Ensemble Learning - 4
Matriochka models

John Klein

centralelille

Matriochka models

image pixabay.com

- A smaller model into a larger model into a larger one...
- Mixture models
- Hierarchical Bayesian models

Mixture models
definition

- A mixture model is a convex combination of posterior (predictive) distributions $p_{Y|X;\boldsymbol{\theta}_i}(y|\mathbf{x})$:

$$p_{Y|X;\boldsymbol{\Theta}}(y|\mathbf{x}) = \sum_{m=1}^{M} \pi_m p_{Y|X;\boldsymbol{\theta}_m}(y|\mathbf{x})$$

- The combined distribution depends on the sources parameters $\boldsymbol{\theta}_i$ and the combination parameters π_m :

$$\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M, \pi_1, \ldots, \pi_M\}.$$

- We have $\sum_{i=1}^{M} \pi_m = 1$ and $\pi_m \geq 0, \forall m$.

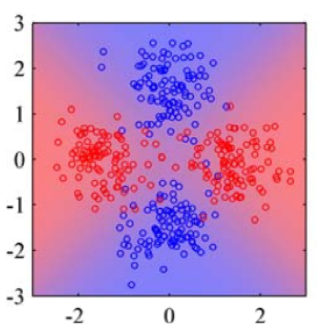
Mixture models
definition

- Each sub-model is probabilistic :

$$\hat{f}_m(\mathbf{x}) = p_{Y|X;\boldsymbol{\theta}_m}(y|\mathbf{x}).$$

- $\arg\max_y p_{Y|X;\boldsymbol{\Theta}}(y|\mathbf{x})$ is a weighted vote of the sub-model predictions $\left\{\arg\max_y f_m(\mathbf{x})\right\}_{m=1}^{M}$.
- All parameters can be jointly learned from $\mathcal{D}_{\text{train}}$ using EM.
- This is in contrast with LOPs which train each f_m on (a smaller) $\mathcal{D}_{\text{train}}$ and learn parameter $(\pi_m)_{m=1}^{M}$ on $\mathcal{D}_{\text{val}}$.
- Non-probabilistic techniques (SVM, k-NN) cannot be assembled in this way.

- Impossible to fit a linear model to this dataset :



- Let's try to fit a mixture of 2 logistic regressions.

- Outputs y are binary variables $\{0;1\}$ and inputs $\mathbf{x}$ are vectors in $\mathbb{R}^d$.

- The posterior is :

$$\begin{aligned} p_{Y|X;\boldsymbol{\Theta}}(y|\mathbf{x}) &= \pi_1 (1-\hat{y}_1)^{1-y}\,\hat{y}_1^y + \pi_2 (1-\hat{y}_2)^{1-y}\,\hat{y}_2^y, \\ &= \begin{cases} \pi_1\hat{y}_1 + \pi_2\hat{y}_2 & \text{if } y = 1 \\ \pi_1(1-\hat{y}_1) + \pi_2(1-\hat{y}_2) & \text{if } y = 0 \end{cases}, \end{aligned}$$

with $\hat{y}_1 = \mathrm{sgm}\left(\boldsymbol{\theta}_1^T \cdot \mathbf{x}_+\right)$ and $\hat{y}_2 = \mathrm{sgm}\left(\boldsymbol{\theta}_2^T \cdot \mathbf{x}_+\right)$ the logistic outputs and

$$\mathbf{x}_+^{(i)} = \begin{bmatrix} x_1^{(i)} \\ \vdots \\ x_d^{(i)} \\ 1 \end{bmatrix}$$

- The likelihood is

$$
\begin{aligned}
\mathcal{L}(\boldsymbol{\Theta}) &= p(\mathrm{data}|\boldsymbol{\Theta}), \\
&= \prod_{i=1}^{n} p(\mathrm{datum\ number\ } i|\boldsymbol{\Theta}), \\
&= \prod_{i=1}^{n} p_{Y|X=\mathbf{x}^{(i)};\boldsymbol{\Theta}}\left(y^{(i)}\right), \\
&= \prod_{i=1}^{n} \pi_1\left(1-\hat{y}_1^{(i)}\right)^{1-y^{(i)}}\left(\hat{y}_1^{(i)}\right)^{y^{(i)}} + \pi_2\left(1-\hat{y}_2^{(i)}\right)^{1-y^{(i)}}\left(\hat{y}_2^{(i)}\right)^{y^{(i)}}.
\end{aligned}
$$

