Efficient Visualization of Large-scale Data Tables through Reordering and Entropy Minimization

Nemanja Djuric, Slobodan Vucetic
Temple University, Philadelphia
December 10th, 2013, in Dallas, Texas

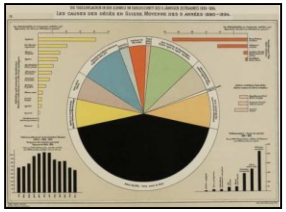
Data visualization

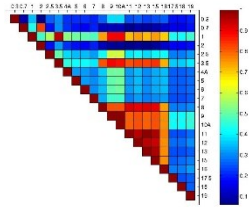
- Immediate feedback that can lead to faster knowledge discovery
  - Intuitive way of interacting with unknown data
  - Practical even for non-experts
- Visualizing large data matrices
  - Data given in a form of a large 2-D table
  - Long history, however novel methods required to tackle emerging Big Data problems

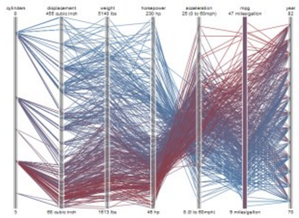
Visualizing data tables

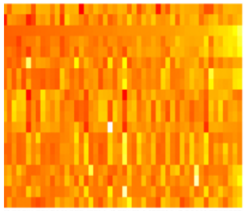
- Existing approaches

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
0.9058	0.6797	0.7943	0.0497	0.3786	0.5468	0.6820	0.7011	0.0942	0.0012
0.1270	0.6551	0.3112	0.9027	0.8116	0.5211	0.0424	0.6663	0.5985	0.4624
0.9134	0.1626	0.5285	0.9448	0.5328	0.2316	0.0714	0.5391	0.4709	0.4243
0.6324	0.1190	0.1656	0.4909	0.3507	0.4889	0.5216	0.6981	0.6959	0.4609
0.0975	0.4984	0.6020	0.4893	0.9390	0.6241	0.0967	0.6665	0.6999	0.7702
0.2785	0.9597	0.2630	0.3377	0.8759	0.6791	0.8181	0.1781	0.6385	0.3225
0.5469	0.3404	0.6541	0.9001	0.5502	0.3955	0.8175	0.1280	0.0336	0.7847
0.9575	0.5853	0.6892	0.3692	0.6225	0.3674	0.7224	0.9991	0.0688	0.4714
0.9649	0.2238	0.7482	0.1112	0.5870	0.9880	0.1499	0.1711	0.3196	0.0358
0.1576	0.7513	0.4505	0.7803	0.2077	0.0377	0.6596	0.0326	0.5309	0.1759
0.9706	0.2551	0.0838	0.3897	0.3012	0.8852	0.5186	0.5612	0.6544	0.7218
0.9572	0.5060	0.2290	0.2417	0.4709	0.9133	0.9730	0.8819	0.4076	0.4735
0.4854	0.6991	0.9133	0.4039	0.2305	0.7962	0.6490	0.6692	0.8200	0.1527
0.8003	0.8909	0.1524	0.0965	0.8443	0.0987	0.8003	0.1904	0.7184	0.3411
0.1419	0.9593	0.8258	0.1320	0.1948	0.2619	0.4538	0.3689	0.9686	0.6074
0.4218	0.5472	0.5383	0.9421	0.2259	0.3354	0.4324	0.4607	0.5313	0.1917
0.9157	0.1386	0.9961	0.9561	0.1707	0.6797	0.8253	0.9816	0.3251	0.7384
0.7922	0.1493	0.0782	0.5752	0.2277	0.1366	0.0835	0.1564	0.1056	0.2428
0.9595	0.2575	0.4427	0.0598	0.4357	0.7212	0.1332	0.8555	0.6110	0.9174
0.6557	0.8407	0.1067	0.2348	0.3111	0.1068	0.1734	0.6448	0.7788	0.2691
0.0357	0.2543	0.9619	0.3532	0.9234	0.6538	0.3909	0.3763	0.4235	0.7655
0.8491	0.8143	0.0046	0.8212	0.4302	0.4942	0.8314	0.1909	0.0908	0.1887



