Exercise for MA-INF 2201 Computer Vision WS15/16
23.11.2015
Submission on 30.11.2015

Expectation Maximization and Submodularity

1. The foreground of image `gnome.png` is bounded by the rectangle `Rect(92,65,105,296)`. Model the foreground as GMM_{fg} and the background as GMM_{bg} using Gaussian Mixture Models (K=10 mixtures). Display the image that visualizes the following probability for each image pixel:

$$\frac{P(x|GMM_{fg})}{P(x|GMM_{fg}) + P(x|GMM_{bg})}$$

   1.1. Solve the task with the use of `cv::EM`. *(3 Points)*

   1.2. Solve the task with your own custom implementation of Gaussian Mixture Models and Expectation Maximization (EM). To this end,

      1.2.1. Use `cv::kmeans` to initialize the EM algorithm. *(*3 points)

      1.2.2. Implement the EM algorithm. Initialize mixture weights as $1/K$ and check for maximum number of iterations for termination. *(*7 points)

   Note: Use 5–20 iterations according to the speed of your implementation. A single iteration refers to the execution of both the *E-step* and *M-step*.

2. A function is *submodular* when it satisfies the equation:

$$P(\beta, \gamma) + P(\alpha, \delta) - P(\beta, \delta) - P(\alpha, \gamma) \geq 0$$

   for all $\alpha, \beta, \gamma, \delta$ such that $\beta > \alpha$ and $\delta > \gamma$. Show that:

   2.1. the *Quadratic Function* $P(\omega_m, \omega_n) = c(\omega_m - \omega_n)^2$ is *submodular*, *(*2 points)

   2.2. the *Potts model* $P(\omega_m, \omega_n) = c(1 - \delta(\omega_m - \omega_n))$ is not *submodular*, by providing a counter-example to the above criterion. *(*2 points)

3. Provide a graph structure using the *alpha expansion* method that encodes the initial state of 6 nodes (a,b,c,d,e,f) with initial states $\beta\beta\gamma\alpha\alpha\gamma$ for the case where the label α is expanded. *(*3 points)