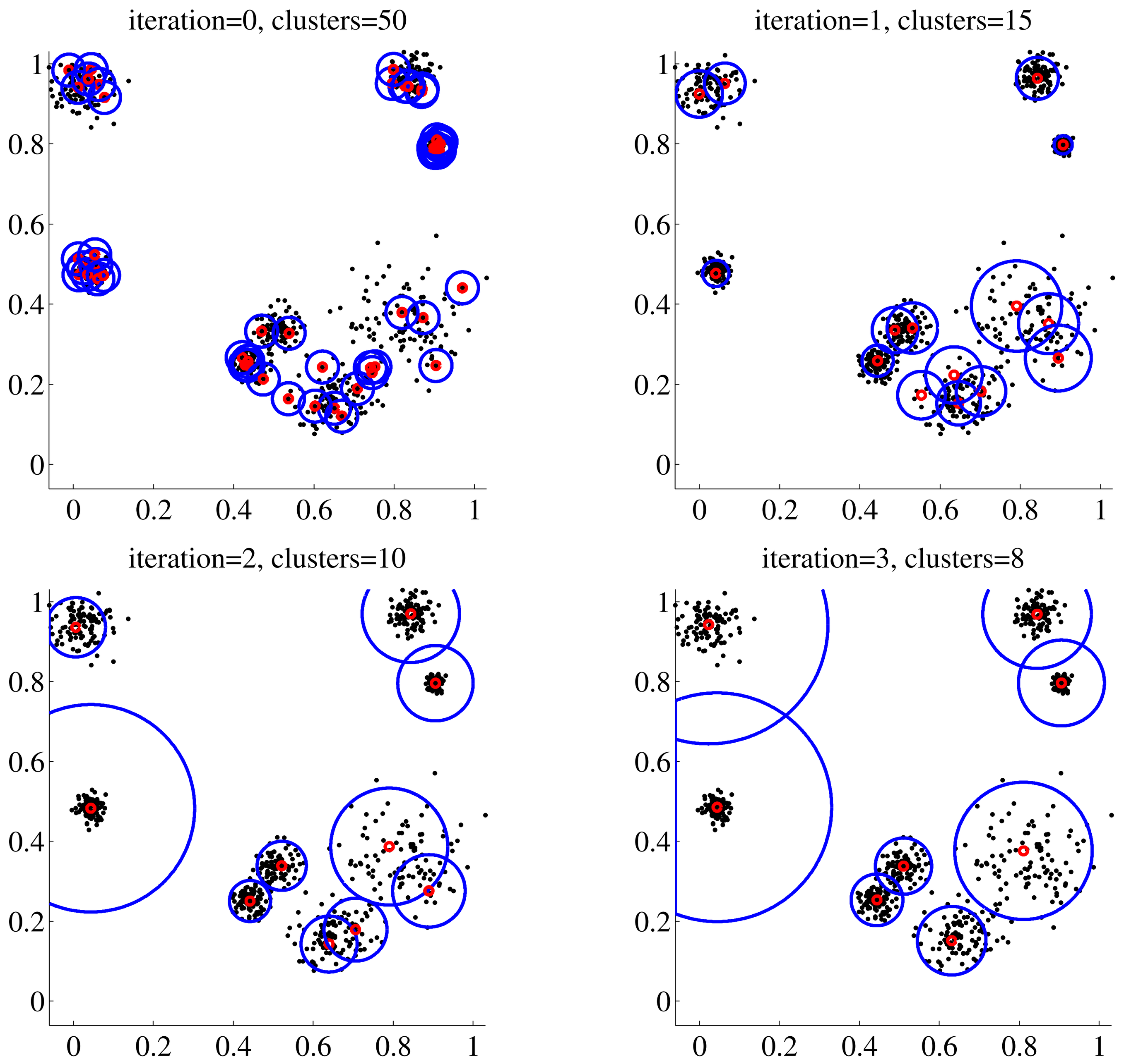
Approximate Gaussian Mixtures for Large Scale Vocabularies

Yannis Avrithis and Yannis Kalantidis
National Technical University of Athens

Expanding Gaussian mixtures

- 800 points from an 8-mode 2d Gaussian, initialized at 50 points.



