Machine Learning/Advanced Machine Learning

Lecture 8.1:
Combining Models: Committee Machines

Sami S. Brandt

Department of Computer Science
IT University of Copenhagen

Based on Haykin, 1999, Bishop, 2006, and Alpaydin, 2014
21 October 2019

IT UNIVERSITY OF COPENHAGEN

Learning Objectives for Monday

- Reflect the principle of ensemble learning and apply it to combine multiple experts to improve performance
- Explain and implement boosting by filtering
- Implement and Apply AdaBoost to improve performance of weak learners with fixed number of training examples
- Explain the principles of the the mixture of experts model and hierarchical mixture of experts model

Outline of lecture

Bayesian Model Averaging

Ensemble Averaging

Boosting

Mixture of Experts

Hiearchial Mixture of Experts

Bayesian Model Averaging

- Bayesian model averaging refers to marginalising over different models
- Let $p(\mathbf{x}, \mathbf{z})$ be the joint distribution of observed variable $\mathbf{x}$ and a latent variable $\mathbf{z}$.
- The probability density of $\mathbf{x}$ is obtained by **marginalising** over the the joint distribution, or

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}).$$

- Example: Gaussian mixture model.
- Assume that the observed data $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ is i.i.d. and there are several possible models $h = 1, 2, \ldots, H$
- The **Bayesian Model Averaging** refers to considering the marginal

$$p(\mathbf{X}) = \sum_{h} p(\mathbf{X}|h)p(h)$$

where $p(h)$ is the prior probability of the model h.

Bayesian Model Averaging

- Bayesian model averaging refers to marginalising over different models
- Let $p(\mathbf{x}, \mathbf{z})$ be the joint distribution of observed variable $\mathbf{x}$ and a latent variable $\mathbf{z}$.
- The probability density of $\mathbf{x}$ is obtained by **marginalising** over the the joint distribution, or

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}).$$

Difference between Combining Models and Bayesian model averaging

- In Bayesian model averaging the data is generated by a *single* model
- In combining models, two data points can be generated by different components

- The **Bayesian Model Averaging** refers to considering the marginal

$$p(\mathbf{X}) = \sum_{h} p(\mathbf{X}|h)p(h)$$

where $p(h)$ is the prior probability of the model h.

