Synopsis

1 -Title of project:
2 - Name and study ID of all group members:

-
- Jacob Johansen - s103808

3 - Background and motivation:
4 - Milestones:

- Implement CNN network and get descent accuracy on non corrupted Urban8k data
-

This exercise is based on C. M. Bishop: *Pattern Recognition and Machine Learning* Chapter 9. Print and comment on the figures produced by the software as outlined below at the **Checkpoints**.

Density Estimation

We observe a stochastic multi-channel d-dimensional signal $\mathbf{x}$ and our aim is to model the density $p(\mathbf{x}) \sim p(\mathbf{x}|\mathbf{w})$, where the family $p(\mathbf{x}|\mathbf{w})$ is a given parametric density. The training set is $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_N\}$ and the likelihood function for $\mathbf{w}$ is given by $p(\mathcal{D}|\mathbf{w}) = \prod_{n=1}^{N} p(\mathbf{x}_n|\mathbf{w})$. The cost function is minus the log likelihood:

$$E(\mathbf{w}) = \sum_{n=1}^{N} -\log p(\mathbf{x}_n|\mathbf{w}).$$

The *test error* of a density model can be estimated by evaluating the cost function on a test set.

The mixture of Gaussians model family is defined as

$$p(\mathbf{x}|\mathbf{w}) = \sum_{k=1}^{K} p(\mathbf{x}|\mathbf{w}_k)\pi_k \ ,$$

where each component density is a normal distribution $\mathbf{w}_k = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$. Here we will invoke a family of "isotropic" Gaussians, i.e., Gaussians with covariance matrices that are scaled unit matrices,

$$p(\mathbf{x}|\boldsymbol{\mu}_k, \sigma_k^2) \ = \ \frac{1}{(2\pi\sigma_k^2)^{d/2}} \exp\left(-\frac{||\mathbf{x} - \boldsymbol{\mu}_k||^2}{2\sigma_k^2}\right) \ .$$

The Expectation-Maximization algorithm is a general scheme for maximum likelihood estimation. For the mixture of Gaussians it leads to the iterative procedure discussed in Bishop page 438-439. In the E-step, the means, variances and mixing proportions are fixed and we update all responsibilities $\gamma_{nk} \equiv p(k|\mathbf{x}_n)$, $n = 1, \ldots, N$ and $k = 1, \ldots, K$ in parallel

$$\gamma_{nk} = p(k|\mathbf{x}_n) = \frac{p(\mathbf{x}|\mathbf{w}_k)\pi_k}{\sum_{k'=1}^{K} p(\mathbf{x}|\mathbf{w}_{k'})\pi_{k'}} \ .$$

In the M-step, the responsibilities are fixed and we update the means, variances and mixing proportions. We can write the update compactly by introducing N_k the effective number of points assigned to component k:

$$\begin{aligned} N_k &= \sum_{n=1}^{N} \gamma_{nk} \\ \boldsymbol{\mu}_k^{\text{new}} &= \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} \mathbf{x}_n \\ (\sigma_k^{\text{new}})^2 &= \frac{1}{dN_k} \sum_{n=1}^{N} \gamma_{nk} ||\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}||^2 \\ \pi_k^{\text{new}} &= \frac{N_k}{N} \ . \end{aligned}$$

The rules for updating $\boldsymbol{\mu}_k$ and σ_k^2 are very similar to the well known rules for computing mean and variance of a normal distribution, but here weighted by the probability γ_{nk} that a given data point $\mathbf{x}_n$ belongs to mixture component k.

The *K-means* clustering algorithm is an important simplification of these rules obtained by using each data point only once, namely for updating the Gaussian component which it is most associated with and by letting all the component variances be equal (Bishop pages 424-430, 438-439). In the context of the K-means algorithm a component is often referred to as a *cluster*.

Checkpoint 7.1

Use the program `main7a.m` to perform K-means analysis on synthetic two-dimensional data with three clusters. The program creates two plots. The first shows the training points and the current position of the cluster centers as time progress. You can zoom to inspect the details of the convergence. The second figure shows the assignment of points to clusters at the end of the procedure. Is the final configuration sensitive to how the initial cluster centers are located? You can change the distribution of the initial clusters by changing the parameter `initial-width` and you can change the number of clusters `K`. Sometimes you will see that a cluster never moves away from its initial position, why?

Answer:

If the variance of the distribution of the clusters become to large some clusters may start 'too far away' and thus never be assigned any points. This causes the cluster to never move. This happens more the more clusters we initialize.

Checkpoint 7.2

The program `main7b.m` uses the Expectation-Maximization algorithm to adapt a Gaussian mixture, as described above. The program shows three plots. The first plot shows the training (blue) and test (yellow) points, the second shows the temporal evolution of the variance parameters and the training and test errors. Create a flow chart of the program `main7b.m` and its functions. The test error is the mean negative log likelihood (the cost function) estimated on the test set. Change the number of clusters $K = 2, 3, 4, 5, 6$ and

inspect the temporal evolution and final value of the test error. Explain how the Gaussian mixture can over-fit.
Answer:

Checkpoint 7.3

In `main7b.m` you can choose three different strategies for initializing the cluster centers and initial variances. Set $K = 5$ and run the program with the three different schemes. Describe the strategies and their results. You will see a problem with the Gaussian mixture and the EM algorithm for `method 3`, where one component converge towards a very small variance and is centered on a specific data point. Explain why this is a serious overfit.

Challenge I (not part of the curriculum)

Modify the EM mixture of Gaussians program so that instead of the isotropic Gaussian full covariances $\mathbf{\Sigma}_k$ is estimated. Visualise the one standard deviation contour of each of the components of the mixture. Hint: the points can created by $\mathbf{\Sigma}_k^{\frac{1}{2}}\mathbf{u}(t) + \boldsymbol{\mu}_k$, where $\mathbf{u}(t) = \begin{bmatrix} \cos(t) \\ \sin(t) \end{bmatrix}$ is taken as the points on the unit circle: $t \in [0, 2\pi]$. The square root of the covariance matrix can be found using the `sqrtm`-function in Matlab. We will use these results in the Challenge to Exercise 8.

Challenge II (not part of the curriculum)

Implement the K-mediods algorithm. Run it one the Gaussian simulated data using the Manhatten norm (sum of absolute difference for each dimension) as the cost funtion.

Challenge III (not part of the curriculum)

Explain why K-means cannot be seen as a special limit of the mixture of Gaussians, that is there is not a specific setting of the covariance matrix where K-means and mixture of Gaussians give exactly the same updates in both the E- and M-steps.

DTU, 1999 Lars Kai Hansen (2007 Ole Winther)