Overview

- **Approximate Gaussian Mixtures (AGM):** a clustering method that combines the flexibility of Gaussian mixtures with the scaling properties needed to construct visual vocabularies for image retrieval. The algorithm can *dynamically* estimate the number of clusters.
- **Approximate:** Keep a fixed number m of nearest neighbors per data point across iterations so that we: (a) have enough information for an *approximate Gaussian mixture* model and (b) speed-up the nearest neighbor search process.
- **Purge:** Initialize with all data points as cluster centers and purge them as necessary using an overlap criterion on neighboring clusters.
- **Expand:** Clusters neighboring to the ones being purged expand towards empty space, boosting convergence rate.
- **Algorithm:** A modification of EM, where (a) a P-step (purge) is interleaved with the E- and M- steps at each iteration; (b) the E-step is approximate and incremental (N-step); (c) σ is over-estimated at the M-step (expand).

Purge

- If function q represents any component or cluster, we define the *generalized responsibility* $\hat{\gamma}_{ik} = \hat{\gamma}_k(p_i) \in [0,1]$ of component k for *component* i, similar to responsibility $\gamma_k(x)$ of k for point x:

$$\gamma_k(x) = \frac{p_k(x)}{\sum_{j=1}^K p_j(x)} \quad \rightarrow \quad \hat{\gamma}_k(q) = \frac{\langle q, p_k \rangle}{\sum_{j=1}^K \langle q, p_j \rangle},$$

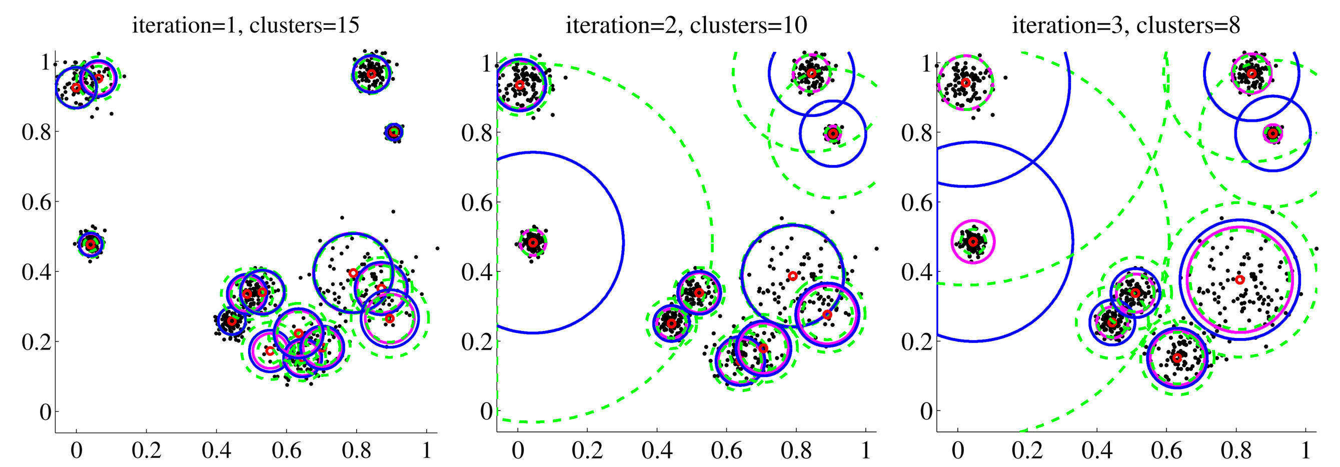
where $p_k(x) = \pi_k \mathcal{N}(x|\mu_k, \sigma_k^2)$ and the L^2 inner product $\langle p_i, p_k \rangle = \pi_i \pi_k \mathcal{N}(\boldsymbol{\mu}_i | \boldsymbol{\mu}_k, (\sigma_i^2 + \sigma_k^2)\mathbf{I})$ measures the *overlap* of components p_i, p_k under the spherical Gaussian model.

- If $\hat{\gamma}_{ii}$ is the responsibility of component i for *itself* and given a set $\mathcal{K}$ of components and one component $i \notin \mathcal{K}$, we define the responsibility $\rho_{i,\mathcal{K}} \in [0,1]$ of component i for itself *relative to* $\mathcal{K}$ as

$$\rho_{i,\mathcal{K}} = \frac{\hat{\gamma}_{ii}}{\hat{\gamma}_{ii} + \sum_{j \in \mathcal{K}} \hat{\gamma}_{ij}} = \frac{\|p_i\|^2}{\|p_i\|^2 + \sum_{j \in \mathcal{K}} \langle p_i, p_j \rangle}.$$

- If $\rho_{i,\mathcal{K}}$ is large, component i can 'explain' itself better than set $\mathcal{K}$ *as a whole*; otherwise i appears to be redundant. So, if $\mathcal{K}$ represents the components we have decided to *keep* so far, it makes sense to purge component i if $\rho_{i,\mathcal{K}}$ drops below *overlap threshold* $\tau \in [0,1]$.