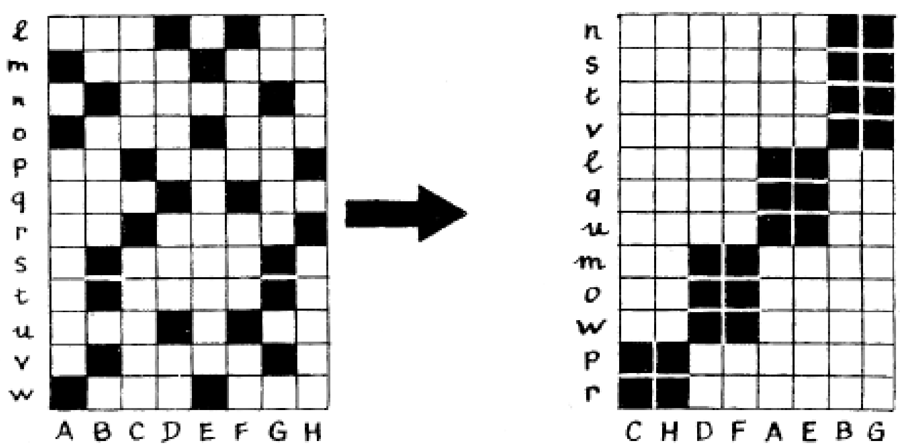






Data reordering

- Idea: Reorder data matrix so that similar rows and columns are grouped together



Jacques Bertin, 1967.

Data reordering: Related work

- Used in bioinformatics, anthropology, archeology, …
- Low-dimensional projection approaches
  - PCA, LLE, Spectral Clustering (SC)
- Hierarchical clustering (HC) approaches
  - HC with optimal leaf ordering
- Traveling salesman solvers
  - Lin-Kernighan heuristic

Algorithm	Time	Space
PCA	$\mathcal{O}(n\log(n))$	$\mathcal{O}(n)$
LLE	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2)$
SC	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2)$
HC	$\mathcal{O}(n^2)$	$\mathcal{O}(n)$
HC-olo	$\mathcal{O}(n^3)$	$\mathcal{O}(n^2)$
LK	$\mathcal{O}(n^{2.2})$	$\mathcal{O}(n)$
TSP-means	$\mathcal{O}(n\log(n))$	$\mathcal{O}(n)$

EM-ordering

- Reordering from the viewpoint of data compression
  - Assume data set $D = \{\mathbf{x}_i, i = 1, .., n\}$, where $\mathbf{x}_i = [x_{i1}, x_{i2}, .., x_{im}]$ is an m-dimensional example
  - **Task**: Reorder the data so that it is maximally compressible
- Differential Predictive Coding (DPC)
  - Use local context to code the value of $\mathbf{x}_i$

$$D = \{\mathbf{x}_i, i = 1,\ldots,n\} \rightarrow D_{DPC} = \{\mathbf{x}_1, \varepsilon_2, \ldots, \varepsilon_n\}$$
$$\text{where } \varepsilon_i = (\mathbf{x}_i - \mathbf{x}_{i-1}),\ i = 2,\ldots,n$$

EM-ordering: Intuition

- Before reordering:

3	3
5	1
2	4
4	2
1	5

DPC →

3	3
2	-2
-3	3
2	-2
-3	3

- After reordering:

1	5
2	4
3	3
4	2
5	1

DPC →

1	5
1	-1
1	-1
1	-1
1	-1

EM-ordering

- Entropy of differences used to estimate data compressibility
  - Differences independent, sampled from $N(0, \sigma_j^2), j = 1, \ldots, m$

$$H(\varepsilon) = \frac{n}{2}(m \cdot \log(2\pi) + \sum_{j=1}^{m} \log(\sigma_j(\varepsilon))) + 0.5 \sum_{i=2}^{n} \sum_{j=1}^{m} \frac{(\mathbf{x}_{\pi(i),j} - \mathbf{x}_{\pi(i-1),j})^2}{\sigma_j^{\,2}}$$

