Our approach

In this section, we give implementation details about the L2 regularization. Before we get start formally, we shall introduce the definition of L2 distance between two distributions. Then we will give the closed form of L2 distance between two gaussian mixture models. With these tools, we present our regularization term for the Mixture Density Network.

Definition of L2 distance Denote two random variables X and Y with respective support $R_X \in \mathbb{R}^d$ and $R_Y \in \mathbb{R}^d$. Suppose the pdf of X and Y are $f_X(\cdot)$ and $f_Y(\cdot)$ respectively. The L2 distance between X and Y, defined as

$$L_2(f_X, f_Y) = \left\{ \int_{\mathbb{R}^d} (f_X(x) - f_Y(x))^2 dx \right\}^{1/2} \tag{1}$$

is used to measure the distance between P and Q. L2 distance satisfies the following requirements:

1. $L_2(f_X, f_Y) \geq 0$ (non-negativity)
2. $L_2(f_X, f_Y) = 0$ if and only if X and Y has the same distribution
3. $L_2(f_X, f_Y) = L_2(f_Y, f_X)$ (symmetry)
4. $L_2(f_X, f_Z) \leq L_2(f_X, f_Y) + L_2(f_Y, f_Z)$, for any random variable Z supported in $\mathbb{R}^d$ with density function f_Z (subadditivity / triangle inequality).

Other well-studied distance metrics between random variables include Jensen-Shennon divergence, Wasserstein metric, maximum mean discrepancy, etc,. Among these metrics, the L2 distance is computationally desirable and is easy to implement, as we will show later.

L2 distance for GMM Suppose we have two mixtures of Gaussians

$$P(x) = \sum_{n=1}^{N} \alpha_n N\left(x \mid \mu_n, \Sigma_n\right) \text{ and } Q(x) = \sum_{m=1}^{M} \beta_m N\left(x \mid \eta_m, \Lambda_m\right) \tag{2}$$

where α_n and β_m are nonnegative (actually, positive) coefficients that sum to 1 respectively, i.e.. $\sum_{n=1}^{N} \alpha_n = 1$ and $\sum_{m=1}^{M} \beta_m = 1$. $N(\cdot \mid \mu, \Sigma)$ and $N(\cdot \mid \eta, \Lambda)$ are Gaussian distributions in $\mathbb{R}^d$ with mean
vectors μ, η and covariance matrices Σ, Λ respectively. Note that the density function for $N(x \mid \mu, \Sigma)$ is in form of

$$f(x) = \frac{1}{|\Sigma|^{1/2}(2\pi)^{d/2}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right) \tag{3}$$

The L2 distance between P and Q has an explicit formula, given by

$$L_2(P, Q) = \left\{ \sum_{n,n'} \alpha_n \alpha_{n'} A_{n,n'} + \sum_{m,m'} \beta_m \beta_{m'} B_{m,m'} - 2 \sum_{n,m} \alpha_n \beta_m C_{n,m} \right\}^{1/2}, \tag{4}$$

where

$$
\begin{aligned}
A_{n,n'} &= N\left(\mu_n \mid \mu_{n'}, \Sigma_n + \Sigma_{n'}\right) \\
B_{m,m'} &= N\left(\eta_m \mid \eta_{m'}, \Lambda_m + \Lambda_{m'}\right) \\
C_{n,m} &= N\left(\mu_n \mid \eta_m, \Sigma_n + \Lambda_m\right).
\end{aligned}
$$

For the derivation steps to get this result, we refer to http://kyoustat.com/pdf/note004gmml2.pdf.

Regularization for Mixture Density Network

Recall that for any condition $\boldsymbol{x}_0 \in \mathbb{R}^{d_x}$, the conditional density estimate $\hat{p}(\boldsymbol{y} \mid \boldsymbol{x}_0)$ suggested by the Mixture Density Network is

$$
\hat{p}(\boldsymbol{y} \mid \boldsymbol{x}_0) = \sum_{k=1}^{K} w_k(\boldsymbol{x}_0; \theta) \mathcal{N}\left(\boldsymbol{y} \mid \mu_k(\boldsymbol{x}_0; \theta), \sigma_k^2(\boldsymbol{x}_0; \theta)\right), \tag{5}
$$

where $w_k(\boldsymbol{x}_0; \theta)$, $\mu_k(\boldsymbol{x}_0; \theta)$ and $\sigma_k^2(\boldsymbol{x}_0; \theta)$ are the output of Mixture Density Network. For notation simplicity, we shall denote $\hat{p}(\cdot \mid \boldsymbol{x}_0)$ as $P_{\boldsymbol{x}_0}(\cdot)$.