Outline of lecture

Bayesian Model Averaging

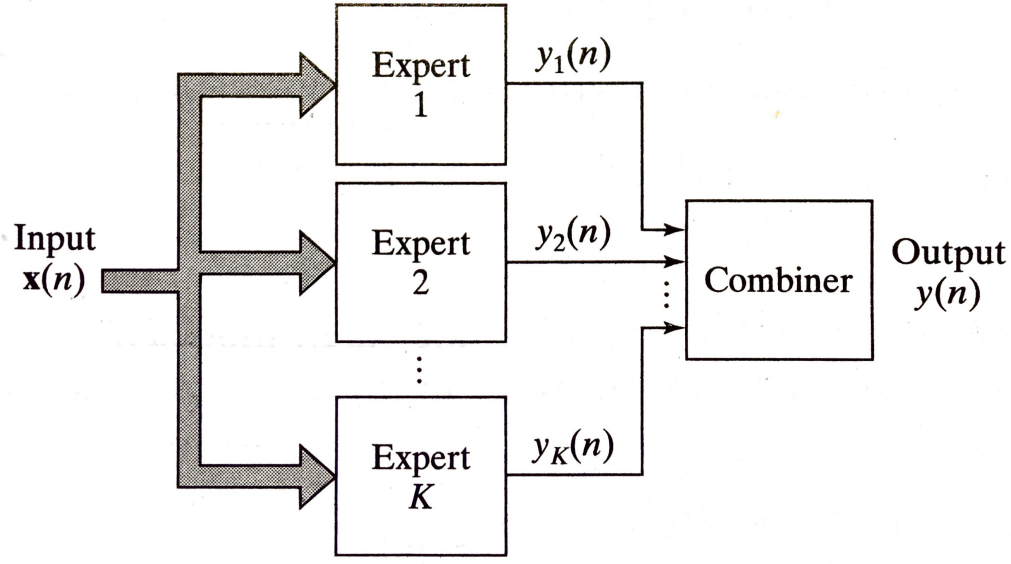
Ensemble Averaging

Boosting

Mixture of Experts

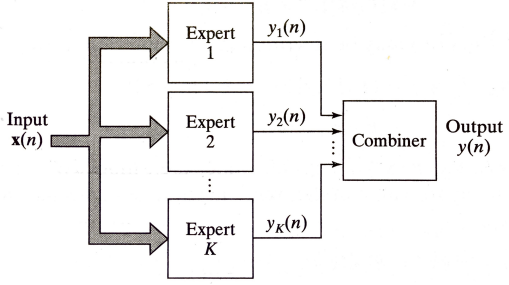
Hiearchial Mixture of Experts

Ensemble Averaging



Ensemble Averaging

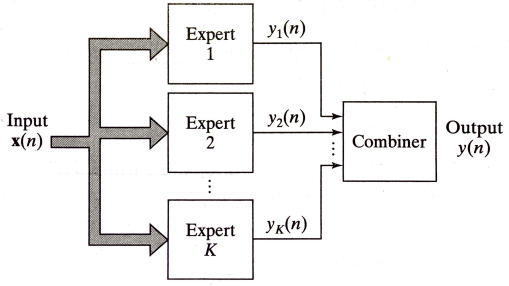
- Assume separately trained experts, e.g, MLPs
- Outputs are combined to get an overall output
- This is a referred to **ensemble averaging** method
- If the input data is bootstrapped for the experts, the approach known as bootstrap aggregation or **bagging**



Ensemble Averaging

Motivation:

- A single such network would have a **vast number of parameters**, thus, longer training time
- Risk of **overfitting** increases when the number of parameters is large
- Differently trained experts suffer from **different local minima** that suggest that a combination should work better



Rationale of Ensemble Averaging

Assume that the true function we want to predict is $h(\mathbf{x})$ so that

$$y_m(\mathbf{x}) = h(\mathbf{x}) + \epsilon_m(\mathbf{x}) \tag{1}$$

where $\epsilon_m(\mathbf{x})$ is the error.

The average sum of squares error by the models acting individually is

$$E_{\mathrm{AV}} = \frac{1}{M}\sum_{m=1}^{M} \mathrm{E}_{\mathbf{x}}\left[\epsilon_m(\mathbf{x})^2\right] \tag{2}$$

The expected committee error is

$$E_{\mathrm{COM}} = \mathrm{E}_{\mathbf{x}}\left[\left(\frac{1}{M}\sum_{m=1}^{M} y_m(\mathbf{x}) - h(\mathbf{x})\right)^2\right] = \mathrm{E}_{\mathbf{x}}\left[\left(\frac{1}{M}\sum_{m=1}^{M} \epsilon_m(\mathbf{x})\right)^2\right] \tag{3}$$

Assuming zero mean, uncorrelated errors implies

$$E_{\mathrm{COM}} = \frac{1}{M} E_{\mathrm{AV}}. \tag{4}$$

Rationale of Ensemble Averaging

Assume that the true function we want to predict is $h(\mathbf{x})$ so that

$$y_m(\mathbf{x}) = h(\mathbf{x}) + \epsilon_m(\mathbf{x}) \tag{1}$$

where $\epsilon_m(\mathbf{x})$ is the error.

The average sum of squares error by the models acting individually is

$$E_{\mathrm{AV}} = \frac{1}{M}\sum_{m=1}^{M} \mathrm{E}_{\mathbf{x}}\left[\epsilon_m(\mathbf{x})^2\right] \tag{2}$$

Have you ever thought of why democracy works?

- The result can be seen as reasoning why democracy works better than dictatorship!
- Another kind of voting schemes may be preferred if outliers are assumed

Assuming zero mean, uncorrelated errors implies

$$E_{\mathrm{COM}} = \frac{1}{M} E_{\mathrm{AV}}. \tag{4}$$

Training Strategy

Further theoretical study suggests that the overall error subject to *varying initial conditions* is reduced if

- The constituted experts are purposedly **overtrained** (low bias, high variance)
- The high variance is reduced by ensemble averaging (bias unchanged).