- Solve the following optimization problem

$$(\pi^*, \{\sigma_1^*, \ldots, \sigma_m^*\}) = \arg \min_{\pi, \{\sigma_1, \ldots, \sigma_m\}} H(\varepsilon)$$

EM-ordering

- The optimization can be split into two parts
  1. Fix variance of differences → Minimize the overall distance between neighbors in the ordering (equivalent to TSP)
  2. Fix ordering → Find variance of the differences
- Or, more formally:

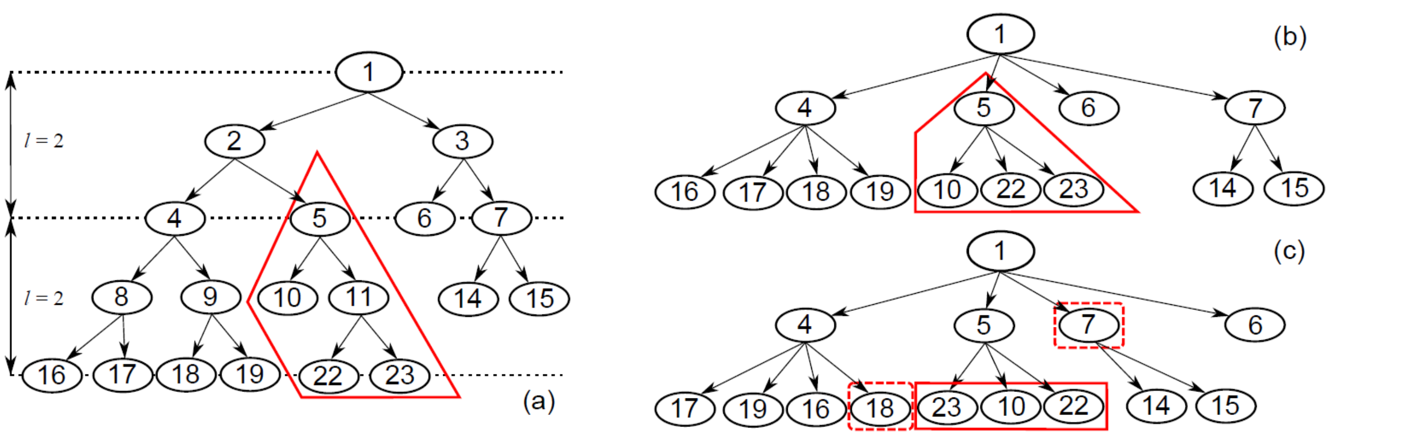
Algorithm 1 EM-ordering

Inputs: data set D; initial guess for $\{\sigma_j\}_{j=1,\ldots,m}$
Output: ordered set D; learned $\{\sigma_j\}_{j=1,\ldots,m}$

1. **repeat** until convergence
2.   **run** TSP solver for current σ_j to find π
3.   **calculate** σ_j for current ordering of D

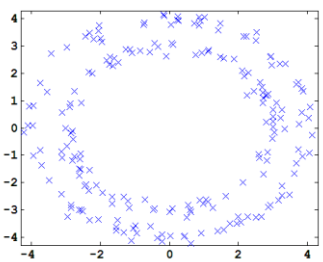
TSP-solver

- The best TSP solvers have super-quadratic time complexity
- We propose an $O(n\log(n))$ method, called TSP-means
  1. Create a 2^l-ary tree through recursive runs of k-means (k = 2)
  2. Traverse the tree breath-first, and solve TSP defined on children of the current node and their immediate neighbors

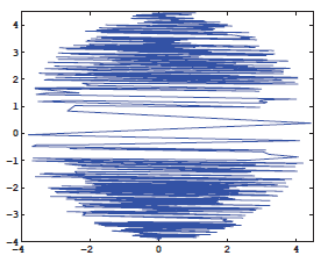


Results: Synthetic data set

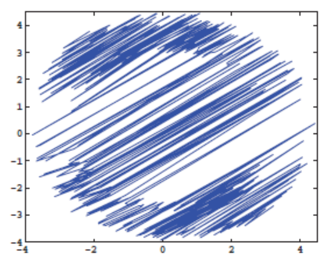
- Synthetic 2-D data set with data points located on two concentric circles of different radii



