Conditional Mixture Models and Mixture of Experts

Piyush Rai

Topics in Probabilistic Modeling and Inference (CS698X)

Feb 15, 2018

Conditional Mixture Models

- Standard mixture model are of the form $p(y_n|\Theta) = \sum_{k=1}^{K} \pi_k p(y_n|\theta_k)$
- Given data of the form $(\boldsymbol{x}_n, y_n)$, can model y_n conditioned on "inputs" $\boldsymbol{x}_n$ as a mixture model

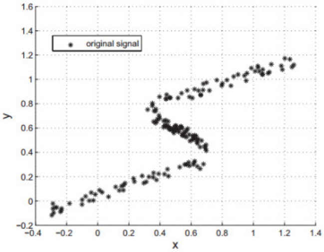
$$p(y_n|\Theta, \boldsymbol{x}_n) = \sum_{k=1}^{K} \pi_k p(y_n|\theta_k, \boldsymbol{x}_n)$$

- Can also assume that π_k too depends on $\boldsymbol{x}_n$. One way is to model it via a softmax

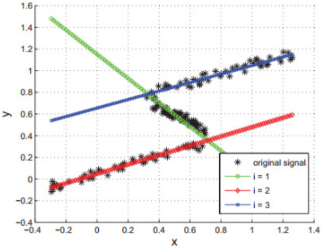
$$\pi_k(\boldsymbol{x}_n) = \frac{\exp(\eta_k^\top \boldsymbol{x}_n)}{\sum_{\ell=1}^{K} \exp(\eta_\ell^\top \boldsymbol{x}_n)} \qquad \text{(now } \pi_k \text{ a function of } \boldsymbol{x}_n \text{ and } \boldsymbol{\eta} = [\eta_1, \ldots, \eta_K])$$

- Called conditional mixture models since the mixture model for y_n is conditioned on inputs $\boldsymbol{x}_n$
- Such modeled as referred to by various other names too, e.g.,
  - Covariate-dependent mixture model: Since the mixture model depends on the inputs/covariates $\boldsymbol{x}_n$
  - Density Regression: Since we are doing density estimation for y_n by also "regressing" on $\boldsymbol{x}_n$
  - Mixture of Experts (MoE): Since the model can be seen as combining K experts $\{p(y_n|\theta_k, \boldsymbol{x}_n)\}_{k=1}^{K}$
- The forms of the K experts $\{p(y_n|\theta_k, \boldsymbol{x}_n)\}_{k=1}^{K}$ depends on the type of y_n (real? binary? count?)

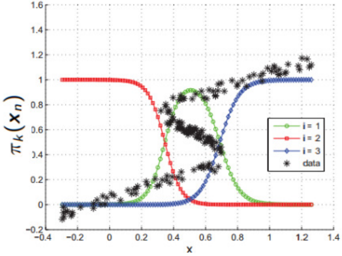
Mixture of Experts: An Illustration



Original Data



Three Linear Experts



Expert's Mixing Weights
(probabilities for each input)

- A nice way to construct powerful supervised learning models (e.g., nonlinear regression)
- The construction can utilize simpler models as its components (e.g., linear regression)
- The input to expert assignments are "soft" in nature (as probabilities)
- Also has a "divide and conquer" flavor (e.g., cluster data; learn a regression model in each cluster)

Mixture of Experts as Latent Variable Models

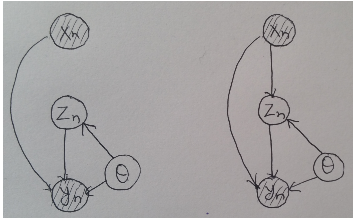
- As we saw, the MoE models are of the form $p(y_n|\Theta, \boldsymbol{x}_n) = \sum_{k=1}^{K} \pi_k p(y_n|\theta_k, \boldsymbol{x}_n)$
- Just like the standard mixture models, can think of MoE as a latent variable model
- Suppose $\boldsymbol{z}_n \in \{1, \ldots, K\}$ denotes which expert "generates" y_n
- Can write the density $p(y_n|\Theta, \boldsymbol{x}_n)$ as a marginal of the joint distribution $p(y_n, \boldsymbol{z}_n|\Theta, \boldsymbol{x}_n)$

$$p(y_n|\Theta, \boldsymbol{x}_n) = \sum_{\boldsymbol{z}_n} p(y_n, \boldsymbol{z}_n|\Theta, \boldsymbol{x}_n) = \sum_{k=1}^{K} p(\boldsymbol{z}_n = k|\Theta, \boldsymbol{x}_n) p(y_n|\boldsymbol{z}_n = k, \Theta, \boldsymbol{x}_n)$$

