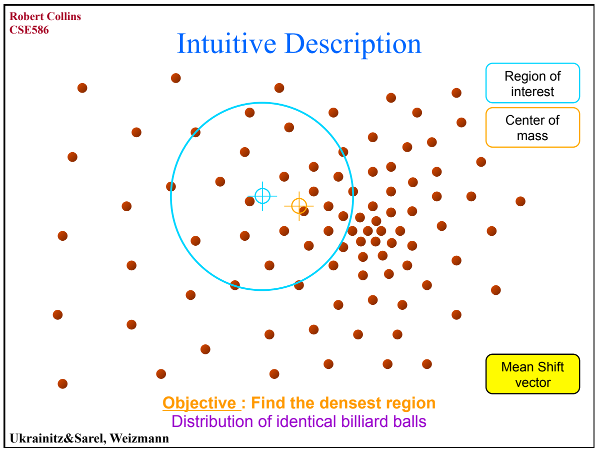


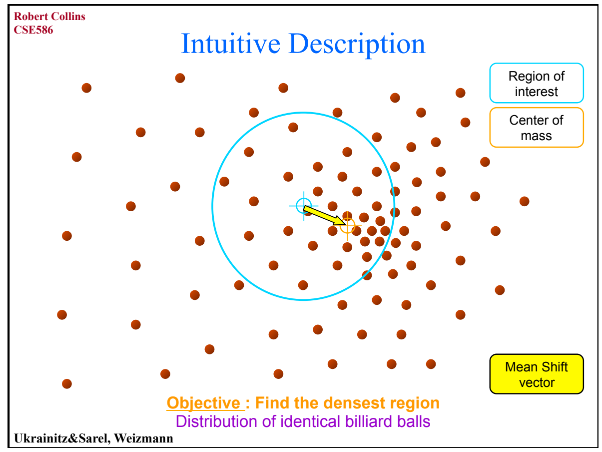
Intuitive Description

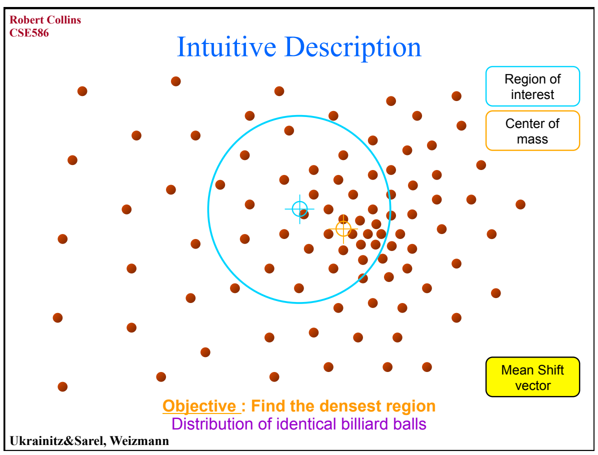


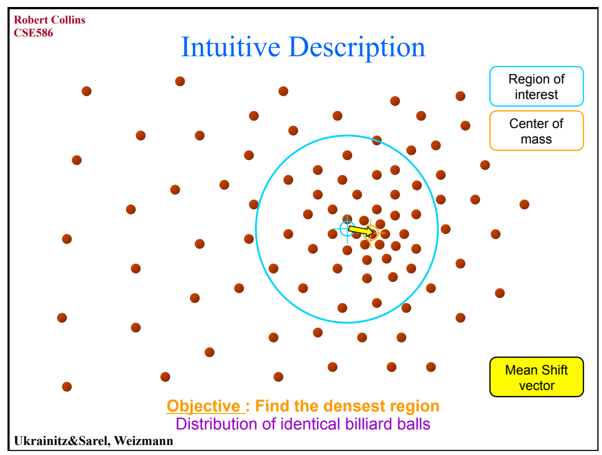
Objective : Find the densest region
Distribution of identical billiard balls