- Let us introduce latent variables $z^{(i)} \in \{1;2\}$ standing for the fact that example $\mathbf{x}^{(i)}$ was generated by mixture component number.
- $z^{(i)} \sim \mathrm{Ber}\left(\pi_2\right) : P\left(z^{(i)}=1\right)=\pi_1$ and $P\left(z^{(i)}=2\right)=\pi_2=1-\pi_1$.
- The complete data[1] likelihood is then :

$$
\begin{aligned}
\mathcal{L}_{\text{comp}}\left(\boldsymbol{\Theta}\right) &= \prod_{i=1}^{n} p\left(y^{(i)}, z^{(i)} | \mathbf{x}^{(i)}, \boldsymbol{\Theta}\right), \\
&= \prod_{i=1}^{n} \prod_{k=1}^{2} \left(\pi_k \left(1-\hat{y}_k^{(i)}\right)^{1-y^{(i)}} \left(\hat{y}_k^{(i)}\right)^{y^{(i)}}\right)^{\mathbb{1}_k\left(z^{(i)}\right)} .
\end{aligned}
$$

- A fake multiplication appears because now each point is concerned with only one component of the mixture !

1. hidden and observed data

- E step : one can show that

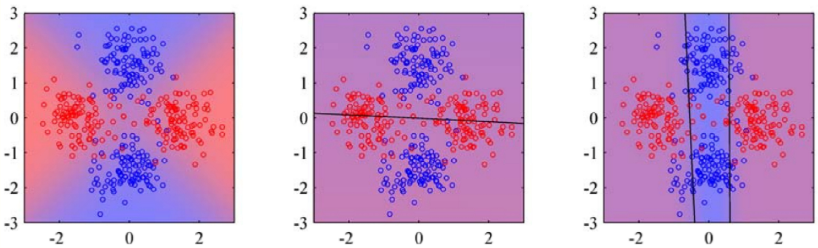
$$\mathbb{E}_{\substack{z^{(1)},\ldots,z^{(n)} \\ |\mathcal{D};\boldsymbol{\Theta}}}\left[\log \mathcal{L}_{\text{comp}}\left(\boldsymbol{\Theta}\right)\right] = \sum_{i=1}^{n}\sum_{k=1}^{2}\gamma_k^{(i)}\left[\log\left(\pi_k\right) + \left(1 - y^{(i)}\right)\log\left(1 - \hat{y}_k^{(i)}\right) + y^{(i)}\log\left(\hat{y}_k^{(i)}\right)\right],$$

$$\gamma_k^{(i)} = \frac{\pi_k\left(1-\hat{y}_k^{(i)}\right)^{1-y^{(i)}}\left(\hat{y}_k^{(i)}\right)^{y^{(i)}}}{\sum_{k'}\pi_{k'}\left(1-\hat{y}_{k'}^{(i)}\right)^{1-y^{(i)}}\left(\hat{y}_{k'}^{(i)}\right)^{y^{(i)}}}.$$

- M step : parameters $\boldsymbol{\theta}_i$ need to be estimated using a gradient ascent (Newton's method) while mixing weights are given by :

$$\pi_k = \frac{1}{n}\sum_{i=1}^{n}\gamma_k^{(i)}.$$

- The fit result is



- See [Bishop 14.5.2] for more details.