Outline of lecture

Bayesian Model Averaging

Ensemble Averaging

Boosting

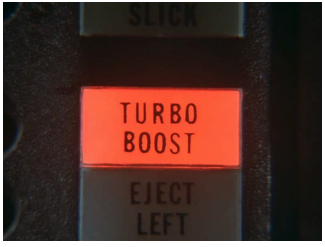
Mixture of Experts

Hiearchial Mixture of Experts

Boosting

A weak learning algorithm is transformed to a strong one obtaining arbitrarily high level of learning accuracy

- In contrast to ensemble averaging, in **boosting** the experts are trained with entirely different distributions
- General method that can be use to improve the performance of **any** learning algorithm.



Boosting Approaches

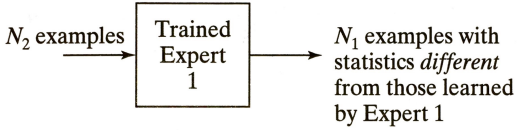
1. Boosting by **filtering**
   - Filtering of training examples by different versions of weak learning algorithm
   - Needs large source of examples
2. Boosting by **subsampling**
   - Training of fixed size
   - Examples are resampled
3. Boosting by **reweighting**
   - Weak learning algorithm needs to be able to handle examples with weights

Boosting by Filtering

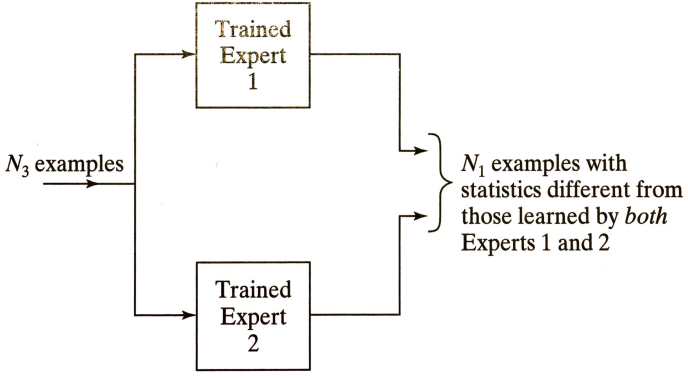
Algorithm (Training)

1. Train Expert 1 with N_1 examples (Training Set 1).
2. Filter the data using Expert 1 to **filter** independent examples
   - Toss a fair coin
   - If **heads**: pass the examples through Expert 1 until an example is misclassified. Add the example to the Training Set 2.
   - If **tails**: pass the examples through Expert 1 until an example is correctly classified. Add the example to the Training Set 2
   - Iterate until Training Set 2 has N_1 examples.
3. Train the Expert 2 using Training Set 2.
4. Filter independent examples using the Experts 1 and 2.
   - Draw a new example and pass through Experts 1 and 2.
   - If the experts agree, discard the example. Otherwise add to the Training Set 3.
   - Iterate until Training Set 3 has N_1 examples.
5. Train the Expert 3 using Training Set 3.

Boosting by Filtering



(a) Filtering of examples performed by Expert 1



(b) Filtering of examples performed by Experts 2 and 3

Boosting by Filtering

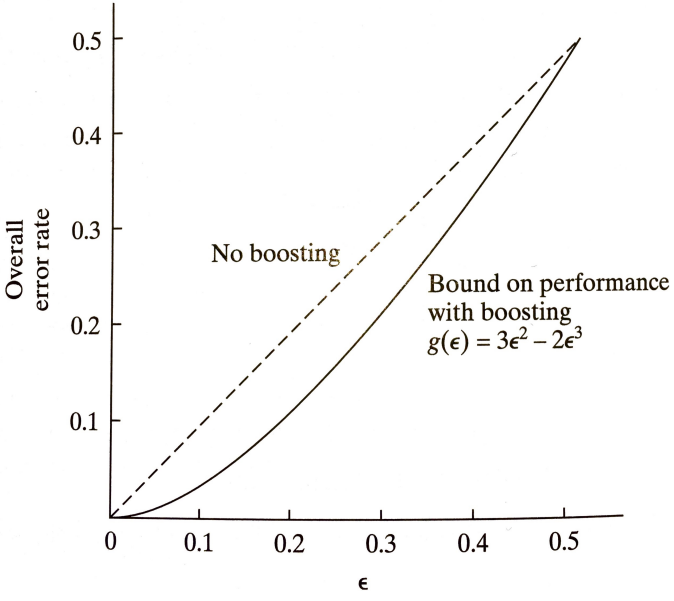
Algorithm (Testing)