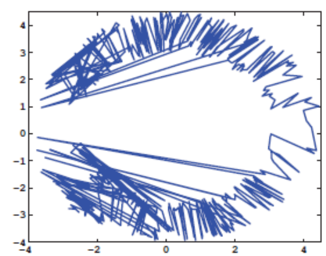
Original



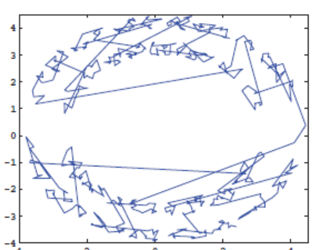
PCA



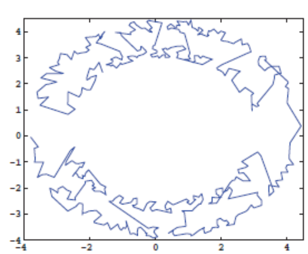
LLE



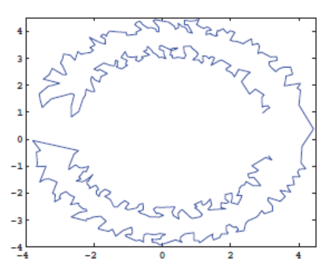
SC



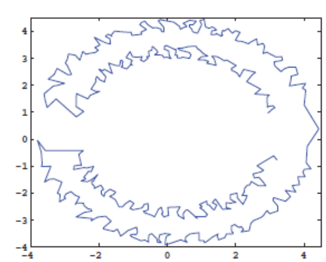
HC



HC-olo



LK

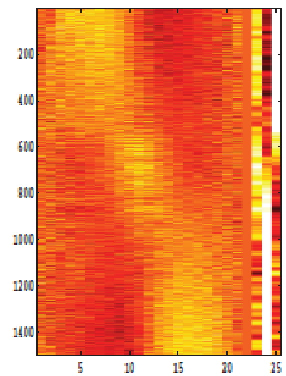


TSP-means

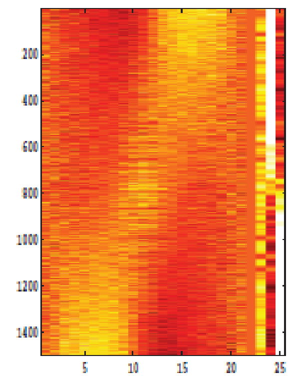
Results: *Waveform* data set

Figure of Merit scores are given in the parentheses:

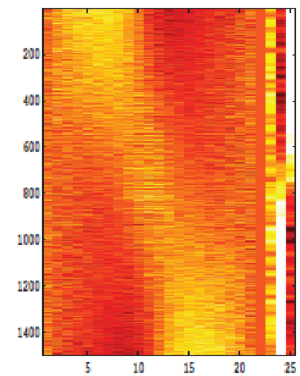
$$\mathrm{FOM}(\pi) = \frac{1}{n-1}\sum_{i=1}^{n-1} I\big(y(\pi(i)) \neq y(\pi(i+1))\big)$$



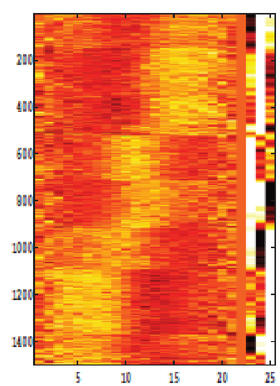
PCA (0.462)



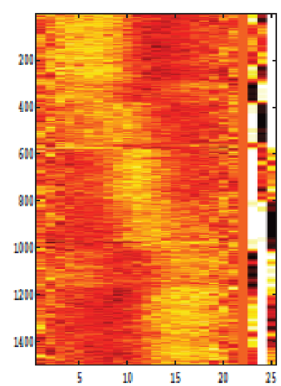
LLE (0.461)



