Web-scale image clustering revisited

Yannis Avrithis[†], **Yannis Kalantidis**[‡], Evangelos Anagnostopoulos[†], Ioannis Z. Emiris[†]

[†]University of Athens, [‡]Yahoo Labs San Francisco

ICCV 2015, 13-16th December 2015, Santiago, Chile

Outline

Outline

Large-scale clustering

problem formulation

- given a dataset X of n points in $\mathbb{R}^d$, find k cluster centroids minimizing distortion (as in k-means)

k-means iteration

- assignment step: for every point, find closest centroid
- update step: given point assignments, update centroids

related ideas & challenges

approximations & speed-ups

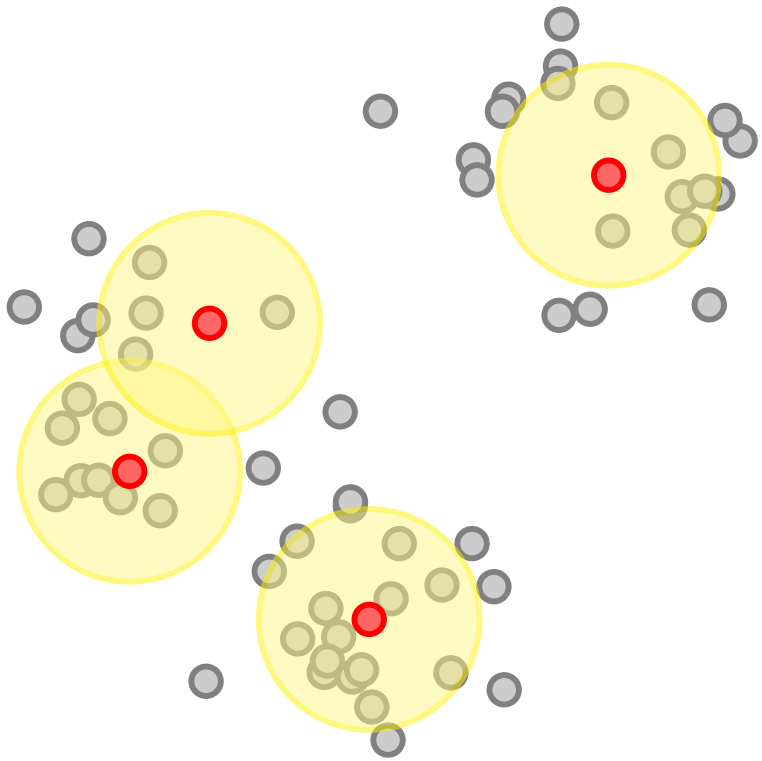
- the assignment step is the bottleneck
- approximate k-means [Philbin *et al.* CVPR, 2007]: use ANN to speed-up assignment step – all data points needed in memory
- binary k-means [Gong *et al.* CVPR, 2015]: binarize points and centroids – data now in compressed form, search in Hamming space

Ranked retrieval

[Broder et al. WSDM, 2014]

inverse search

- data remain fixed across iterations: index points, search for centroids
- dataset required in memory

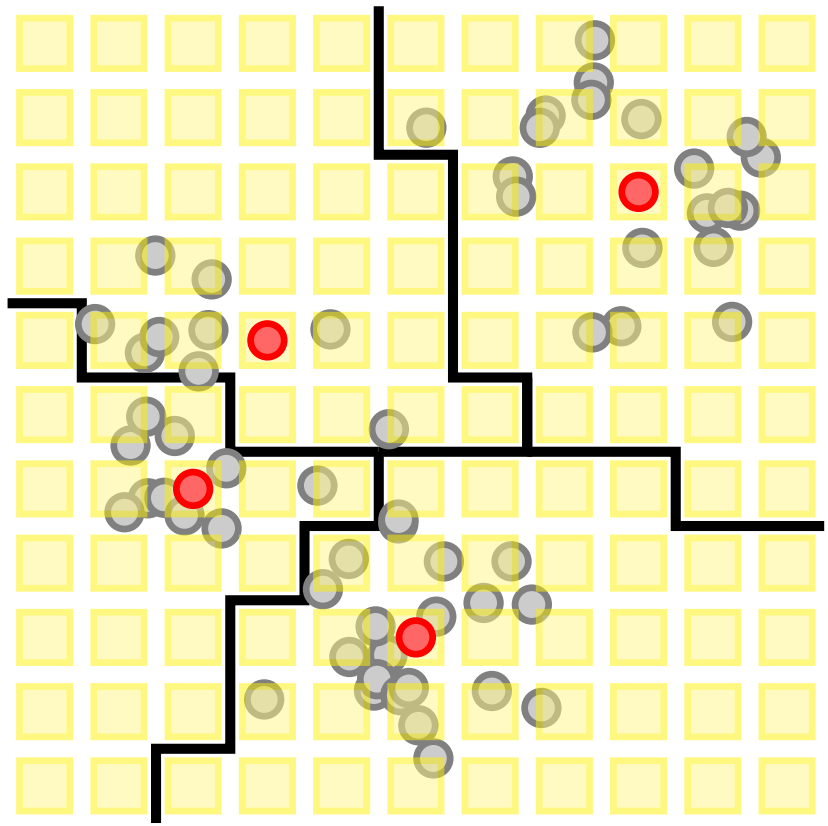


Dimensionality-recursive vector quantization

[Avrithis, ICCV, 2013]

data compression & inverse search

- quantize points to centroids using *inverted multi-index* [Babenko & Lempitsky, 2012], adopt inverse search
- search is a propagation on a 2d-grid, joint priority queue