Input: a test example

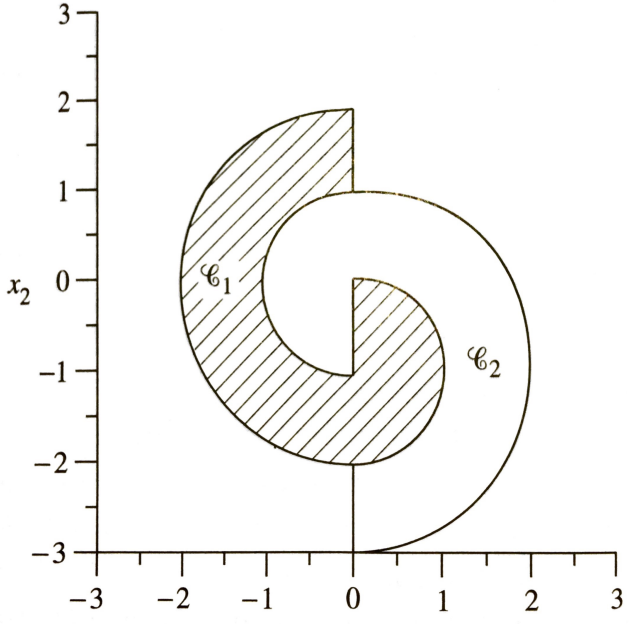
1. Compute the response by the three Experts
2. Combine the results (voting/summing)

Original work used voting, however, combination by summation has been shown to yield even better results.

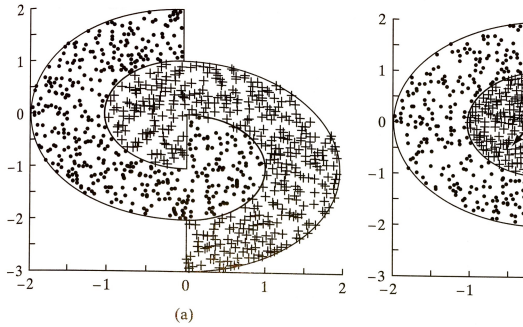
Error Rate


Overall error rate
0.5
0.4
0.3
0.2
0.1
No boosting
Bound on performance with boosting
$g(\epsilon) = 3\epsilon^2 - 2\epsilon^3$
0
0.1
0.2
0.3
0.4
0.5
ϵ

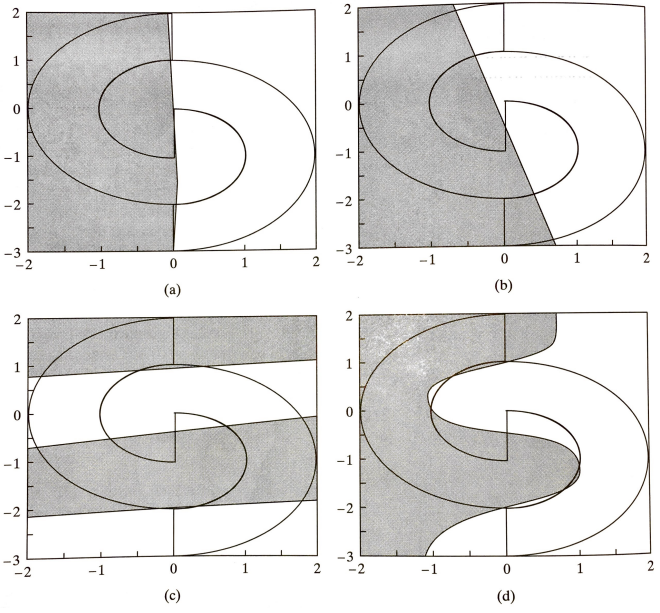
Example: Boosting by Filtering



Example: Boosting by Filtering



Example: Boosting by Filtering



Boosting by Subsampling: AdaBoost

Boosting by filtering requires a large number of training samples.

AdaBoost permits the data to be reused.