Intuitively, for another condition $\boldsymbol{x}_1$ that is close to $\boldsymbol{x}_0$ (for example, $||\boldsymbol{x}_0 - \boldsymbol{x}_1|| < \epsilon$, where ϵ is a positive constant close to zero), we expect that $P_{\boldsymbol{x}_1}$ is close to $P_{\boldsymbol{x}_0}$, that is $L_2(P_{\boldsymbol{x}_1}, P_{\boldsymbol{x}_2})$ is small enough. If we define $D_{\boldsymbol{x}_0}(\boldsymbol{x}) \triangleq L_2(P_{\boldsymbol{x}}, P_{\boldsymbol{x}_0})$, we hope $D_{\boldsymbol{x}_0}(\boldsymbol{x})$ takes value close to zero when $\boldsymbol{x} < \epsilon$. Such intuition can be implemented by penalize (minimize) $||\nabla_{\boldsymbol{x}} D_{\boldsymbol{x}_0}(\boldsymbol{x})|_{\boldsymbol{x}=\boldsymbol{0}}||$. To interpret this penalization, the term $\nabla_{\boldsymbol{x}} D_{\boldsymbol{x}_0}(\boldsymbol{x})$ represents the gradient of $D_{\boldsymbol{x}_0}$ with respect to $\boldsymbol{x}$, and is a m-dimension vector. $\nabla_{\boldsymbol{x}} D_{\boldsymbol{x}_0}(\boldsymbol{x})|_{\boldsymbol{x}=\boldsymbol{0}}$ is the gradient of $D_{\boldsymbol{x}_0}$ with respect to $\boldsymbol{x}$ when $\boldsymbol{x}$ takes value as $\boldsymbol{0}$. Finally, $||\nabla_{\boldsymbol{x}} D_{\boldsymbol{x}_0}(\boldsymbol{x})|_{\boldsymbol{x}=\boldsymbol{0}}||$ is the L2 norm of the gradient vector $\nabla_{\boldsymbol{x}} D_{\boldsymbol{x}_0}(\boldsymbol{x})|_{\boldsymbol{x}=\boldsymbol{0}}$.

By penalize $||\nabla_{\boldsymbol{x}} D_{\boldsymbol{x}_0}(\boldsymbol{x})|_{\boldsymbol{x}=\boldsymbol{0}}||$, we can encourage the Mixture Density Network gives "smooth" conditional distribution at $\boldsymbol{x}_0$. To make the Mixture Density Network being able to give smooth conditional distribution at all the conditions, we add

$$
\lambda \cdot \mathbb{E}_{\boldsymbol{x}_0 \sim \mathcal{P}_{\boldsymbol{x}}}\left(||\nabla_{\boldsymbol{x}} D_{\boldsymbol{x}_0}(\boldsymbol{x})|_{\boldsymbol{x}=\boldsymbol{x}_0}||\right) \tag{6}
$$

to the objective function, where λ is a positive constant hyper-parameter which controls the weight of this regularization term.

Experiments

In this section, we present how our approach can improve Mixture Density Network on both synthetic dataset and real-world dataset. In particular, we simulate data from a 4 -dimensional Gaussian Mixture $(d_x = 2, d_y = 2)$ and a Skew-Normal distribution whose parameters are functionally dependent on x $(d_x = 1, d_y = 1)$. In terms of real-world data, we use the following three data sources. **EuroStoxx**: Daily returns of the Euro Stoxx 50 index conditioned on various stock return factors. **NYC Taxi**: Dropoff locations of Manhattan taxi trips conditioned on the pickup location, weekday and time. **UCI datasets**: Boston Housing, Concrete and Energy datasets from the UCI

machine learning repository (Dua & Graff, 2017). For each dataset, we use 70% as the training set and 30% as the test set. The hyper-parameter λ is tuned with cross validation. The reported scores are test log-likelihoods, averaged over at least 5 random seeds alongside the respective standard deviation. For further details regarding the data sets, network structure and hyper parameter tuning, we refer to the appendix.

Results for synthetic dataset

For each synthetic dataset, we simulate 1000 samples. The parameters of the Gaussian Mixture and Skew-Normal distribution are discussed in detail in appendix. The 'MDN without regularization' row represents the test log-likelihoods given by the original MDN, and the 'MDN with our regularization' row gives the test log-likelihoods given by the MDN with our regularization approach. The 'True model' row is the log-likelihoods computed by the true model on the test set. A larger log-likelihood indicates a more accurate model fitting. Observe that MDN with our regularization approach can compete MDN without regularization for both synthetic dataset, and get log-likelihood close to true model.

	Gaussian Mixture	**Skew-Normal**
MDN without regularization	-5.99±2.45	-3.99±0.66
MDN with our regularization	-2.48±0.11	-3.12±0.14
True model	-3.12±0.39	-3.03±0.13

Results for real-world dataset

For real-world dataset, the true model is unknown, so we only report test log-likelihood for MDN with and without regularization. MDN with our regularization gets larger test-loglikelihood for all the datasets.

	Euro Stoxx	**NCY Taxi**	**Boston**	**Concrete**	**Energy**
MDN without regularization	3.26±0.43	5.08±0.03	-3.46±0.47	-3.19±0.21	-1.25±0.23
MDN with our regularization	**3.94±0.03**	**5.25±0.04**	**-2.49±0.11**	**-2.92±0.08**	**-1.04±0.09**