Mixture of Experts
definition

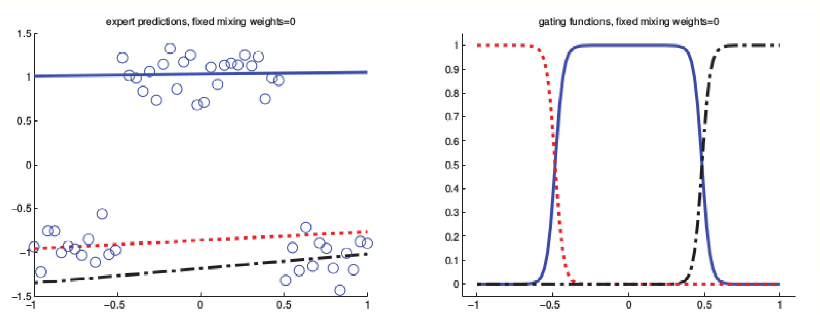
- The fact that sub-models f_m are functions of $\mathbf{x}$ is not exploited in the previous model.

- **Mixtures of experts** generalize this model by making mixing weights input-dependent :

$$\pi_k(\mathbf{x}) = \mathrm{smax}\left(\mathbf{V}^T \cdot \mathbf{x}\right).$$

- In this context, mixing weights are called **gating functions** and each f_m is called an **expert**.

- Obviously, a linear regression is a good model for subsets of the following data :



- The right figure shows trained gating functions for each regressor.

- E step : one can show that

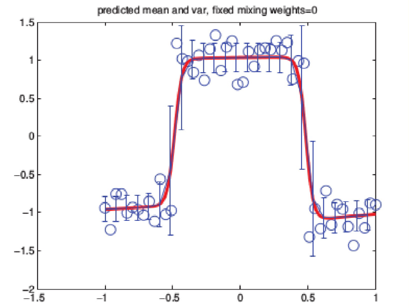
$$\mathbb{E}_{z^{(1)},\ldots,z^{(n)}}\left[\log \mathcal{L}_{\text{comp}}(\boldsymbol{\Theta})\right] = \sum_{i=1}^{n}\sum_{k=1}^{2}\gamma_k^{(i)}\left[\log\left(\pi_k^{(i)}\right) - \frac{\left(y^{(i)} - \boldsymbol{\theta}_k^T\cdot\mathbf{x}^{(i)}\right)^2}{2\sigma_k^2}\right],$$

$$\gamma_k^{(i)} = \frac{\pi_k^{(i)} \times \frac{1}{\sqrt{2\pi}\sigma_k} e^{\frac{\left(y^{(i)}-\boldsymbol{\theta}_k^T\cdot\mathbf{x}^{(i)}\right)^2}{2\sigma_k^2}}}{\sum_{k'}\pi_{k'}^{(i)} \times \frac{1}{\sqrt{2\pi}\sigma_{k'}} e^{\frac{\left(y^{(i)}-\boldsymbol{\theta}_{k'}^T\cdot\mathbf{x}^{(i)}\right)^2}{2\sigma_{k'}^2}}},$$

$$\pi_k^{(i)} = \mathrm{smax}\left(\mathbf{V}^T\cdot\mathbf{x}^{(i)}\right).$$

- M step : there is a closed-form MLE solution for parameters $\boldsymbol{\theta}_k$ and σ_k. $\mathbf{V}$ is estimated by gradient ascent (Newton's method).

- The fit result is



- See [Murphy 2012 - 11.4.3] for more details.

- A catch sentence for this chapter could be :

  « Why use only one classifier when I can use many ? »

- With Bayesian learning, this would become :

  « Why use only one classifier when I can use infinitely many ? »

- Let us see under which circumstances such a result can be achieved.

- Most of learning algorithms translate into an optimization problem of the following kind :

$$\operatorname*{arg\,min}_{\boldsymbol{\theta}} \quad \mathrm{DataFit}\,(\boldsymbol{\theta}) \quad + \quad \mathrm{Regularizer}\,(\boldsymbol{\theta})\,.$$

- In this setting, each $f \in \mathcal{H}$ is in bijective correspondence with a given $\boldsymbol{\theta} \in \boldsymbol{\Theta}$.

- Almost all such algorithms have an equivalent **probabilistic** formulation :

$$\operatorname*{arg\,max}_{\boldsymbol{\theta}} \quad \mathrm{Likelihood}\,(\boldsymbol{\theta}) \quad \times \quad \mathrm{Prior}\,(\boldsymbol{\theta})\,.$$

- Suppose we are trying to predict the **selling price** y of a house.