- Adjusts to the errors of the weak hypothesis **adaptively**
- The bound on performance only depends on the performance of the learnt weak models

AdaBoost concentrates on those training examples that are hardest to classify.

AdaBoost Algorithm

TABLE 7.2 Summary of AdaBoost

Input: Training sample $\{(\mathbf{x}_i, d_i)\}_{i=1}^{N}$
Distribution $\mathcal{D}$ over the N labeled examples
Weak learning model
Integer T specifying the number of iterations of the algorithm

Initialization: Set $\mathcal{D}_1(i)=1/N$ for all i

Computation: Do the following for $n = 1, 2, \ldots, T$:

1. Call the weak learning model, providing it with the distribution $\mathcal{D}_n$.
2. Get back hypothesis $\mathcal{F}_n : \mathbf{X} \rightarrow Y$
3. Calculate the error of hypothesis $\mathcal{F}_n$:

$$\epsilon_n = \sum_{i:\mathcal{F}_n(\mathbf{x}_i) \neq d_i} \mathcal{D}_n(i)$$

4. Set $\beta_n = \dfrac{\epsilon_n}{1 - \epsilon_n}$
5. Update the distribution $\mathcal{D}_n$:

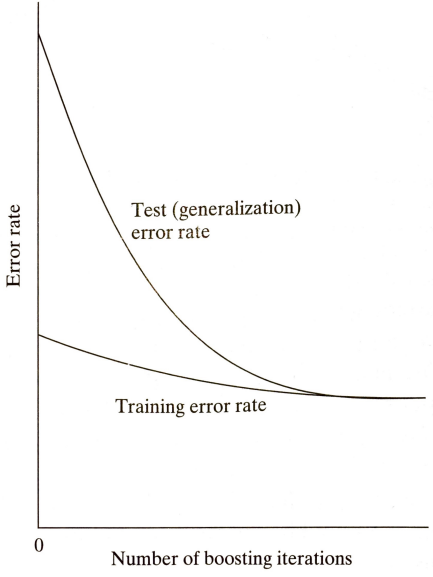
$$\mathcal{D}_{n+1}(i) = \frac{\mathcal{D}_n(i)}{Z_n} \times \begin{cases} \beta_n & \text{if } \mathcal{F}_n(\mathbf{x}_i) = d_i \\ 1 & \text{otherwise} \end{cases}$$

where Z_n is a normalization constant (chosen so that $\mathcal{D}_{n+1}(i)$ is a probability distribution).

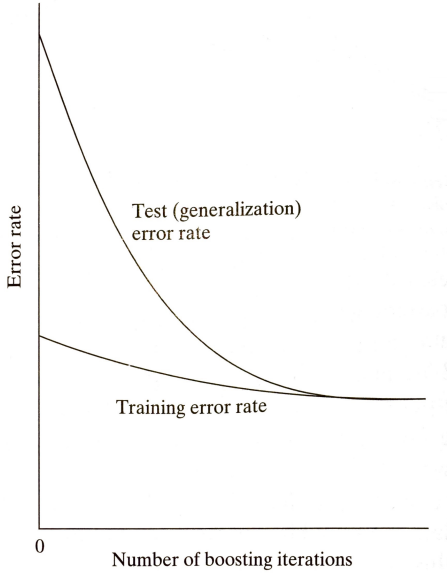
Output: The final hypothesis is

$$\mathcal{F}_n(\mathbf{x}) = \arg\max_{d \in \mathcal{D}} \sum_{n:\mathcal{F}_n(\mathbf{x})=d} \log\frac{1}{\beta_n}$$

Error Performance of AdaBoost



Error performance of AdaBoost



- Surprising behaviour what we have learnt from generalisation error of a single neural network
- Seems to contradict **Occam's razor**: more complex model, better generalisation
- Can be explained by relating the error to the distribution of margins vs voting classification error (Shapire et al. 1997)

Outline of lecture

Bayesian Model Averaging

Ensemble Averaging

Boosting

Mixture of Experts

Hiearchial Mixture of Experts

Associative Gaussian Mixture Model

Consider a **regression** problem with the regressor $\mathbf{x}$ and the response d and the corresponding random variable D.