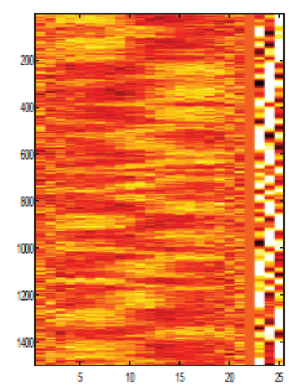
SC (0.461)



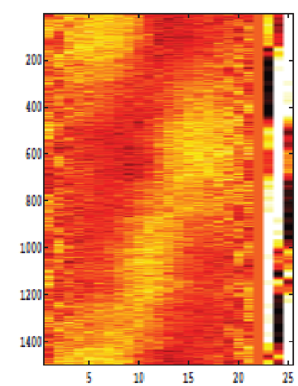
HC (0.266)



HC-olo (0.250)



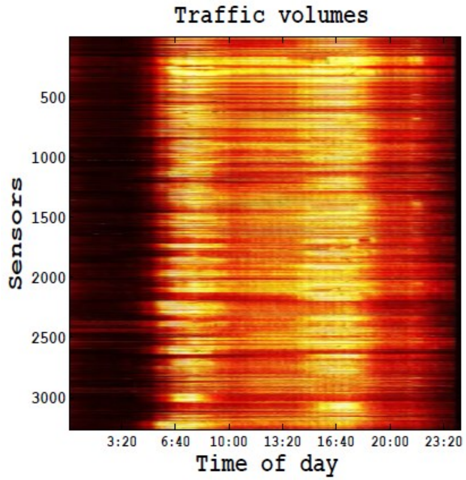
LK (0.249)



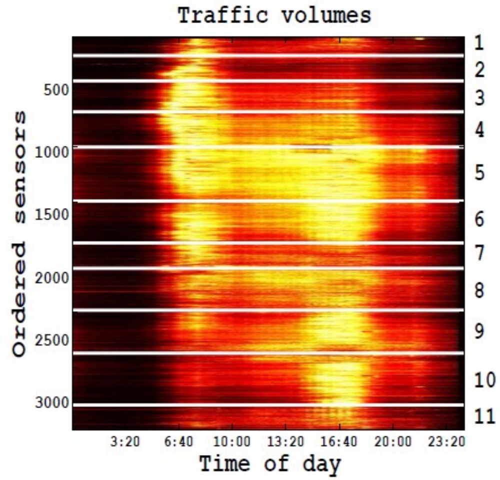
TSP-means (0.239)