Expand



- We *overestimate* the extent of each component as much as this does not overlap with its neighboring components.
- The re-estimation equation for the covariance of each component can be decomposed into $D\sigma_k^2 = \frac{\underline{N}_k}{N_k}\underline{\Sigma}_k + \frac{\overline{N}_k}{N_k}\overline{\Sigma}_k$ where the *inner sum* $\underline{\Sigma}_k$ expresses a weighted average distance from $\boldsymbol{\mu}_k$ of data points that are better 'explained' by component k, hence fits the underlying data of the corresponding cluster.
- We *bias* the weighted sum towards the *outer sum* $\overline{\Sigma}_k$, and the re-estimation equation becomes $D\sigma_k^2 = w_k \underline{\Sigma}_k + (1 - w_k)\overline{\Sigma}_k$, where $w_k = \frac{\underline{N}_k}{N_k}(1 - \lambda)$ and $\lambda \in [0,1]$ is an *expansion factor*.

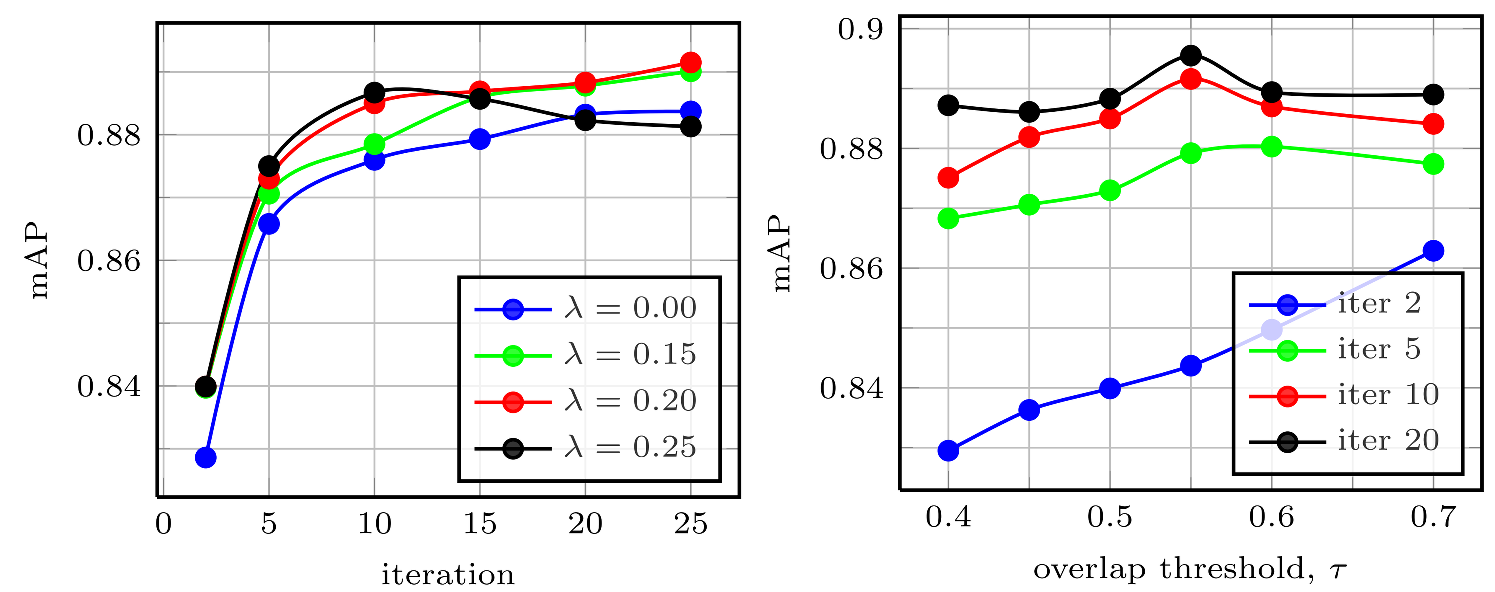
Approximate Gaussian mixtures

Algorithm 2: Incremental m-nearest neighbors (N-step)