- In the EM language, $p(y_n|\Theta, \boldsymbol{x}_n)$ is incomplete data log-lik, $p(y_n, \boldsymbol{z}_n|\Theta, \boldsymbol{x}_n)$ is complete data log-lik
- Prior prob. of y_n generated by expert k, i.e., $p(\boldsymbol{z}_n = k|\Theta, \boldsymbol{x}_n)$ is modeled by a "gating function"
  - Usually it depends on $\boldsymbol{x}_n$ but in some MoE models it doesn't, i.e., $p(\boldsymbol{z}_n = k|\Theta, \boldsymbol{x}_n) = p(\boldsymbol{z}_n|\Theta)$
  - If it depends on $\boldsymbol{x}_n$, the gating network is basically a multiclass classifier from $\boldsymbol{x}_n$ to latent $\boldsymbol{z}_n$
  - In the simplest setting, $p(\boldsymbol{z}_n = k|\Theta, \boldsymbol{x}_n) = \frac{1}{K}$, i.e., *a priori*, all experts equally likely to generate y_n

Mixture of Experts as LVM

- The figure below illustrates two MoE architectures



- Left figure: $p(\boldsymbol{z}_n = k|\Theta, \boldsymbol{x}_n)$ doesn't depend on $\boldsymbol{x}_n$
- Right figure: $p(\boldsymbol{z}_n = k|\Theta, \boldsymbol{x}_n)$ depends on $\boldsymbol{x}_n$
- Note: When conditioned on y_n, the posterior of $\boldsymbol{z}_n$, i.e., $p(\boldsymbol{z}_n = k|y_n, \Theta, \boldsymbol{x}_n)$ DOES depend on $\boldsymbol{x}_n$
  - Reason: The likelihood $p(y_n|\boldsymbol{z}_n = k, \Theta, \boldsymbol{x}_n)$ depends on $\boldsymbol{x}_n$

EM for Mixture of Experts

- EM is easy to derive for MoE (though fully Bayesian inference also possible)
- The derivation proceeds very similarly to standard mixture models
- As usual, MLE using $\log p(y_n|\Theta, \boldsymbol{x}_n)$ will be hard since $p(y_n|\Theta, \boldsymbol{x}_n)$ doesn't have a simple form

$$p(y_n|\Theta, \boldsymbol{x}_n) = \sum_{k=1}^{K} p(\boldsymbol{z}_n = k|\Theta, \boldsymbol{x}_n)p(y_n|\boldsymbol{z}_n = k, \Theta, \boldsymbol{x}_n) = \sum_{k=1}^{K} \pi_k(\boldsymbol{x}_n)p(y_n|\theta_k, \boldsymbol{x}_n)$$

- With LVM formulation, can do EM, i.e., MLE on expected CLL $\mathbb{E}[\log p(y_n, \boldsymbol{z}_n|\Theta, \boldsymbol{x}_n)]$
- Assume linear regression experts $\mathcal{N}(y_n|\boldsymbol{w}_k^\top \boldsymbol{x}_n, \beta^{-1})$, easy to show that the overall CLL is

$$\log p(\boldsymbol{y}, \mathbf{Z}|\Theta, \mathbf{X}) = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk} \log\left\{\pi_k(\boldsymbol{x}_n)\mathcal{N}(y_n|\boldsymbol{w}_k^\top \boldsymbol{x}_n, \beta^{-1})\right\} \quad \text{(verify; did it for GMM)}$$

where $\pi_k(\boldsymbol{x}_n) = \frac{\exp(\eta_k^\top \boldsymbol{x}_n)}{\sum_{\ell=1}^{K}\exp(\eta_\ell^\top \boldsymbol{x}_n)}$, and $\Theta = \{\{\eta_k, \boldsymbol{w}_k\}_{k=1}^K, \beta\}$ are parameters (estimated in M step)

EM for Mixture of Experts

- In the E step, we will estimate the posterior $p(\boldsymbol{z}_n|\Theta, \boldsymbol{x}_n, y_n)$

$$p(\boldsymbol{z}_n = k|\Theta, \boldsymbol{x}_n, y_n) = \frac{\pi_k(\boldsymbol{x}_n) \times \mathcal{N}(y_n|\boldsymbol{w}_k^\top \boldsymbol{x}_n, \beta^{-1})}{\sum_{\ell=1}^K \pi_\ell(\boldsymbol{x}_n) \times \mathcal{N}(y_n|\boldsymbol{w}_\ell^\top \boldsymbol{x}_n, \beta^{-1})}$$