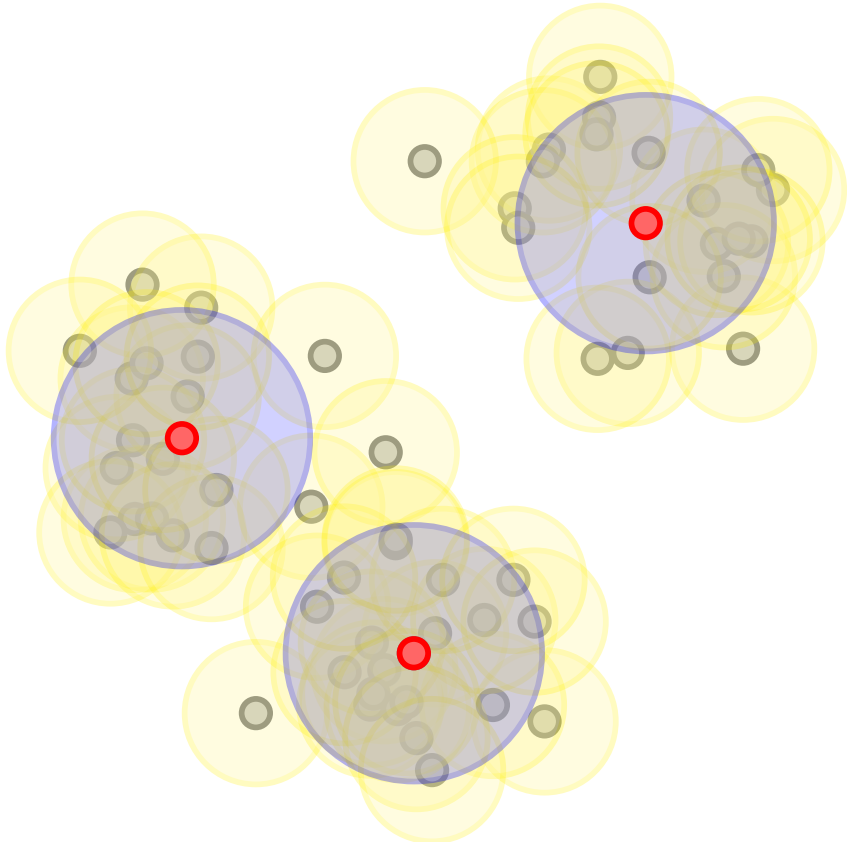


Expanding Gaussian mixtures

[Avrithis & Kalantidis, ECCV, 2012]

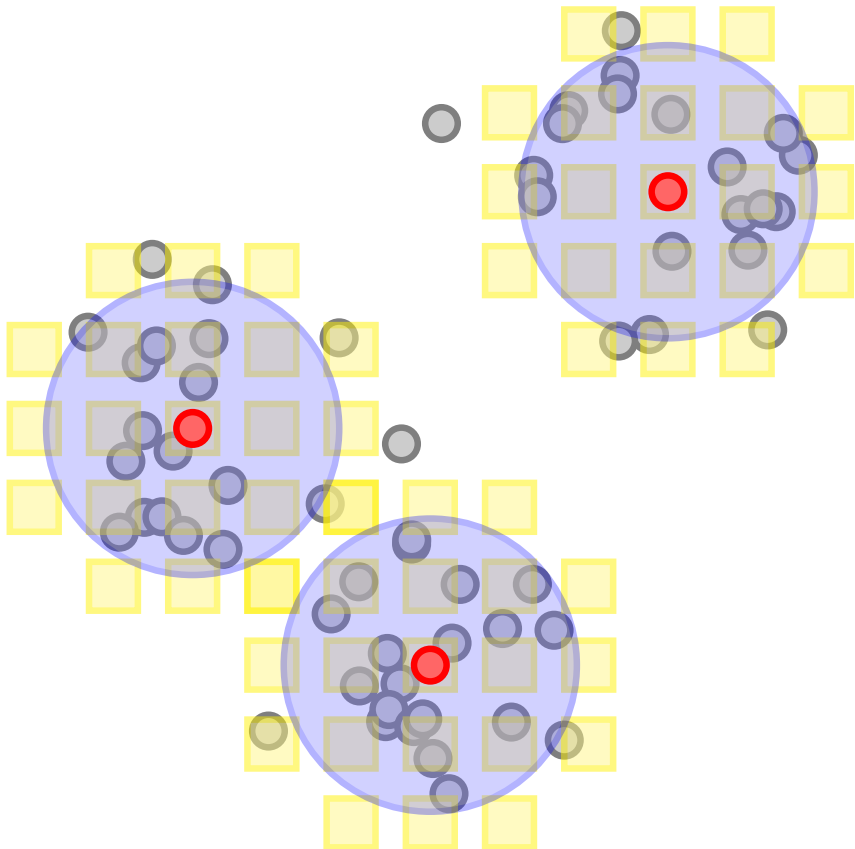
dynamic estimation of the number of centroids

- probabilistic model that allows estimation of cluster overlap
- point-to-centroid search & centroid-to-centroid search



Inverted Quantized k-means (IQ-means)

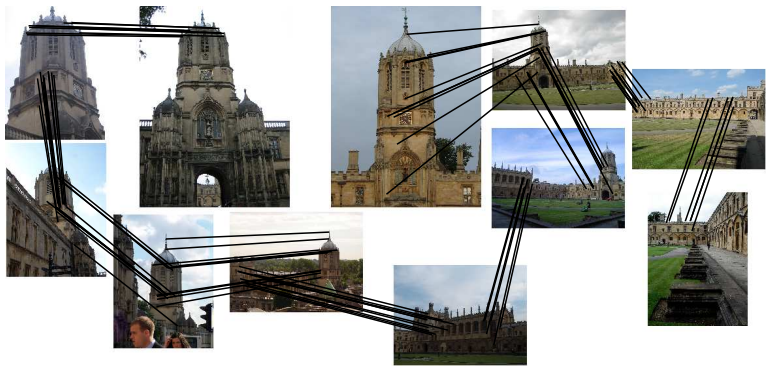
- subspace quantization & search via multi-index
- centroid-to-cell search, independent queries per centroid
- dynamic estimation of k at nearly zero cost



Web-scale image clustering

[Chum & Matas, PAMI, 2010]