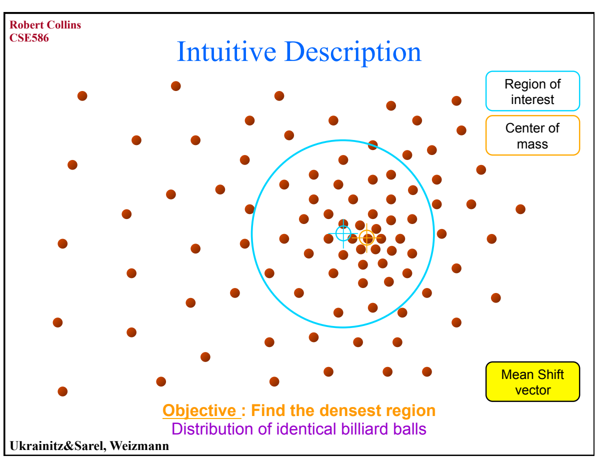
Ukrainitz&Sarel, Weizmann

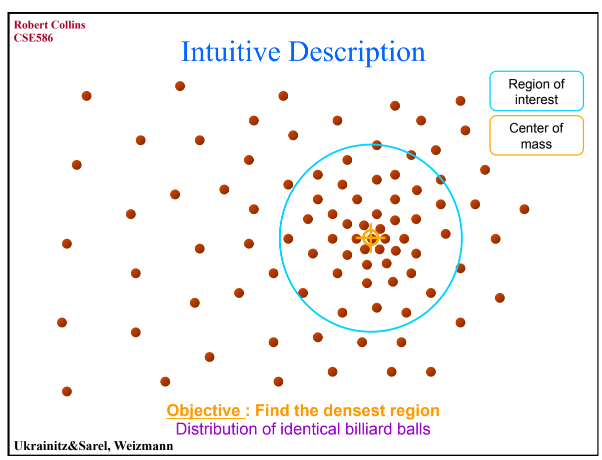

Robert Collins
CSE586
Intuitive Description
Region of interest
Center of mass
Mean Shift vector
Objective : Find the densest region
Distribution of identical billiard balls
Ukrainitz&Sarel, Weizmann


Robert Collins
CSE586
Intuitive Description
Region of interest
Center of mass
Mean Shift vector
Objective : Find the densest region
Distribution of identical billiard balls
Ukrainitz&Sarel, Weizmann

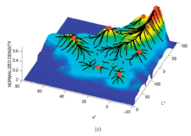

Robert Collins
CSE586
Intuitive Description
Region of interest
Center of mass
Mean Shift vector
Objective : Find the densest region
Distribution of identical billiard balls
Ukrainitz&Sarel, Weizmann


Robert Collins
CSE586
Intuitive Description
Region of interest
Center of mass
Mean Shift vector
Objective : Find the densest region
Distribution of identical billiard balls
Ukrainitz&Sarel, Weizmann


Robert Collins
CSE586
Intuitive Description
Region of interest
Center of mass
Mean Shift vector
Objective : Find the densest region
Distribution of identical billiard balls
Ukrainitz&Sarel, Weizmann


Robert Collins
CSE586
Intuitive Description
Region of interest
Center of mass
Objective : Find the densest region
Distribution of identical billiard balls
Ukrainitz&Sarel, Weizmann

MeanShift Clustering



For each data point x_i

Let $y_i(0) = x_i$; $t = 0$

Repeat

Compute weights $w_1, w_2, \dots w_N$ for all data points, using $y_i(t)$

Compute updated location $y_i(t+1)$ using mean-shift update eqn

If ($y_i(t)$ and $y_i(t+1)$ are "close enough")
declare convergence to a mode and exit
else
$t = t + 1$;

end (repeat)

number distinct modes found = number clusters

points that converge to "same" mode are labeled as belonging to one cluster

What is MedoidShift?

Instead of solving for continuous-valued y to maximize the quadratic lower bound on the Parzen density estimate,

consider only a discrete set of y locations, taken from the initial set of data points $\{x_1,x_2,\dots,x_N\}$

That is, for the first step towards the mode

$$y_i(1) = \underset{y\in\{x_1,x_2,\dots,x_N\}}{\operatorname{argmax}} \frac{1}{N}\sum_{j=1}^{N} \|y-x_j\|^2 \,\dot{\phi}(\|x_i-x_j\|^2)$$

y is now chosen from a discrete set of data points

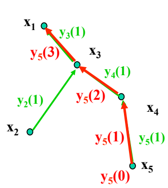
$y_i(0) = x_i$

MedoidShift

Implications of this strategy for clustering:

Since we are choosing the best y from among $\{x_1,x_2,\dots,x_N\}$, $y_i(1)$ will be one of the original data points.

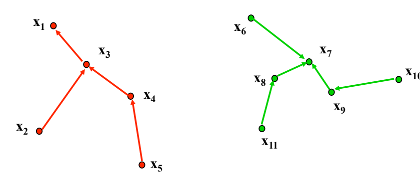
Therefore, we only need to compute a single step starting from each data point, because $y_i(2)$ will then be the result of $y_{y_i(1)}(1)$



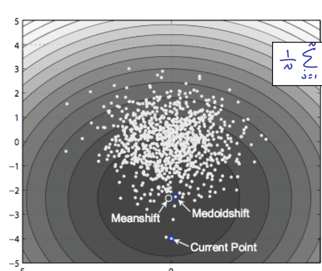
MedoidShift

Different clusters are found as separate connected components (trees, actually... so can do efficient tree traversal to locate all points in each cluster.