- Using the one-hot notation, the above is the same as $p(z_{nk} = 1|\Theta, \boldsymbol{x}_n, y_n) = \mathbb{E}[z_{nk}] = \gamma_{nk}$ (say)
- The M step maximizes the expected CLL (w.r.t. Θ) which is simply

$$\begin{aligned}
\mathbb{E}[\log p(\boldsymbol{y}, \mathbf{Z}|\Theta, \mathbf{X})] &= \sum_{n=1}^N \sum_{k=1}^K \mathbb{E}[z_{nk}] \log \left\{\pi_k(\boldsymbol{x}_n)\mathcal{N}(y_n|\boldsymbol{w}_k^\top \boldsymbol{x}_n, \beta^{-1})\right\} \\
&= \sum_{n=1}^N \sum_{k=1}^K \gamma_{nk}[\log \pi_k(\boldsymbol{x}_n) + \log \mathcal{N}(y_n|\boldsymbol{w}_k^\top \boldsymbol{x}_n, \beta^{-1})] \\
&= \sum_{n=1}^N \sum_{k=1}^K \gamma_{nk} \left[\log \frac{\exp(\eta_k^\top \boldsymbol{x}_n)}{\sum_{\ell=1}^K \exp(\eta_\ell^\top \boldsymbol{x}_n)} + \log \mathcal{N}(y_n|\boldsymbol{w}_k^\top \boldsymbol{x}_n, \beta^{-1})\right]
\end{aligned}$$

- Solving for each $\boldsymbol{w}_k$ and β is like doing MLE for weighted probabilistic linear regression

$$\boldsymbol{w}_k = (\mathbf{X}^\top \mathbf{S}_k \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{S}_k \boldsymbol{y} \qquad \text{and} \qquad \frac{1}{\beta} = \frac{1}{N} \sum_{n=1}^N \sum_{k=1}^K \gamma_{nk}(y_n - \boldsymbol{w}_k^\top \boldsymbol{x}_n)^2$$

where $\mathbf{S}_k = \text{diag}(\gamma_{1k}, \ldots, \gamma_{Nk})$

EM for Mixture of Experts

- Also need to estimate the parameters η_k for the "softmax" gating network $\pi_k(\boldsymbol{x}_n) = \frac{\exp(\eta_k^\top \boldsymbol{x}_n)}{\sum_{\ell=1}^K \exp(\eta_\ell^\top \boldsymbol{x}_n)}$
- This would require maximzing the expected CLL w.r.t. η_k

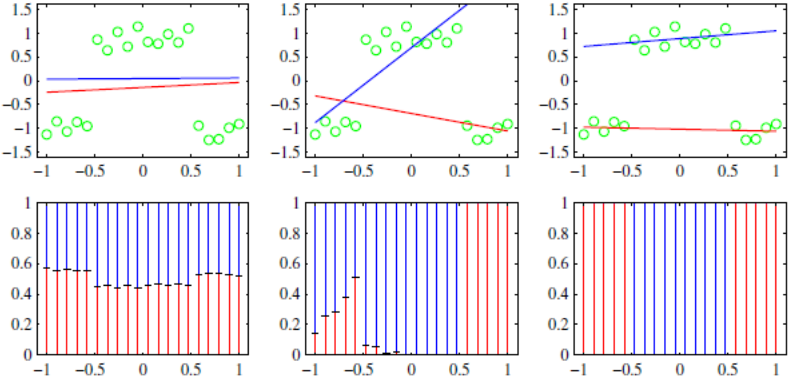
$$\mathbb{E}[\log p(\boldsymbol{y}, \mathbf{Z}|\Theta, \mathbf{X})] = \sum_{n=1}^N \sum_{k=1}^K \gamma_{nk} \left[\log \frac{\exp(\eta_k^\top \boldsymbol{x}_n)}{\sum_{\ell=1}^K \exp(\eta_\ell^\top \boldsymbol{x}_n)} + \underbrace{\log \mathcal{N}(y_n | \boldsymbol{w}_k^\top \boldsymbol{x}_n, \beta^{-1})}_{\text{can ignore, no } \eta_k \text{ in it}} \right]$$

- Can't get a closed form solution for η_k (due to the softmax coupling)
  - Iterative solver used for softmax regression can be used here (e.g., iterative reweighted least squares)
  - Some tricks exist to get closed form solutions using approximate inference methods
- "Softmax" gating network is just one possibility. Other types of gating networks can also be used
  - Note that the softmax gating is basically discriminative classifier from $\boldsymbol{x}_n$ to $\boldsymbol{z}_n \in \{1, \ldots, K\}$
  - A popular alternative is a generative gating network[1], i.e., $p(\boldsymbol{z}_n = k|\boldsymbol{x}_n) \propto p(\boldsymbol{z}_n = k)p(\boldsymbol{x}_n|\boldsymbol{z}_n = k)$. Will now need to learn the parameters defining $p(\boldsymbol{z}_n = k)$ and class-conditionals $p(\boldsymbol{x}_n|\boldsymbol{z}_n = k)$