- For each house, we collected data like surface, previous buying price, GPS coordinates, etc.

- These features are concatenated into a vector $\mathbf{x}$;

- We need to learn the function f_0 mapping vectors $\mathbf{x}$ to y.

- We believe a **linear combination** of the features should be a relevant model :

$$y = \boldsymbol{\theta}^T \cdot \mathbf{x}.$$

- Yet we also believe that this **linear combination** is just an approximation of f_0 and therefore we go for a probabilistic formulation :

$$Y \sim \mathcal{N}\left(\boldsymbol{\theta}^T \cdot \mathbf{x}, \sigma\right).$$

- Now the **likelihood** is given by :

$$\text{Likelihood}\left(\boldsymbol{\theta}\right) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\left(y^{(i)} - \boldsymbol{\theta}^T \cdot \mathbf{x}^{(i)}\right)^2}{2\sigma^2}}$$

- For simplicity, we assume the noise variance σ^2 is known.

- We already have some beliefs on what values of $\boldsymbol{\theta}$ are more likely **before seeing any datum** :

$$\mathrm{Prior}\left(\boldsymbol{\theta}\right) = \frac{1}{(2\pi)^{\frac{d}{2}} \det\left(\mathbf{V}_0\right)^{\frac{1}{2}}} e^{-\frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\theta}_0)^T \cdot \mathbf{V}_0^{-1}(\boldsymbol{\theta}-\boldsymbol{\theta}_0)}$$

- **After seeing data** $\mathcal{D}$, our belief is given by the following posterior distribution

$$p\left(\boldsymbol{\theta}|\mathcal{D}, \boldsymbol{\theta}_0, \mathbf{V}_0\right) \quad \propto \quad \mathrm{Likelihood}\left(\boldsymbol{\theta}\right) \quad \times \quad \mathrm{Prior}\left(\boldsymbol{\theta}\right).$$

- If the prior parameters are such that $\boldsymbol{\theta}_0 = \mathbf{0}$ and $\mathbf{V}_0 = \tau^2 \mathbf{I}$, applying $-\log$ leads to the following cost function (up to an additive constant)

$$J(\boldsymbol{\theta}) \quad = \quad \underbrace{\sum_{i=1}^{n} \frac{\left(y^{(i)} - \boldsymbol{\theta}^T \cdot \mathbf{x}^{(i)}\right)^2}{2\sigma^2}}_{\text{Least Squares}} \quad + \quad \underbrace{\frac{1}{\tau^2} \|\boldsymbol{\theta}\|_2}_{\text{Ridge Reg.}}$$

- Going back to probabilities, one can show[2] that the posterior $p(\boldsymbol{\theta}|\mathcal{D}, \boldsymbol{\theta}_0, \mathbf{V}_0)$ is also Gaussian, in which case our prior is **conjugate**[3].

$$
\begin{aligned}
p(\boldsymbol{\theta}|\mathcal{D}, \boldsymbol{\theta}_0, \mathbf{V}_0) &\sim \mathcal{N}(\boldsymbol{\theta}_n, \mathbf{V}_n)\,, && (1)\\
\boldsymbol{\theta}_n &= \mathbf{V}_n \left(\mathbf{V}_0^{-1} \cdot \boldsymbol{\theta}_0 + \frac{1}{\sigma^2} \mathbf{X}^T \cdot \mathbf{y} \right), && (2)\\
\mathbf{V}_n &= \left(\mathbf{V}_0^{-1} + \frac{1}{\sigma^2} \mathbf{X}^T \cdot \mathbf{X} \right)^{-1}, && (3)
\end{aligned}
$$

$$
\text{with} \quad \mathbf{X} = \begin{pmatrix} \text{---} \left(\mathbf{x}^{(1)}\right)^T \text{---} \\ \vdots \\ \text{---} \left(\mathbf{x}^{(n)}\right)^T \text{---} \end{pmatrix} \quad \text{and} \quad \mathbf{y} = \begin{pmatrix} y^{(1)} \\ \vdots \\ y^{(n)} \end{pmatrix}.
$$

---

2. We assumed data are centered.
3. Under conjugacy, learning boils down to updating the prior parameters and the updates are easy to compute.