- detect seed images using minHash
- grow seeds via retrieval & expansion $\Rightarrow$ 100K images



revisiting with IQ-means

- cluster 100M images in less than an hour on a single machine

Outline

Quantization

compressing the dataset

- express $\mathbb{R}^d$ as the Cartesian product of two orthogonal subspaces, $S^1 \times S^2$, of $d/2$ dimensions each – subject to optimization [Ge *et al.* , 2013]
- train two sub-codebooks U^1, U^2 of size s independently on projections of sample data on S^1, S^2
- codebook $U = U^1 \times U^2$ contains $s \times s$ *cells* – can be seen as a discrete two dimensional *grid* [Babenko & Lempitsky, 2012]
- vector $x = (x^1, x^2)$ can be quantized to a cell using quantizer $q(x) = (q^1(x^1), q^2(x^2))$, where $q^\ell(x^\ell) = \arg\min_{u^\ell \in U^\ell} \|x^\ell - u^\ell\|$ for $\ell = 1, 2$

Representation

discarding original data

- for cell u_α, probability $p_\alpha = |X_\alpha|/n$, with $X_\alpha = \{x \in X : q(x) = u_\alpha\}$
- the mean $\mu_\alpha = \frac{1}{|X_\alpha|} \sum_{x \in X_\alpha} x$ of all points in X_α is kept for each cell u_α
- cells with their sample mean μ_α and probability p_α replace the original data
- create an index with cell means

Update step

moving the centroids

- for all $c_m \in C$:

$$c_m \leftarrow \frac{1}{P_m} \sum_{\alpha \in A_m} p_\alpha \mu_\alpha,$$

where:

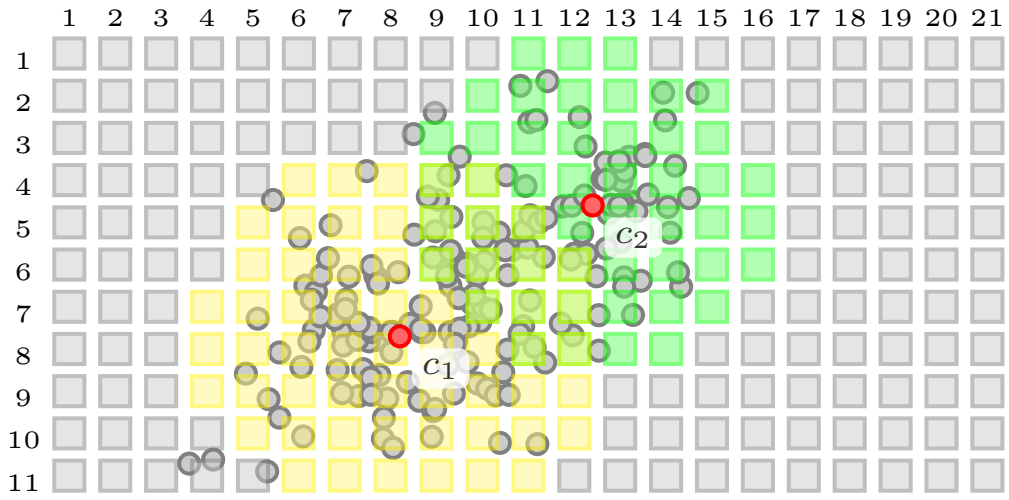
- $A_m = \{\alpha \in I : a(u_\alpha) = m\}$: the indices of all cells assigned to c_m during the assignment step
- $P_m = \sum_{\alpha \in A_m} p_\alpha$: the proportion of points assigned to centroid c_m, with $a(u) = \arg\min_{c_m \in C} \|u - c_m\|$

Assignment step

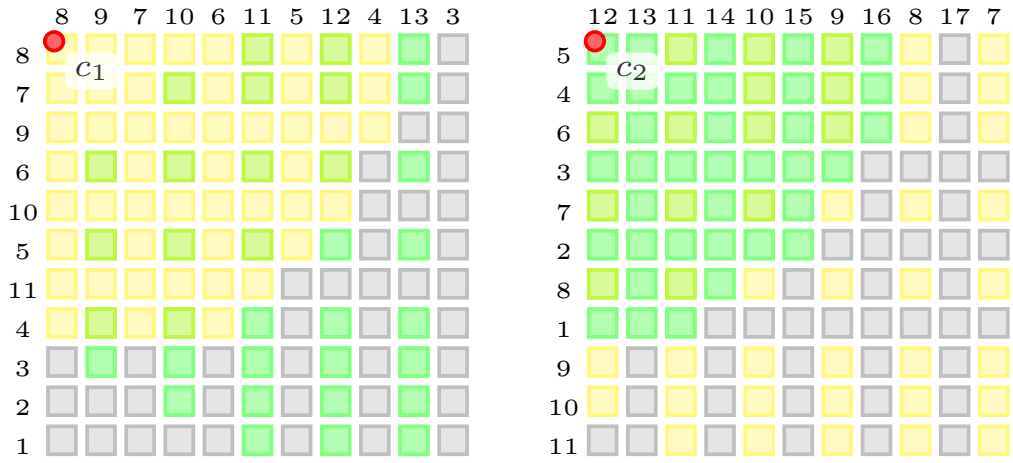
multi-index search independently for every centroid

- for each centroid c_i, the w nearest sub-codewords are found in U^1, U^2, and ordered by ascending distance to c_i, for $i = 1, 2$
- a $w \times w$ *search block* is thus determined for c_i
- the multi-sequence [Babenko & Lempitsky, 2012] algorithm is used for traversing the cells in the search block
- termination: count the total number of underlying points in visited cells, and terminates when this reaches a target number T