We assume the following generative model for d

Probabilistic Generative Model

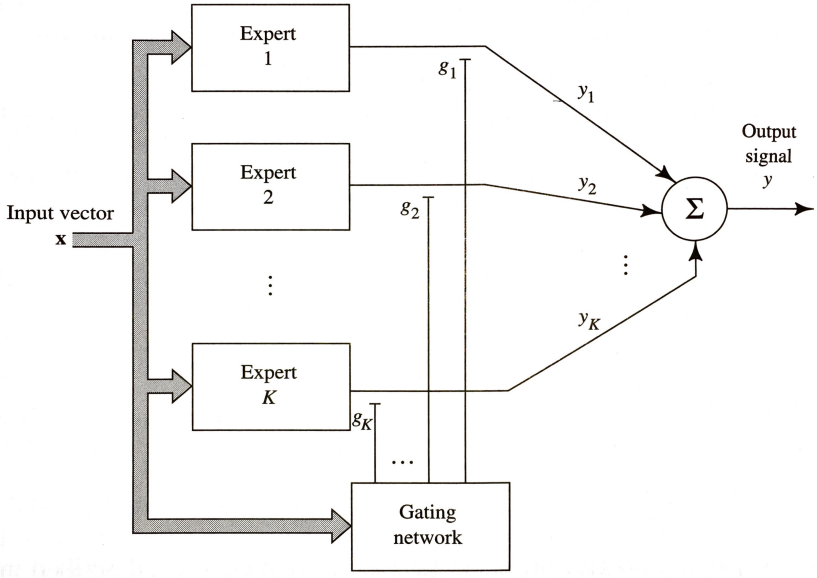
1. An input vector $\mathbf{x}$ is randomly picked from a prior distribution
2. The k^{th} rule is selected from the conditional $P(k|\mathbf{x}, \mathbf{a}^{(0)})$, given $\mathbf{x}$ and the parameter vector $\mathbf{a}^{(0)}$.
3. For the rule k, $k = 1, 2, \ldots, K$, the model response d is linear in $\mathbf{x}$ with and additive error ϵ_k, $\epsilon_k \sim N(0, \sigma_k^2)$.

The conditional probability for the response D is

$$P(D = d|\mathbf{x}, \theta^{(\mathbf{0})}) = \sum_{k=1}^{K} P(D = d|\mathbf{x}, \mathbf{w}_k^{(0)}) P(k|\mathbf{x}, \mathbf{a}^{(0)}) \tag{5}$$

where $\theta^{(\mathbf{0})} = (\mathbf{a}^{(0)}, \mathbf{w}_1^{(0)}, \ldots, \mathbf{w}_K^{(0)}, \sigma_1^{(0)}, \ldots, \sigma_K^{(0)})$ is the generative model parameter vector.

Mixture of Experts (ME) Model



Expert k

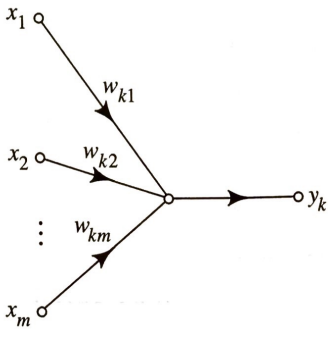


FIGURE 7.9 Signal-flow graph of a single linear neuron constituting expert k.

Each expert is a linear filter

$$y_k = \mathbf{w}_\mathbf{k}^\mathrm{T}\mathbf{x}, \quad k = 1, 2, \ldots, K. \tag{6}$$

Gating Network

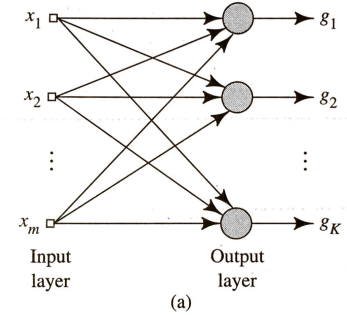


FIGURE 7.10 (a) Single layer of softmax neurons for the gating network. (b) Signal-flow graph of a softmax neuron.

Neurons on the gating networks are nonlinear with the soft max activation g_k.