- No fusion for now .. just Bayesian statistics !
- As learners, what we really need is the **posterior predictive** $p(y|\mathbf{x}, \mathcal{D})$.
- The expectation of this distribution is our proxy for f_0 and allows to make a **prediction** for the selling price of a house whose features are the entries of the unseen example $\mathbf{x}$.

- Observe that the predictive distribution is **free of unobserved parameter conditioning**... because we marginalized them out :

$$p(y|\mathbf{x},\mathcal{D}) \;=\; \int_{\boldsymbol{\Theta}} p(y|\mathbf{x},\mathcal{D},\boldsymbol{\theta})\, p(\boldsymbol{\theta}|\mathbf{x},\mathcal{D})\, d\boldsymbol{\theta} \qquad (4)$$

- The above calculus is the **weighted combination** of an **infinity** of regressors !

- The weights depend on the ability of each regressor to fit well the data.

- In our linear regression case, we have

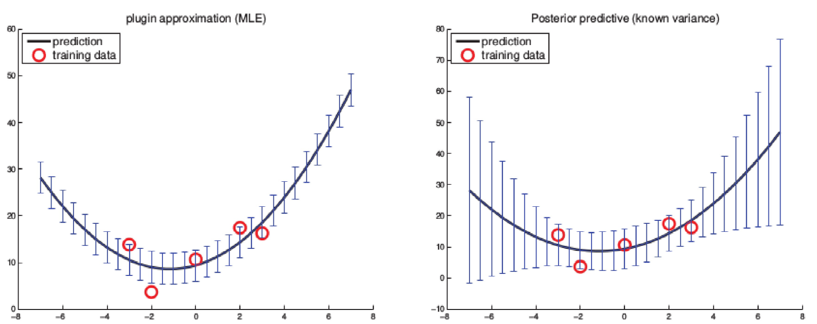
$$
\begin{aligned}
p\left(y|\mathbf{x},\mathcal{D}\right) &= \int_{\boldsymbol{\Theta}} p\left(y|\mathbf{x},\mathcal{D},\boldsymbol{\theta}\right) p\left(\boldsymbol{\theta}|\mathbf{x},\mathcal{D}\right) d\boldsymbol{\theta}, && (5)\\
&= \int_{\boldsymbol{\Theta}} p\left(y|\mathbf{x},\mathcal{D},\boldsymbol{\theta}\right) p\left(\boldsymbol{\theta}|\mathcal{D}\right) d\boldsymbol{\theta}, && (6)\\
&= \int_{\boldsymbol{\Theta}} G\left(y;\boldsymbol{\theta}^T\cdot\mathbf{x},\sigma^2\right) G\left(\boldsymbol{\theta};\boldsymbol{\theta}_n,\mathbf{V}_n\right) d\boldsymbol{\theta}, && (7)
\end{aligned}
$$

with G the Gaussian density function.

- Finally, one can show that

$$
\begin{aligned}
y|\mathbf{x},\mathcal{D} &\sim \mathcal{N}\left(\boldsymbol{\theta}_n^T\cdot\mathbf{x},\sigma_n\right), && (8)\\
\sigma_n^2 &= \sigma^2+\mathbf{x}^T\cdot\mathbf{V}_n\cdot\mathbf{x}. && (9)
\end{aligned}
$$

Illustration (polynomial reg.)



[Murphy 2012 - 7.6]

- The **posterior predictive** is not always known in closed form $\longrightarrow$ use **Monte-Carlo** to approximate the marginalization.

