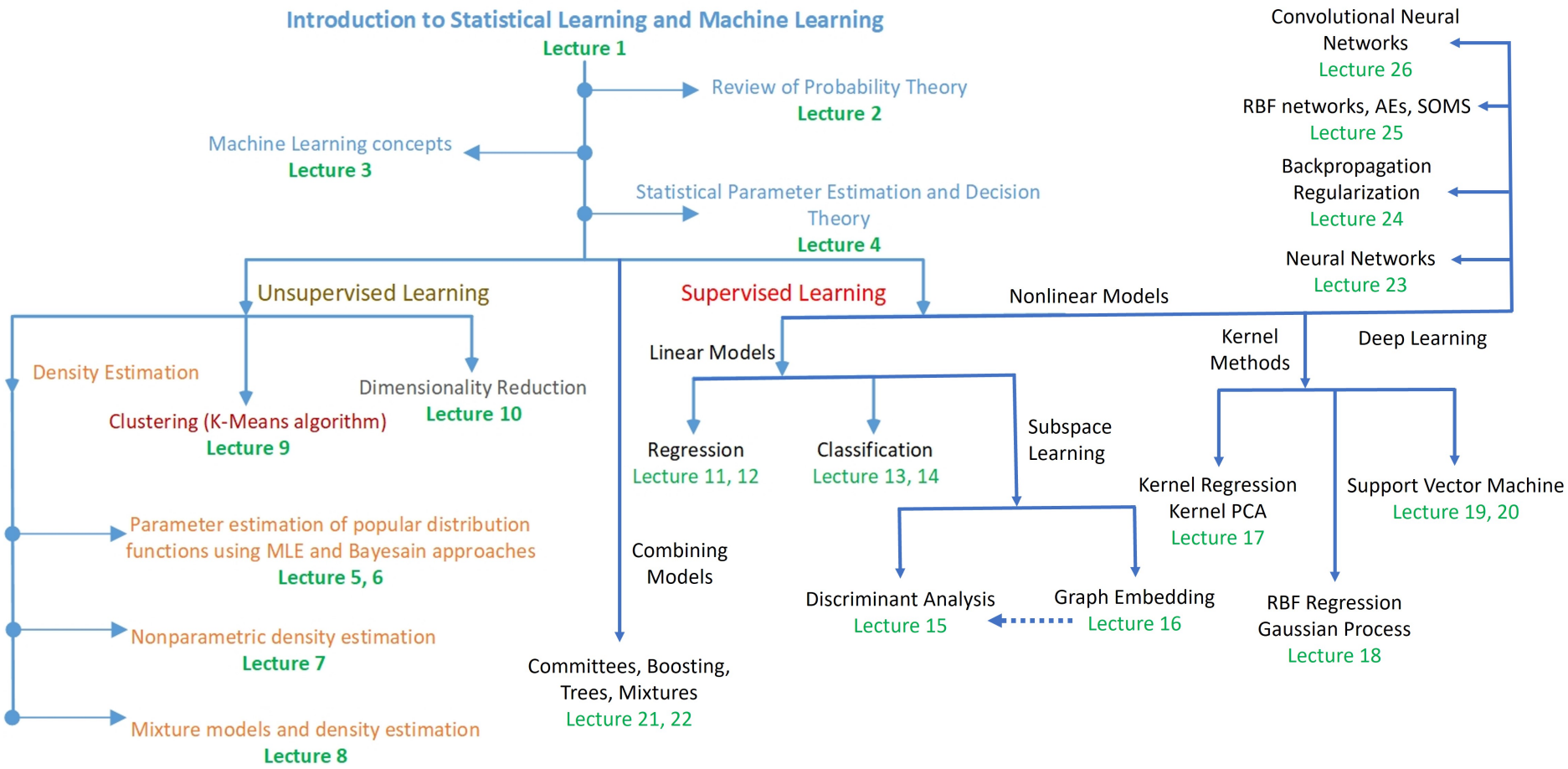
Statistical Learning and Machine Learning

Lecture 22 - Combining Models 2

October 13, 2021

Course overview and where do we stand



Why to combine multiple models?

