Regression with Gaussian Mixture Networks

In this document, we provide technical details about the discriminative model P^d and its optimisation. Recall that $P^d(\vec{X}_k|\vec{X}_k \neq 0)$ is a Gaussian Mixture probability distribution in $\mathbb{R}^2$, where the weight of each Gaussian is predicted by a Neural Network.

We therefore need to learn the parameters of the probability distribution

$$P^d(\vec{X}_k|\vec{X}_k \neq 0) = \sum_{1 \leq m \leq M} f_{\mathrm{h}}(\mathcal{F}^c_k; \theta_{\mathrm{h}})_m \mathcal{N}(\vec{X}_k - \alpha_m; \sigma_m) \ , \tag{1}$$

namely, the Gaussian parameters α_m and σ_m for each mode index m, and the network parameters θ_{h}.

Following Eq.17 of the main paper, we treat each pixel as an independent data-point $\vec{x}_s$. $S = S(Z^0, \ldots, Z^D)$ denotes the set of those data-points. In this section, we assume that we have access to a label in $\mathbb{R}^2$, for each data-point $\vec{x}_s$. We recall that this label is generated by sampling of the generative model, given *ground truth* detections Z.

The procedure that we use to optimise the following loss derived from Eq.17

$$\mathcal{R}(\theta_{\mathrm{h}}, \alpha, \sigma) = - \sum_{(\vec{x}_s \in S} \log(P^d(\vec{x}_s|\mathcal{F}^{c_s}_{k_s}, \theta_{\mathrm{h}}, \alpha, \sigma)) \ , \tag{2}$$

follows the same principles as the standard Gaussian Mixture regression model via Expectation-Maximization algorithm [1] and is also closely related to the recent Neural Decision Forests [2], which introduce a Network producing a probability distribution in the form of a mixture of Histograms.

Updating the Network Parameters We update the parameters of the network θ_{h} by direct back-propagation and stochastic gradient descent on the objective of Eq. 2.

Updating the Gaussian Parameters Let α^t and σ^t denote the current estimates of the the Gaussian parameters. We derive a closed form update which guarantees that

$$\mathcal{R}(\theta_{\mathrm{h}}, \alpha^{t+1}, \sigma^{t+1}) \leq \mathcal{R}(\theta_{\mathrm{h}}, \alpha^t, \sigma^t) \ . \tag{3}$$

For each data point $\vec{x}_s$, let us introduce the distribution over the mixture elements m,

$$\xi^t(m|\vec{x}_s, \mathcal{F}^{c_s}_{k_s}, \theta_{\mathrm{h}}, \alpha^t, \sigma^t) = \frac{f_{\mathrm{h}}(\mathcal{F}^{c_s}_{k_s}; \theta_{\mathrm{h}})_m \mathcal{N}(\vec{x}_s - \alpha^t_m; \sigma^t_m)}{\sum\limits_{1 \leq m' \leq M} f_{\mathrm{h}}(\mathcal{F}^{c_s}_{k_s}; \theta_{\mathrm{h}})_{m'} \mathcal{N}(\vec{x}_s - \alpha^t_{m'}; \sigma^t_{m'})} \tag{4}$$

usually called "responsibilities" in the GMM litterature.

We then use the standard variational trick with the auxiliary distribution $\xi^t(m)$ to minimise an upper-bound on $\mathcal{R}(\theta_{\text{h}}, \alpha, \sigma)$, with respect to the parameters α and σ.

$$
\begin{aligned}
\mathcal{R}(\theta_{\text{h}}, \alpha, \sigma) &= -\sum_{\vec{x}_s \in S} \log \left(\sum_{1 \le m \le M} f_{\text{h}}(\mathcal{F}_{k_s}^{c_s}; \theta_{\text{h}})_m \mathcal{N}(\vec{x}_s - \alpha_m; \sigma_m) \right) \\
&= -\sum_{\vec{x}_s \in S} \log \left(\sum_{1 \le m \le M} \xi^t(m|\vec{x}_s) \frac{f_{\text{h}}(\mathcal{F}_{k_s}^{c_s}; \theta_{\text{h}})_m \mathcal{N}(\vec{x}_s - \alpha_m; \sigma_m)}{\xi^t(m|\vec{x}_s)} \right) \\
&\le -\sum_{\vec{x}_s \in S} \sum_{1 \le m \le M} \xi^t(m|\vec{x}_s) \log \left(\frac{f_{\text{h}}(\mathcal{F}_{k_s}^{c_s}; \theta_{\text{h}})_m \mathcal{N}(\vec{x}_s - \alpha_m; \sigma_m)}{\xi^t(m|\vec{x}_s)} \right) \\
&\le \mathcal{R}(\theta_{\text{h}}, \alpha^t, \sigma^t) - \sum_{\vec{x}_s \in S} \sum_{1 \le m \le M} \xi^t(m|\vec{x}_s) \log \left(\frac{\mathcal{N}(\vec{x}_s - \alpha_m; \sigma_m)}{\mathcal{N}(\vec{x}_s - \alpha_m^t; \sigma_m^t)} \right)
\end{aligned} \tag{5}
$$

Minimizing Eq. 5 with respect to α and σ is a convex problem. Assuming that we can find the values achieving the minimum, let us set α^{t+1} and σ^{t+1} to these. Then, from Eq. 5, we obtain

$$
\mathcal{R}(\theta_{\text{h}}, \alpha^{t+1}, \sigma^{t+1}) \le \mathcal{R}(\theta_{\text{h}}, \alpha^t, \sigma^t) \ ,
$$

with equality if and only if $\alpha^{t+1} = \alpha^t$ and $\sigma^{t+1} = \sigma^t$.

We therefore need to minimize Eq. 5 with respect to α and σ, which is equivalent to maximizing

$$
\sum_{\vec{x}_s \in S} \sum_{1 \le m \le M} \xi^t(m|\vec{x}_s) \log \left(\mathcal{N}(\vec{x}_s - \alpha_m; \sigma_m) \right) \ ,
$$

with respect to the parameters α and σ. This is done by using the standard optimality conditions for convex problems. We obtain

$$
\alpha_m^{t+1} = \frac{\sum\limits_{\vec{x}_s \in S} \xi^t(m|\vec{x}_s) \vec{x}_s}{\sum\limits_{\vec{x}_s \in S} \xi^t(m|\vec{x}_s)} \ , \tag{6}
$$

and,

$$
\sigma_m^{t+1} = \frac{\sum\limits_{\vec{x}_s \in S} \xi^t(m|\vec{x}_s) (\vec{x}_s - \alpha_m^{t+1})^2}{\sum\limits_{\vec{x}_s \in S} \xi^t(m|\vec{x}_s)} \ . \tag{7}
$$

Alternating both In practice, we alternate one epoch of stochastic gradient descent optimizing the network parameters θ_{h} with one update of the Gaussian parameters of Eqs. 6 and 7. For memory usage reasons, the sums in Eqs. 6 and 7, have to be split into mini-batches. However the update is done after summation over the whole dataset or a very large number of samples.

References

[1] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. 1

[2] P. Kontschieder, M. Fiterau, A. Criminisi, and S. R. Bulo. Deep neural decision forests. In *International Conference on Computer Vision*, 2015. 1