Results: Real-world applications

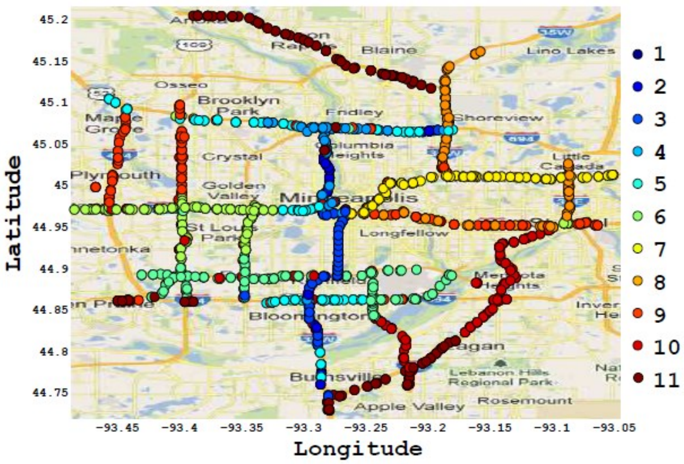
- Minneapolis traffic data set



Original data set



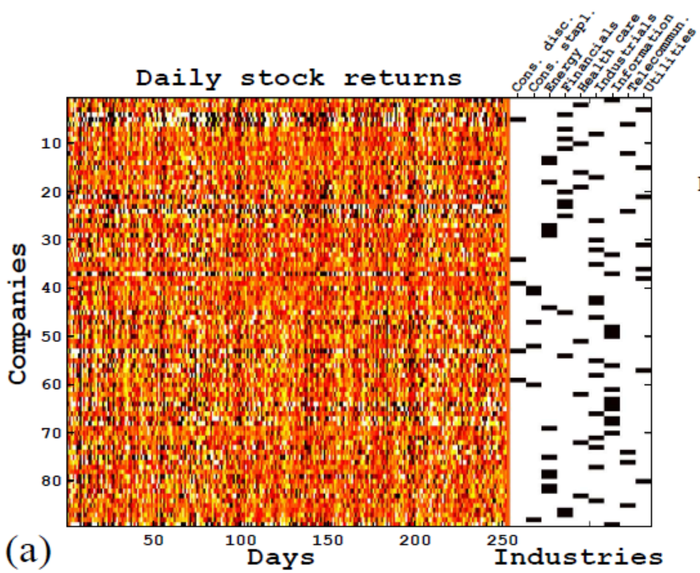
Reordered data set



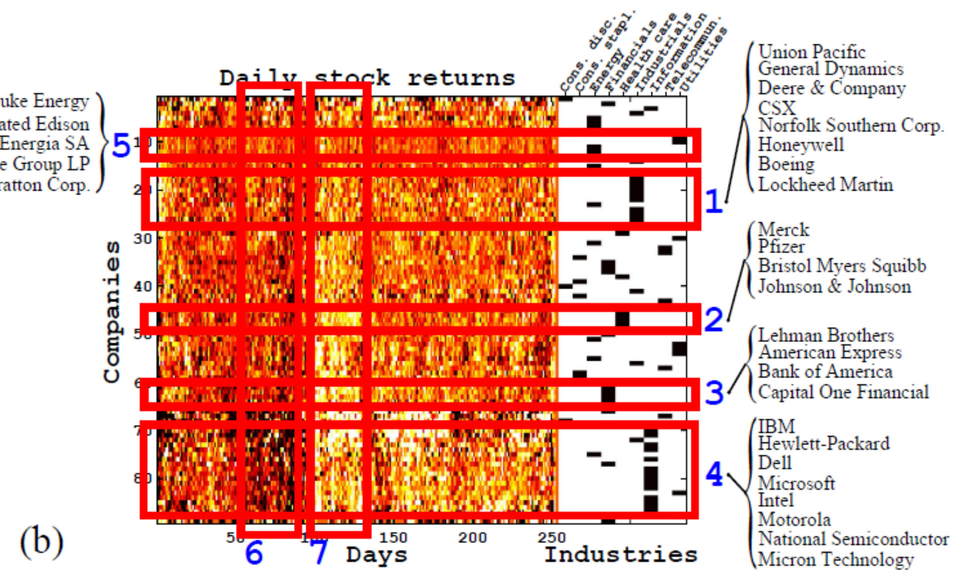
Locations of the sensors

Results: Real-world applications

- Stocks data set



Original data set



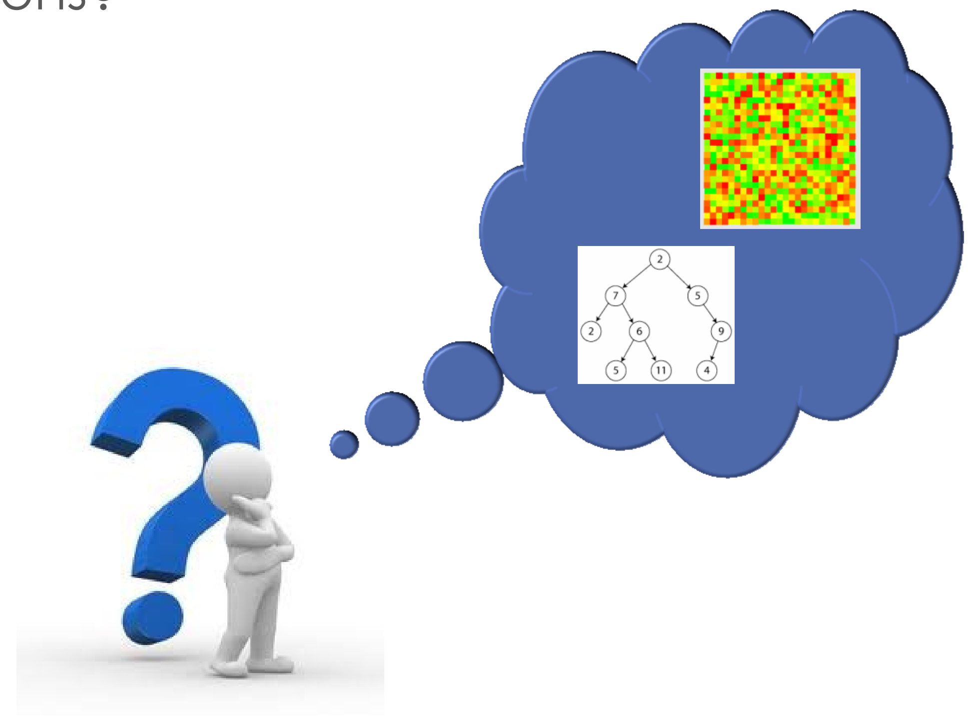
Reordered data set

Conclusion

- Inadequacy of standard visualization tools in large-scale setting is apparent