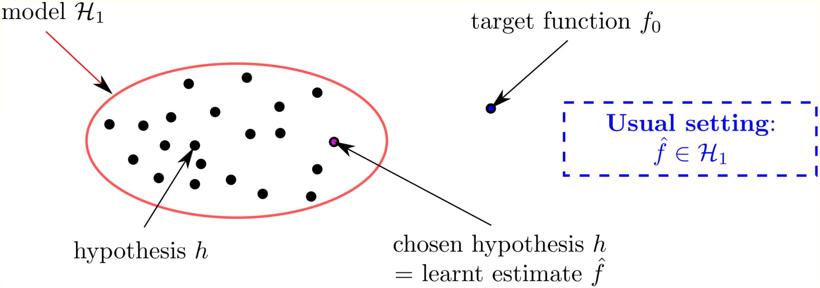
- Have we really gotten rid of all the parameters ?

- No, we are still conditioning w.r.t. $\boldsymbol{\theta}_0 = \mathbf{0}$ and $\mathbf{V}_0$.

- They can be marginalized out too by introducing a distribution for them called a **hyperprior**. This setting is known as **hierarchical Bayes**.

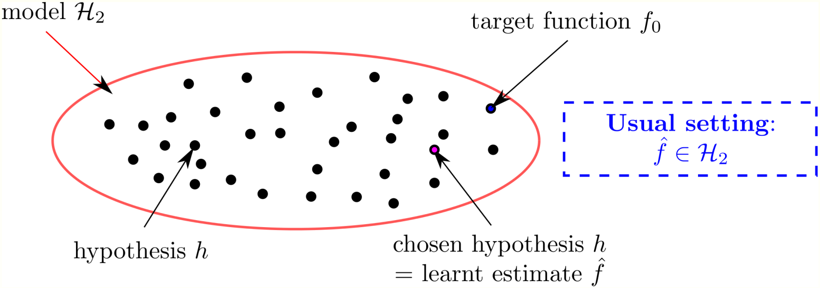
Let's start with model **selection**

Example : Polynomial regression with small degree $q = 1$



Let's start with model **selection**

Example : Polynomial regression with higher degree $q = 2$



Example : Polynomial regression with degree q

- In model **selection**, the candidate value for q is sought using, for example, CV.
  In general, it could be obtained as

$$q^* = \arg\max_{q \in \mathbb{N}^*} p\left(q|\mathcal{D}\right).$$

- In model **averaging**, several candidate values for q are considered. We are now writing the predictive posterior as

$$\begin{aligned} p\left(y|\mathbf{x}, \mathcal{D}\right) &= \sum_{q \in \mathbb{N}^*} p\left(y|\mathbf{x}, \mathcal{D}, q\right) p\left(q|\mathbf{x}, \mathcal{D}\right), \\ &= \sum_{q \in \mathbb{N}^*} p\left(y|\mathbf{x}, \mathcal{D}, q\right) p\left(q|\mathcal{D}\right). \end{aligned}$$

- Linear combination of the conditional predictive distributions $p\left(y|\mathbf{x}, \mathcal{D}, q\right)$.

- This will turn out to be a selection if I have enough data so that chances are concentrated on a given value q_* such that $p(q_*|\mathcal{D}) \approx 1$.

- In the **polynomial regression** setting, we have
$$p(y|\mathbf{x}, \mathcal{D}, q) = \mathcal{N}\left(\text{poly}_q(\mathbf{x}), \sigma^2\right).$$

- For each q, regression parameters $\boldsymbol{\theta}$ have been marginalized out using Bayesian learning.

- Given a **prior** $p(q)$ on polynomial degrees, we also have
$$p(q|\mathcal{D}) \propto p(\mathcal{D}|q)\, p(q).$$

- **BMA** only works for probabilistic models allowing to determine both $p(q|\mathcal{D})$ and $p(y|\mathbf{x}, \mathcal{D}, q)$.

- **BMA** will not select the best model (risk minimizer) if the true hypothesis f_0 is not one of the polynomials poly_q.
- Its philosophy is close to **hierarchical Bayes** in the sense that each hyperprior parameter choice can be regarded as a given model.
- Difference with a **mixture model** :

Mixture Model	**BMA**
1 model	Many models and one of them is the good one
The data is explained by multiple components	The data is explained by one of the model (This model might be itself a mixture model.)