Centroid-to-cell search



visited cells on original grid



search blocks for c_1, c_2

Dynamic estimation of k

centroid-to-centroid search

- record nearest centroid for each cell
- during search: keep list of neighboring centroids (*i.e.* other centroids that have visited the same cells – no extra cost)

centroid modeling

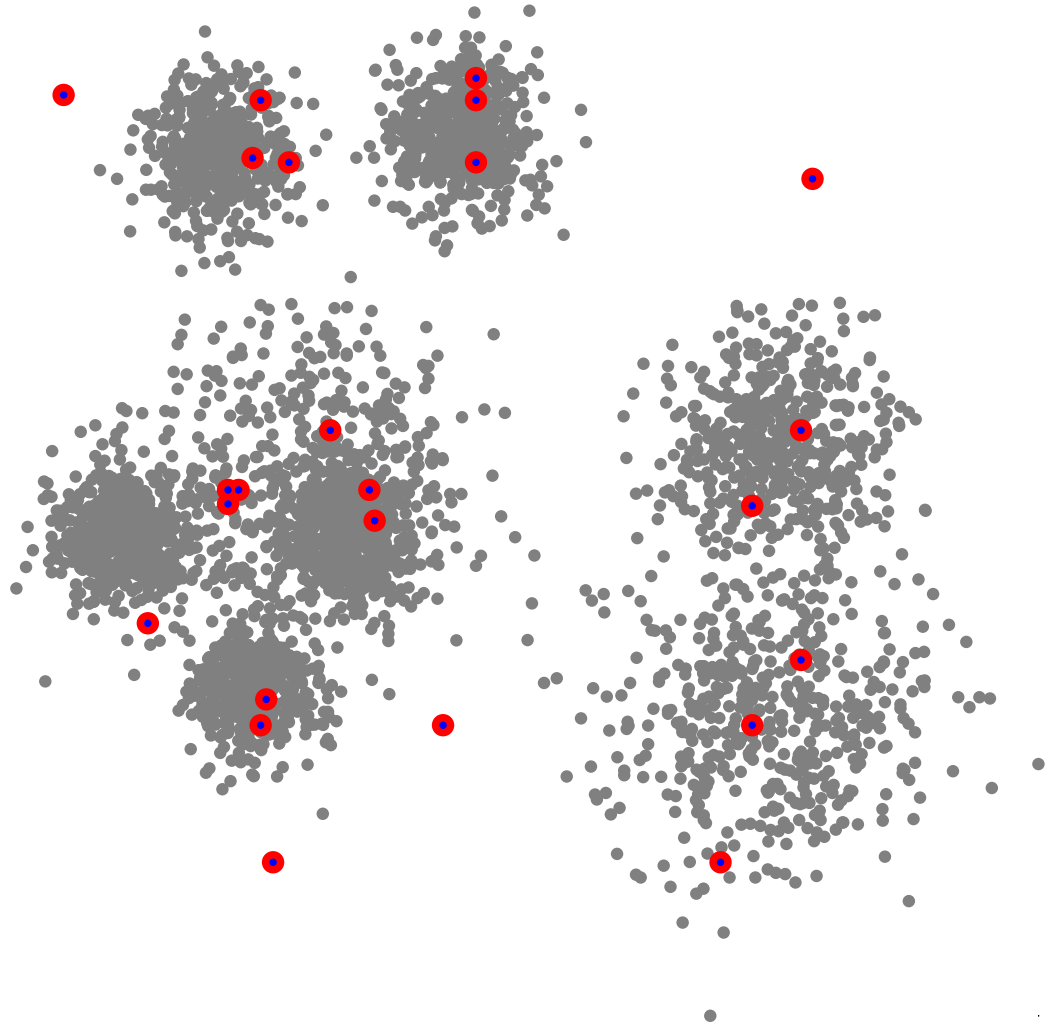
- model the distribution of points assigned to cluster c_m by an isotropic normal density $\mathcal{N}(x|c_m, \sigma_m)$ as in EGM [Avrithis & Kalantidis, ECCV, 2012]

$$\sigma_m^2 \leftarrow \frac{1}{P_m} \sum_{\alpha \in A_m} p_\alpha \|\mu_\alpha - c_m\|^2.$$