Also note: no need for heuristic distance thresholds to test for convergence to a mode, or to determine membership in a cluster.



MedoidShift



$$\frac{1}{N}\sum_{j=1}^{N} \|y-x_j\|^2 \,\dot{\phi}(\|x_i-x_j\|^2)$$

Figure 1. Contour plot of Equation 2 for the current point. Mean-shift selects the location that minimizes this function exactly while medoidshift selects the data point that best minimizes it.

MedoidShift behavior asymptotically approaches MeanShift behavior as number of data samples approaches infinity.

Advantages of MedoidShift

•Automatically determines the number of clusters (as does mean-shift)

•Previous computations can be reused when new samples are added or old samples are deleted (good for incremental clustering applications)

•Can work in domains where only distances between samples are defined (e.g. EMD or ISOMAP); there is no need to compute a mean.

•No need for heuristic terminating conditions (don't need to use a distance threshold to determine convergence)

MeanShift vs MedoidShift

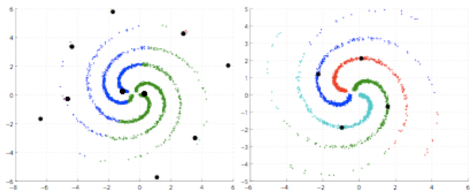
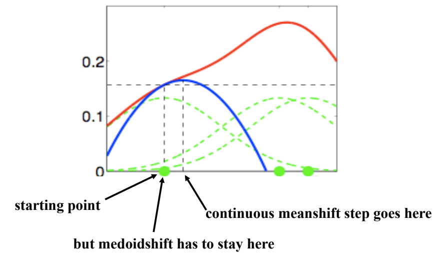


Figure 8. Clustering results obtained by using meanshift (left) in Euclidean space and medoidshift (right) using the manifold distance function described in [18].

Drawbacks of MedoidShift

•$O(N^3)$ [although not really – Vedaldi paper]

•Can fail to find all the modes of a density, thereby causing over-fragmentation



Iterated MedoidShift

•keep a count of how many points move to x_j and apply a new iteration of medoidshift with points weighted by that count.

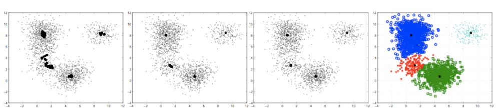


Figure 2. Left to right: Clustering using medoidshift after 1, 2, and 3 iterations, and the final labels for each point.

•Each iteration is essentially using a smoother Parzen estimate

Seeds of QuickShift!

In future work, we intend to investigate a further speed up. The requirement that for the current location $\mathbf{y}_k$, $\mathbf{y}_{k+1}$ be the data point that minimizes Equation 2 is not strictly required for convergence. The sufficient condition in Equation 7 in Theorem 2.1 is simply that the new point $\mathbf{y}_{k+1}$ have a score *better* than $\mathbf{y}_k$. If this condition is used instead of the exact condition the computation can be terminated early. An implementation showed that the computational saving obtained from this was roughly 80% greater than that of the exact algorithm. Further investigation is needed to determine the degree of approximation error. The extension

Vedaldi and Soatto

Contributions:

- If using Euclidean distance, MedoidShift can actually be implemented in O(N^2)
- generalize to non-Euclidean distances using kernel trick
- show that MedoidShift does not always find all modes
- propose QuickShift, an alternative approach that is simple, fast, and yields a one-parameter family of clustering results that explicitly represent the tradeoff between over- and under-fragmentation.

Kernel MedoidShift

Insight: medoidshift is defined in terms of pairwise distances between data points.

Therefore, using the "kernel trick", we can generalize it to handle nonEuclidean distances defined by a kernel matrix K
(similar to how many machine learning algorithms are "kernelized")

What is QuickShift?

A combination of the above kernel trick and...

instead of choosing the data point that optimizes the lower bound function b(x) of the Parzen estimate P(x)

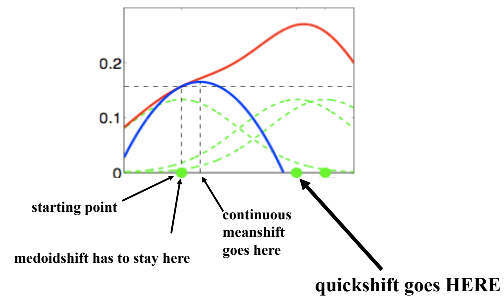
choose the closest data point that yields a higher value of P(x) directly