The activation

$$u_k = \mathbf{a}_k^{\mathrm{T}}\mathbf{x}, \quad k = 1, 2, \ldots, K. \tag{7}$$

The gating network is a "classifier" that maps the input vector $\mathbf{x}$ into multinomial probabilities.

The Probabilistic Model

The probability density function of D, given the input vector $\mathbf{x}$ is the **mixture**

$$p_D(d|\mathbf{x},\theta) = \frac{1}{\sqrt{2\pi}} \sum_{k=1}^{K} \sigma_k^{-1/2} g_k \exp\left(-\frac{1}{2\sigma_k^2}(d-y_k)^2\right), \quad k = 1, 2, \ldots, K, \tag{8}$$

where the mixing coefficients are obtained from the gating network.

Important Insight of the ME model

1. The output y_k of the Expert k is the conditional mean of the random variable representing the response D, given $\mathbf{x}$.
2. The output of the gating network defines the multinomial probability that each expert matches with the value $D = d$ on the basis of the knowledge gained from $\mathbf{x}$.

Outline of lecture

Bayesian Model Averaging

Ensemble Averaging

Boosting

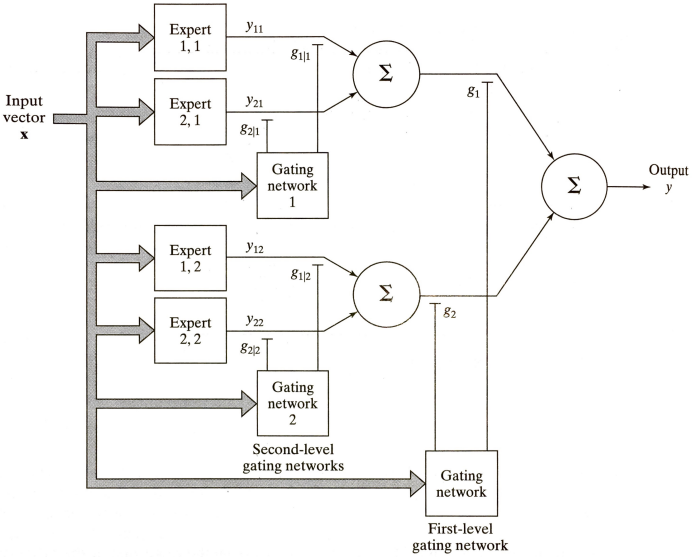
Mixture of Experts

Hiearchial Mixture of Experts

Hierarchical Mixture of Experts (HME) Model

Hierarchical Mixture of Experts (HME)

- Natural generalisation of the ME model
- **Tree** like architecture
- Input space is divided to **nested** subspaces
- Gating networks arranged in a **Hierarchical** manner



Hierarchical Mixture of Experts (HME)

Fundamentals of HME

1. **The HME is a product of the divide and conquer strategy**
   - By recursion, input space is divided into subregions
   - The subregions are modelled **locally**
2. **The HME model is a soft-decision tree**
   - ME is one-level decision tree or **decision stump**
   - Standard decision tree produces **hard** decisions (yes/no) in different regions of the input space
   - HME is viewed as the probabilistic formulation that provides **soft** decisions
     - Loss of information vs. information preservation
     - Greediness vs. decision alternation

Learning of the HME

Due to the probabilistic formulation, the natural approach to estimate the HME is **maximum likelihood estimation**

Learning Strategies

- **Stochastic gradient decent**
- **Expectation Maximisation EM**

Summary

1. **Ensemble averaging** improved the performance by using
   - Reduction of bias by purposely overfitting individual experts
   - Reduction of variance using different initial conditions and averaging the expert responses
2. **Boosting** improves the performance by
   - Individual experts re required to perform only slightly better than random guessing
   - The the weak learner are converted to a strong one by either **filtering** or **resampling** (AdaBoost)
   - The experts eventually learn the entire distribution
3. **Mixture of Experts Model** is an associative Gaussian mixture model with a probabilistic formulation
4. **Hierarchical Mixture of Experts Model** is
   - A collection of local learners by divide and conquer strategy.
   - A soft generalisation of the decision tree

Next lecture

- Reinforcement learning

References I


Alpaydin, E. (2014). *Introduction to Machine Learning*. Third Edition. The MIT Press.


Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.


Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*. Second Edition. Prentice Hall.