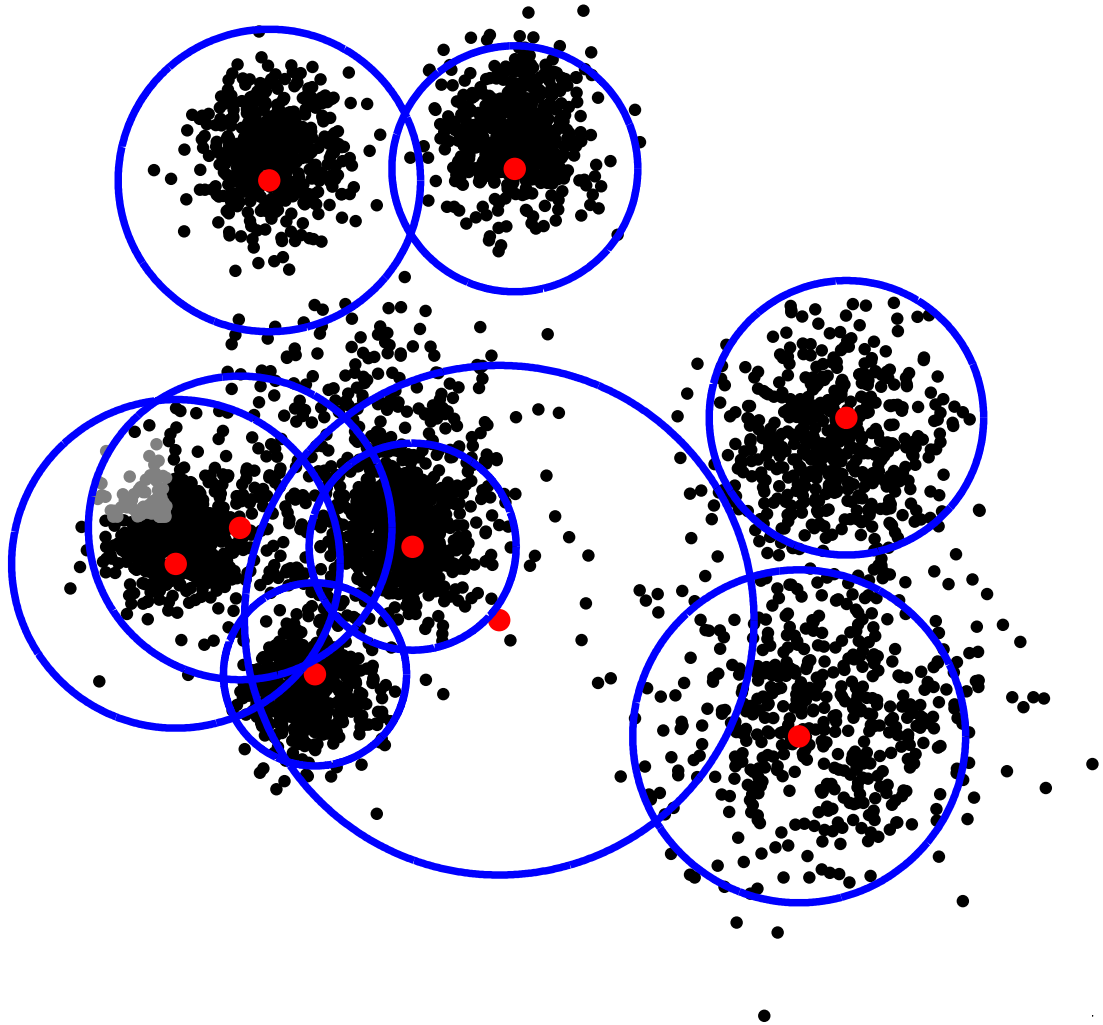
centroid deletion

- iterate over all clusters m in descending order of population P_m
- for every centroid, compute overlap with neighboring centroids
- purge clusters that overlap too much with all clusters kept so far