$$y_i(1) = \operatorname*{argmin}_{j : P(x_j) > P(x_i)} D(x_i, x_j)$$

$$P(x_i) = \frac{1}{N} \sum_{j=1}^{N} \phi(D(x_i, x_j))$$

QuickShift vs MedoidShift

•Recall our failure case for medoidshift



Comparisons

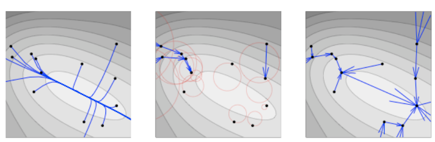


Fig. 1. Mode seeking algorithms. Comparison of different mode seeking algorithms (Sect. 2) on a toy problem. The black dots represent (some of) the data points $x_i \in \mathcal{X} \subset \mathbb{R}^2$ and the intensity of the image is proportional to the Parzen density estimate $P(x)$. **Left.** Mean shift moves the points uphill towards the mode approximately following the gradient. **Middle.** Medoid shift approximates mean shift trajectories by connecting data points. For reason explained in the text and in Fig. 2, medoid shifts are constrained to connect points comprised in the red circles. This disconnects portions of the space where the data is sparse, and can be alleviated (but not solved) by iterating the procedure (Fig. 2). **Right.** Quick shift (Sect. 3) seeks the energy modes by connecting nearest neighbors at higher energy levels, trading-off mode over- and under-fragmentation.

Quickshift Benefits

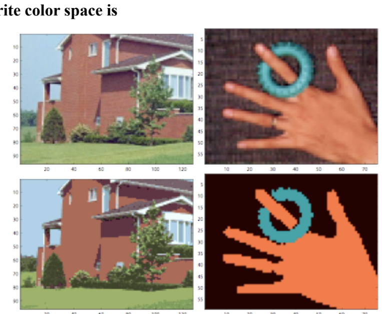
•Similar benefits to medoidshift, plus additional ones:

•All points are in one connected tree. You can therefore explore various clusterings by choosing a threshold T and pruning edges such that $d(x_i,x_j) < T$

•Runs much faster in practice than either meanshift or medoidshift

Applications

image segmentation by clustering (r,g,b), or (r,g,b,x,y), or whatever your favorite color space is



Applications

manifold clustering using ISOMAP distance

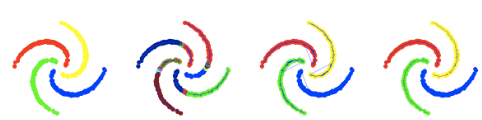


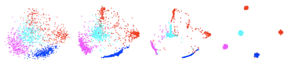
Fig. 4. Clustering on a manifold. By using kernel ISOMAP we can apply kernel mean and medoid shift to cluster points on a manifold. For the sake of illustration, we reproduce an example from [20]. From left to right: Kernel mean shift (7.8s), *non-iterated* kernel medoid shift (0.18s), iterated kernel medoid shift (0.48s), quick shift (0.12s). We project the kernel space to three dimensions $d = 3$ as the residual dimensions are irrelevant. All algorithms but non-iterated medoid shift segment the modes successfully. Compared to [20], medoid shift has complexity $O(dN^2)$, (with a small constant and $d = 3 \ll N$) instead of $O(N^3)$ (small constant) or $O(N^{2.38})$ (large constant)

Robert Collins
CSE586

Applications

clustering bag-of-features signatures using Chi-squared kernel (distances between histograms) to discover visual "categories" in Caltech-4

clusters in 400D space (visualized by projection to rank 2)



5 discovered categories. Ground-truth category "airplanes" is split into two clusters

Algorithm	Scal.	Rate [%]	Time [s]
mean	no	93.9	29.9
mean	yes	94.6	89.0
mean+medoid	no	93.9	3.63
mean+medoid	yes	94.5	6.28
quick	no	96.3	0.62
quick	yes	96.1	0.70

Fig. 6. Automatic visual categorization. We use kernel mean shift to cluster bag-of-features image descriptors of 1600 images from Caltech-4 (four visual categories: airplanes, motorbikes, faces, cars). **Top.** From left to right, iterations of kernel mean shift on the bag-of-features signatures. We plot the first two dimensions of the rank-reduced kernel space (z vectors) and color the points based on the ground truth labels. In the rightmost panel the data converged to five points, but we artificially added random jitter to visualize the composition of the clusters. **Bottom.** Samples from the five clusters found (notice that airplane are divided in two categories). We also report the clustering quality, as the percentage of correct labels compared to the ground truth (we merge the two airplanes categories into one), and the execution time. We use basic implementations of the algorithms, although several optimizations are possible.

Robert Collins
CSE586

Applications

Clustering images using a bhattacharya coefficient on 5D histograms composed of (L,a,b,x,y) values

Figure 10. Clustering a collection of images. Each row shows images selected from a single cluster found by medoidshift

1 2 3 4 5 6 7 8 9

ure 11. Key-frames selected from a video of 1500 frames. Each key-frame corresponds to a mode as estimated by our algorithm.