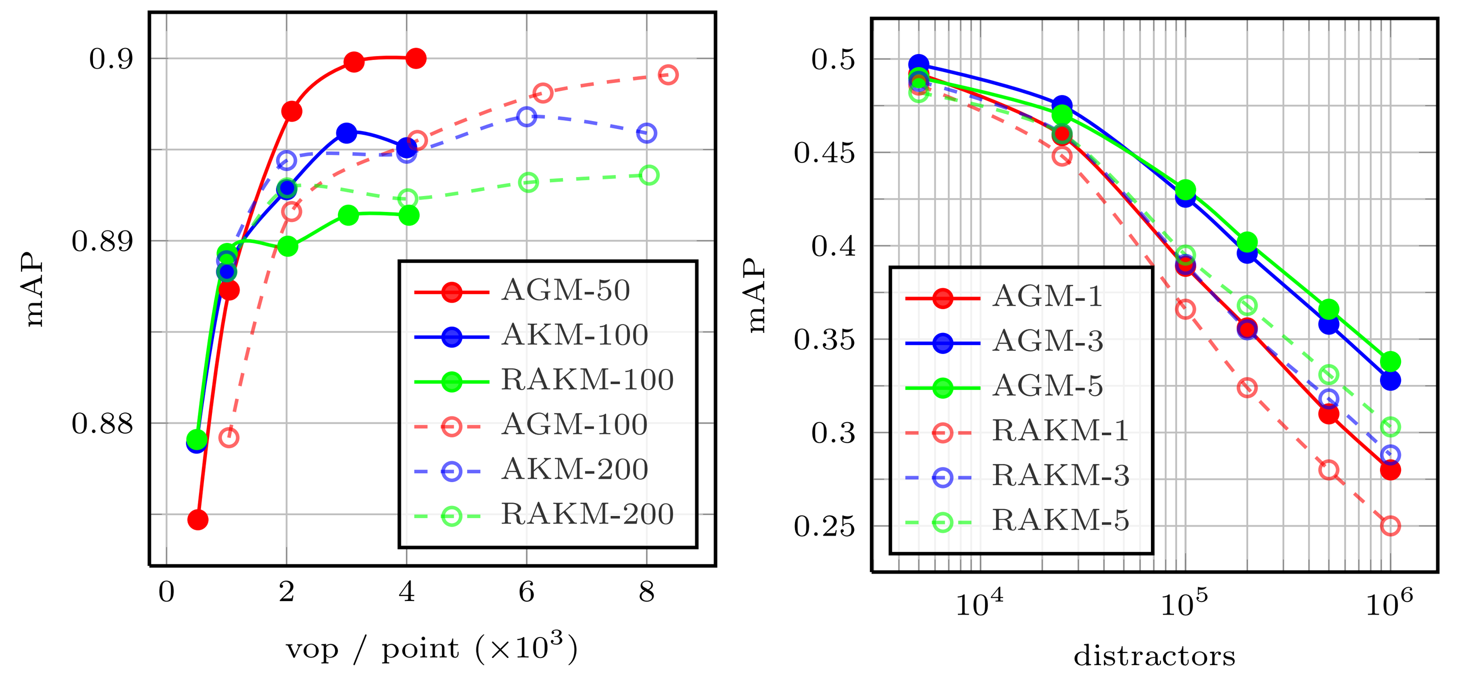
```
input : m best neighbors B(x_n) found so far for n = 1,...,N
output: updated m best neighbors B'(x_n) for n = 1,...,N

for n = 1,...,N do                                  // for all data points
    B(x_n) ← C ∩ B(x_n)                             // ignore purged neighbors
    (R, d) ← NN_m(x_n)                              // R: m-NN of x_n; d: distances to x_n
                                                    // (such that d_k^2 = ||x_n − μ_k||^2 for k ∈ R)
    for k ∈ B(x_n) \ R do                           // for all previous neighbors...
        d_k^2 ← ||x_n − μ_k||^2                     // ...find distance after μ_k update (M-step)
    A ← B(x_n) ∪ R                                  // for all previous and new neighbors...
    for k ∈ A do                                    // ...compute unnormalized...
        g_k ← (π_k/σ_k^D) exp{−d_k^2/(2σ_k^2)}      // ...responsibility of k for x_n
    SORT A such that i < k → g_i ≥ g_k for i,k ∈ A  // keep the top-ranking...
    B'(x_n) ← A[1,...,m]                            // ...m neighbors
```

Retrieval experiments

- **Datasets:** Oxford Buildings and World Cities (WC).
- **Tuning:** Specific vocabulary on *Barcelona dataset*–$550K$ SURF descriptors.



- **Large scale experiment:** Generic vocabulary from $6.5M$ descriptors on Oxford dataset + 1M distractors from WC.



Project page & code: http://image.ntua.gr/iva/research/agm/ Contact: {ykalant, iavr}@image.ntua.gr