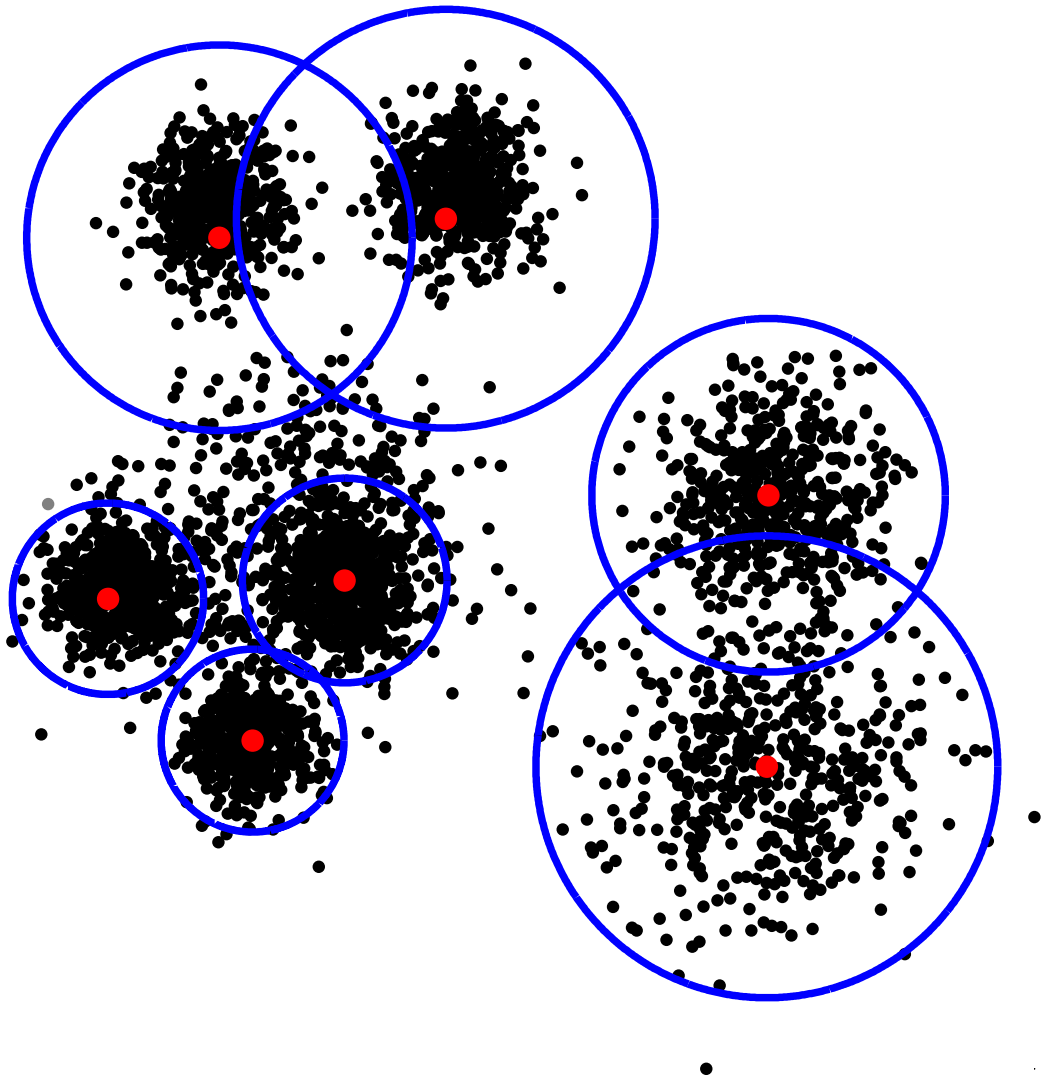
Dynamic IQ-means



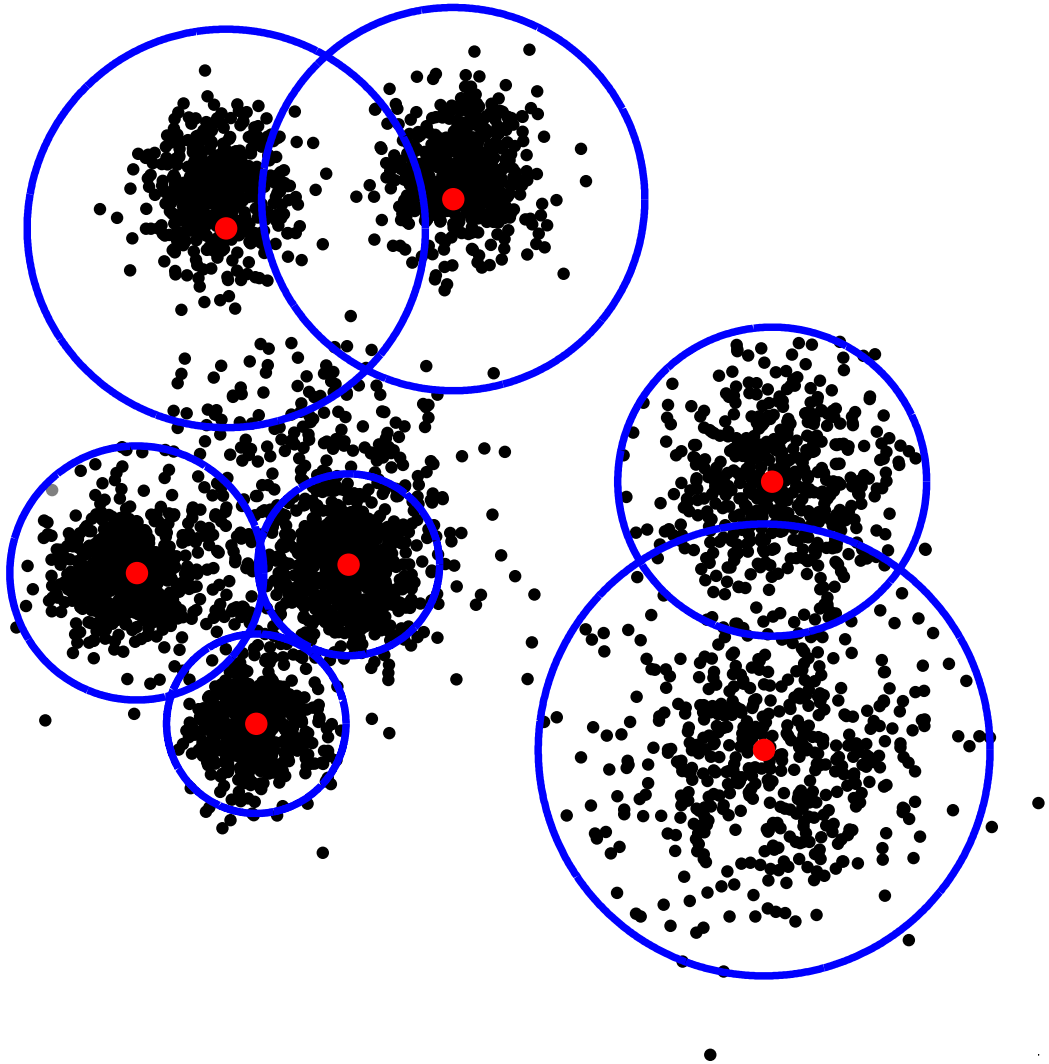
Dynamic IQ-means



Dynamic IQ-means



Dynamic IQ-means



Outline

Experiments

datasets

- SIFT1M [Jegou *et al.* , PAMI, 2011]: 1M 128-dimensional SIFT vectors, and a learning set of 100K vectors
- Paris [Wayand *et al.* , RMLE, 2010]: 500K images from Paris, ground truth of 79 landmark clusters covering 94K dataset images
- Yahoo Flickr Creative Commons 100M (YFCC100M) [Thomee *et al.* , CACM, 2015]: 100 million public Flickr images with a creative commons license

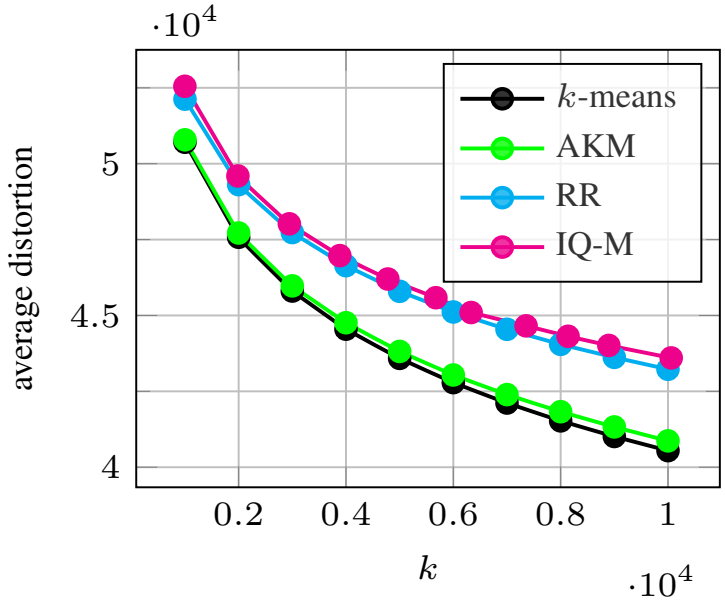
image representation

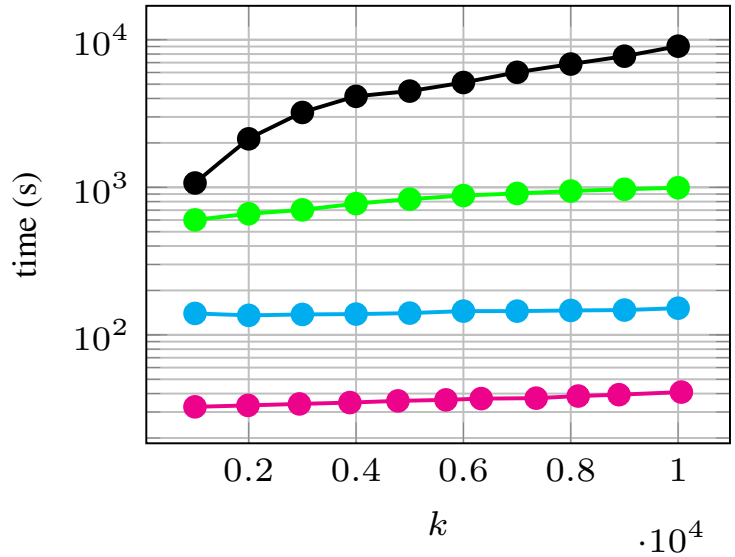
- AlexNet CNN fc7 features, PCA to 128 dimensions, optimized subspace decomposition [Ge *et al.* , 2013]