[1] Twenty Years of Mixture of Experts, Yuksel et al (2012)

EM for Mixture of Experts: An Illustration

Left column: EM at initialization, Center column: EM after 30 iters, Right column: EM after 50 iters



(Figure courtesy: PRML)

Mixture of Experts - A Simple Special Case

- Assume no gating network, i.e. $p(\boldsymbol{z}_n|\Theta, \boldsymbol{x}_n) = \pi_k(\boldsymbol{x}_n) = \frac{1}{K}$ (each expert equally likely, a priori)
- Assume each expert to be a linear regression model
- In this case, the posterior distribution of input to expert assignment

$$p(\boldsymbol{z}_n = k|\Theta, \boldsymbol{x}_n, y_n) = \frac{\pi_k(\boldsymbol{x}_n)\mathcal{N}(y_n|\boldsymbol{w}_k^\top \boldsymbol{x}_n, \beta^{-1})}{\sum_{\ell=1}^{K} \pi_\ell(\boldsymbol{x}_n)\mathcal{N}(y_n|\boldsymbol{w}_\ell^\top \boldsymbol{x}_n, \beta^{-1})} \propto \mathcal{N}(y_n|\boldsymbol{w}_k^\top \boldsymbol{x}_n, \beta^{-1}) \propto \exp\left(-\frac{\beta}{2}||y_n - \boldsymbol{w}_k^\top \boldsymbol{x}_n||^2\right)$$

- We can thus define a hard assignment of input to the best expert as $\hat{\boldsymbol{z}}_n = \arg\min_k ||y_n - \boldsymbol{w}_k^\top \boldsymbol{x}_n||^2$
- This leads to a simple "mixed regression model". Can be learned via a K-means style algo
  - Initialize the K regression weights (K experts) $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K$
  - Repeat until convergence..
    - For each example $(\boldsymbol{x}_n, y_n)$, select the current best expert $\hat{\boldsymbol{z}}_n = \arg\min_k ||y_n - \boldsymbol{w}_k^\top \boldsymbol{x}_n||^2$
    - Learn each expert $\boldsymbol{w}_k$ using examples for which $\hat{\boldsymbol{z}}_n = k$, i.e.,

$$\boldsymbol{w}_k = \arg\min \sum_{n:\hat{\boldsymbol{z}}_n = k} ||y_n - \boldsymbol{w}_k^\top \boldsymbol{x}_n||^2$$

Mixture of Experts vs Other Nonlinear Models

- Nonlinear learning can also be done using kernel methods (GPs in probabilistic setting)
- Some of the benefits of MoE over kernel methods
  - Usually faster to train (no need to work with kernel matrices)
  - Faster at the time. Simply $p(y_n|\Theta, \boldsymbol{x}_n) = \sum_{k=1}^{K} \pi_k(\boldsymbol{x}_n) p(y_n|\theta_k, \boldsymbol{x}_n)$ (compare this with GP prediction)
  - Usually more interpretable (nonlinear model as a mixture of many linear models)
  - Simple plug-and-play architecture (can choose from a variety of gating functions and experts models)

$$p(y_n|\Theta, \boldsymbol{x}_n) = \sum_{k=1}^{K} \underbrace{\pi_k(\boldsymbol{x}_n)}_{\text{gating fn}} \underbrace{p(y_n|\theta_k, \boldsymbol{x}_n)}_{\text{an expert}}$$

- Some of the disadvantages of MoE over kernel methods
  - Training requires care (EM can be sensitive to local optima)
  - Number of experts need to be specified (this can be learned using nonparam. Bayesian methods)
  - Experts needs to be a prob. model (though most reg/class. models anyway have a prob. formulation)

Mixture of Experts: Summary

- Flexible framework for learning powerful models by combining simple probabilistic models
- Illustrates the "modular" nature of probabilistic modeling (ease of model composition)
- Similar in spirit to ensemble methods such as boosting, bagging, etc.
  - But much more general in nature (and has a probabilistic/Bayesian formulation)
  - Input-dependent combination of experts
- A fairly old (rather "classic") model (from early 90s) but still fairly relevant
- Can even use nonlinear models (e.g., deep neural nets, GPs, etc) as experts
  - Can solve regression as well as classification (for classification, each expert is a classification model)
- Gating networks can also be nonlinear (basically any multiclass classification model)
- Also used recently to speed-up very large deep neural networks
  - Helps deep NNs to scale by exploiting the idea of "conditional computation"
  - Conditional computation: Only a part of the whole model is "active" for a given input
  - See "Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer" (ICLR 2017)