  - Novel methods required to address Big Data problems
- EM-ordering and TSP-means
  - Fast, efficient knowledge discovery
  - Easily parallelizable
  - Interesting results on real-world data
- Future work
  - Binary, categorical data?
  - Development of an easy-to-use visualization software

Thank you!

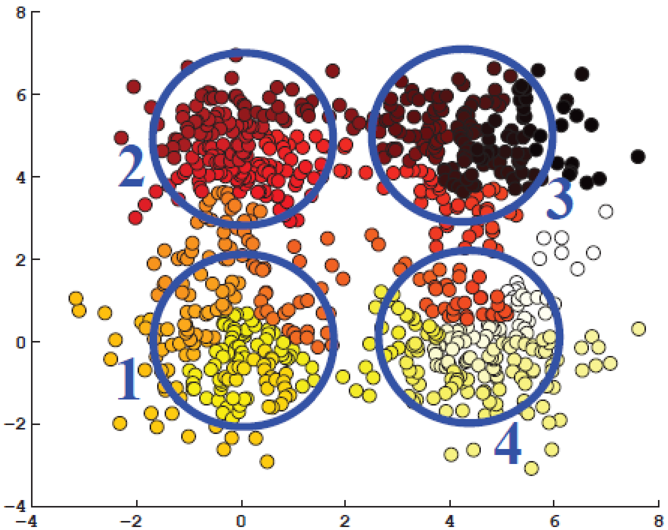
- Questions?



LK vs. TSP-means

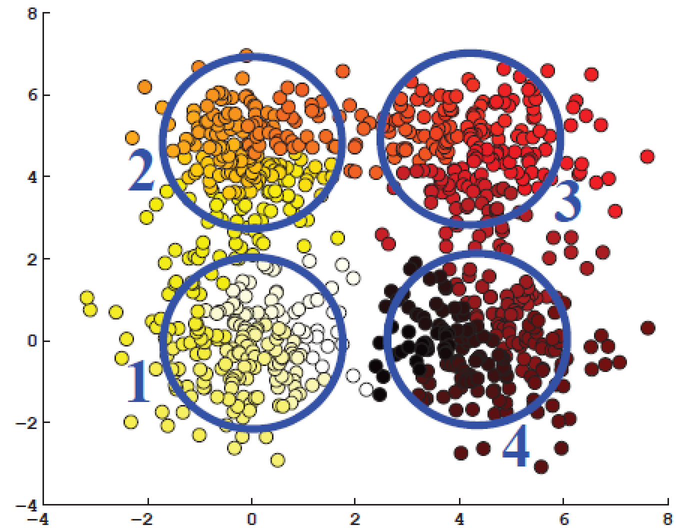
- Effect of user-set parameter *l*
  - Global vs. local solution

(a)



LK solution

(b)



TSP-means solution