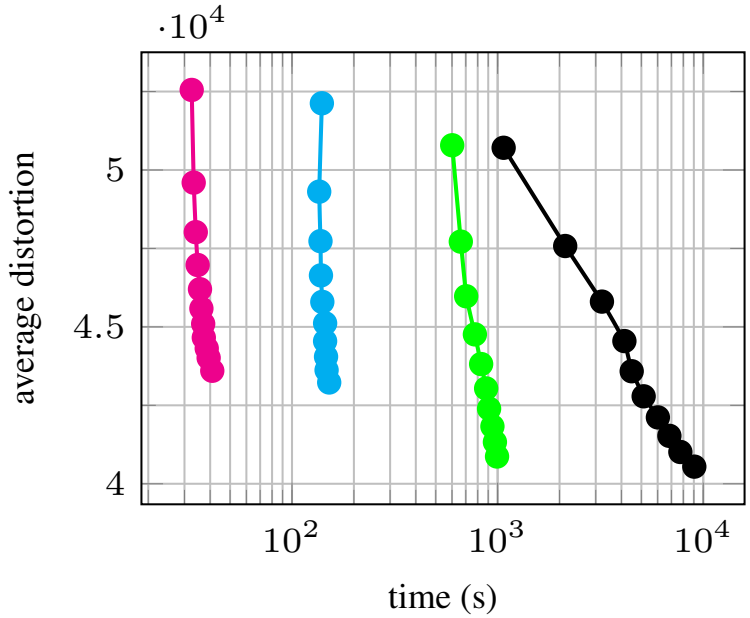
evaluation metrics

- distortion, timing, precision-recall (Paris)
- YFCC100M: cluster precision (or *purity*) on a noisy set of image classification labels (percentage of images that share top class label)

Results: SIFT1M







Results: YFCC100M

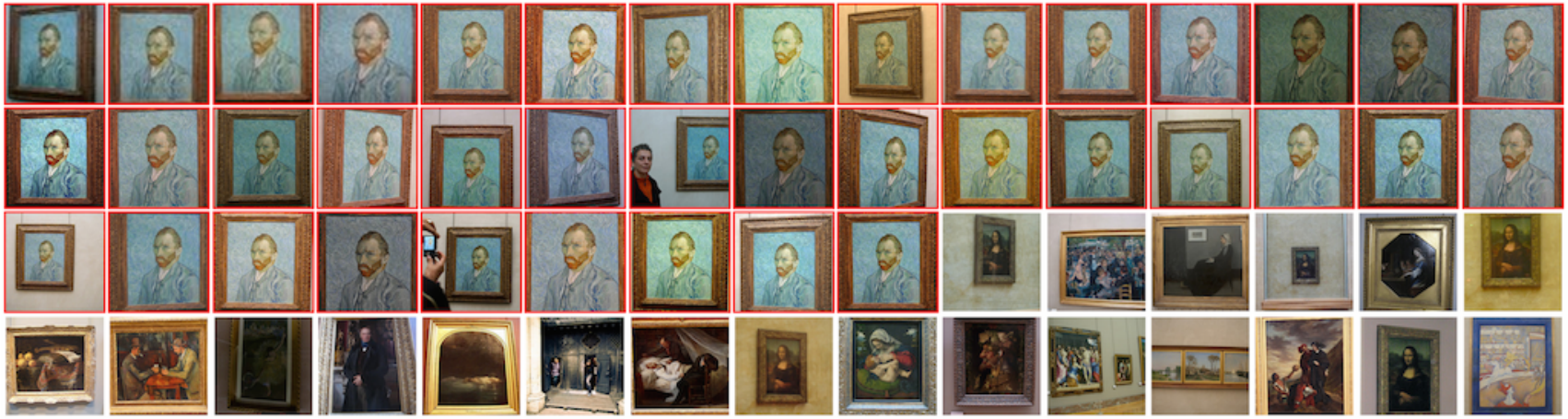
	CKM	distributed k-means (× 300)	dynamic IQ-means
k/k'	100000	100000	85742
time (s)	13068.1	7920.0	**140.6**
precision	0.474	**0.616**	0.550

Table: time per iteration and average precision, initial $k = 10^5$, s=8192

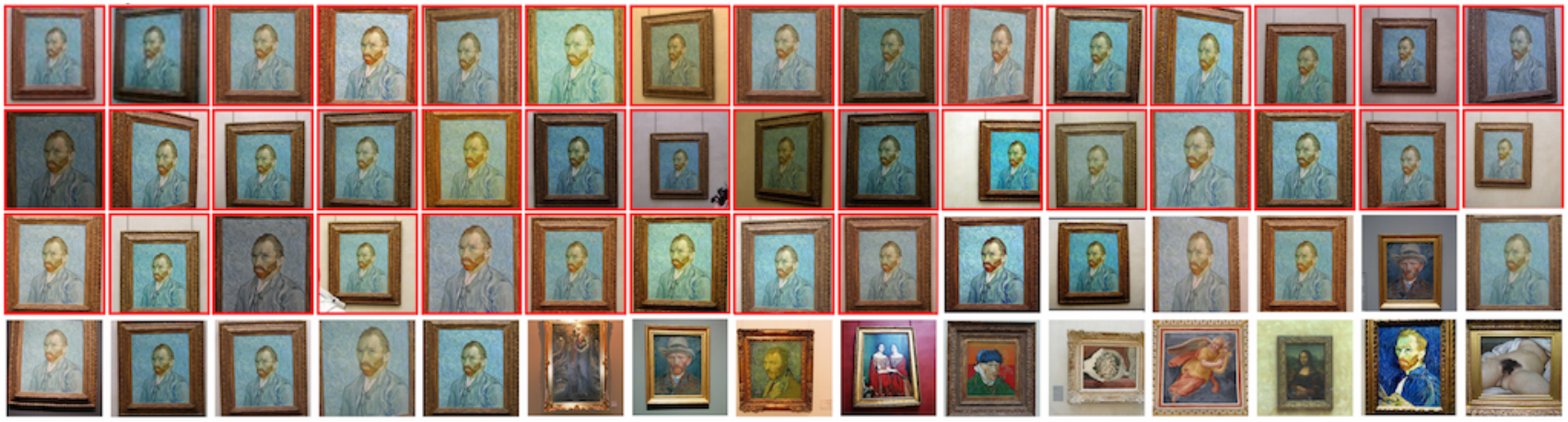
	IQ-M			D-IQ-M		
k/k'	100K	150K	200K	86K	120K	152K
time (s)	212.6	271.1	325.8	140.6	249.6	277.2

Table: time per iteration and k/k'

Mining example: Paris & YFCC100M



clustering on Paris



clustering on Paris & YFCC100M

Conclusions

IQ-means: a very fast k-means variant

- quantize points on a grid of two subspaces
- apply inverted search from centroids to cells
- dynamic estimation at nearly zero cost
- assignment step is faster than update step!

web-scale clustering

- extremely compressed data (26bits per image for YFCC100M)
- clustering of 100M images, on a single machine, in less than an hour
- results worse than using (costly) dedicated mining methods, but on par with much slower k-means variants

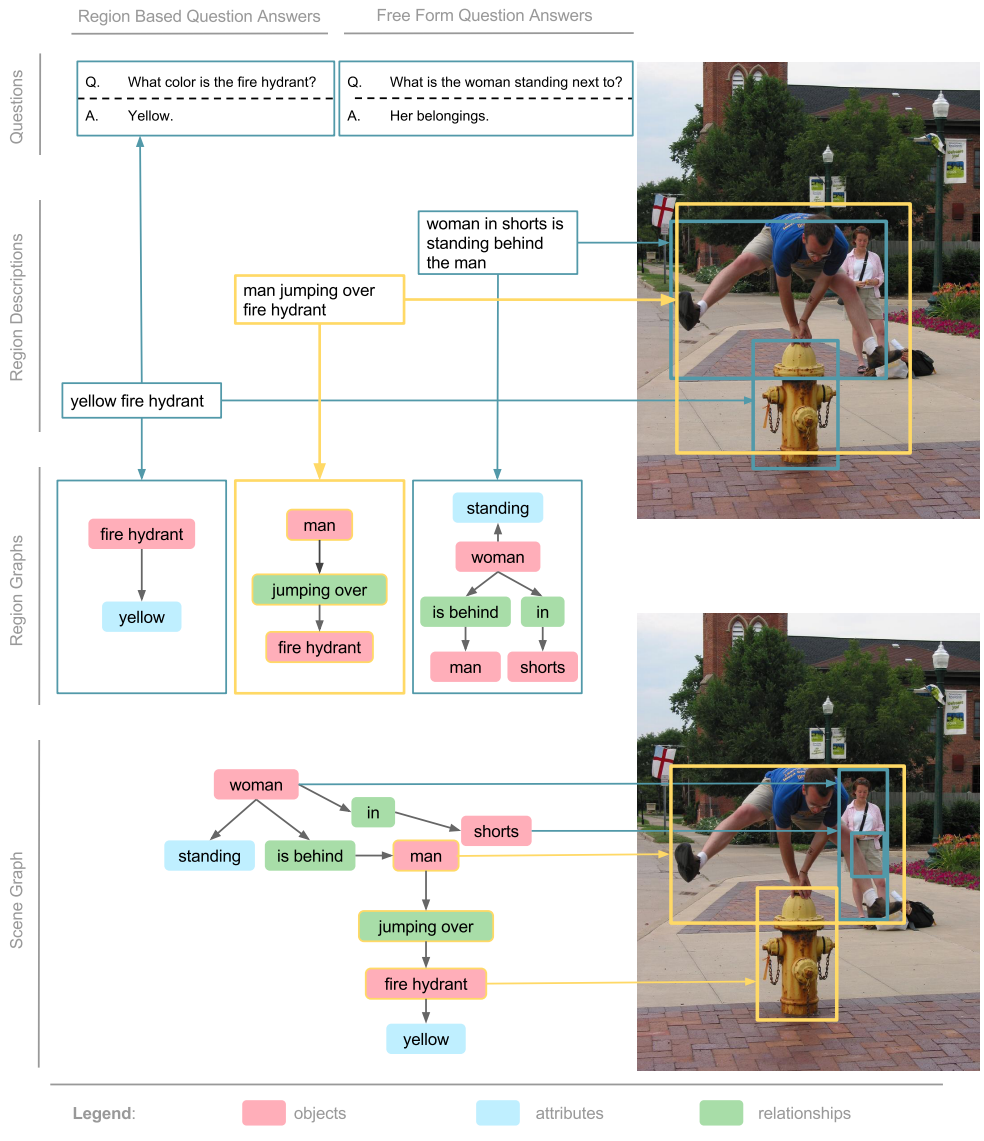
www.visualgenome.org

VISUALGENOME

An ongoing effort to connect structured image concepts to language.

R. Krishna, Y. Zhu, O. Groth, J. Johnson,
K. Hata, J. Kravitz, S. Chen, Y. Kalantidis,
L. Jia-Li, D. A. Shamma, M. S. Bernstein,
L. Fei-Fei

- 108,249 COCO images
- 4.2 million region descriptions
- 1.7 million visual questions and answers
- 2.1 million object instances
- 1.8 million attributes
- 1.8 million relationships
- Everything mapped to WordNet synsets



STANFORD COMPUTER SCIENCE

YAHOO! LABS

Source code on git:

`http://github.com/iavr/iqm`

Thank you!