Sometimes combining multiple models can lead to better performance compared to using only one:

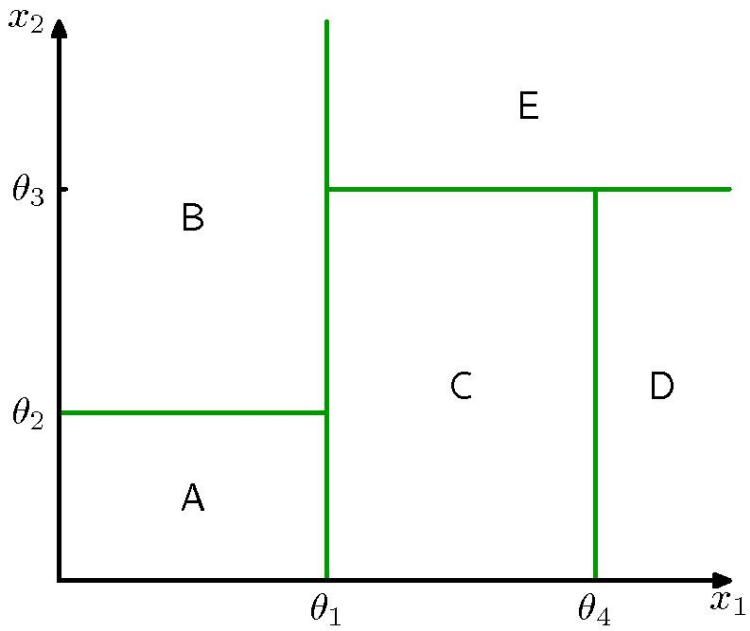
- *Committees*: Use L different models and then make predictions using the average of the predictions of these L models
- *Boosting*: Use multiple models *in a sequence* in which each model's training is depending on the models preceding it in the sequence
- *Decision trees*: Divide the space in multiple regions and train one model for each region.
- *Mixture of experts*: Train K models and combine them based on a probabilistic mixture of the form:

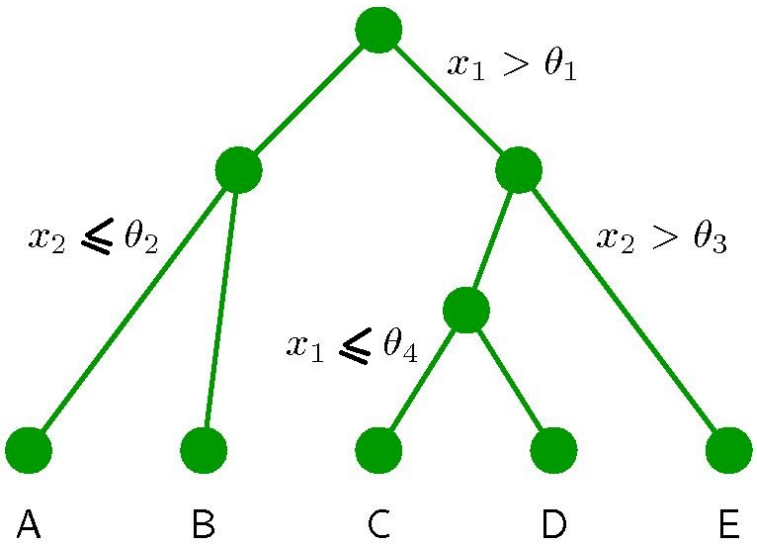
$$p(t|\mathrm{x}) = \sum_{k=1}^{K} p(k|\mathrm{x})p(t|\mathrm{x}, k) \tag{1}$$

Tree-based models

Process:

1. Partition the input space into cuboid regions
2. Assign a simple model (e.g. a constant) to each region





Tree-based models

Given $\{x_1, \ldots, x_N\}$ and the corresponding targets $\{t_1, \ldots, t_N\}$:

1. Start with a single root node, corresponding to the whole input space
2. Grow the tree by adding nodes one at a time:
   - Add a pair of leaf nodes to the existing tree corresponding to a candidate variable that is split in two
   - Determine the corresponding threshold for this variable
   - For a given choice of split variable and threshold, the optimal choice of predictive variable is given by the local average of the data
3. Repeated for all possible regions to be split, and the one that gives the smallest residual sum-of-squares error is retained.

Grow a large tree, using a stopping criterion based on the number of data points associated with the leaf nodes, and then prune back the resulting tree.

Tree-based models

Denote by T_0 the starting tree formed by $\tau = 1, \ldots, |T|$ leaf nodes, with leaf node τ representing a region $\mathcal{R}_\tau$ including N_τ data points.
The optimal prediction for $\mathcal{R}_\tau$ is:

$$y_\tau = \frac{1}{N_\tau} \sum_{x_n \in \mathcal{R}_\tau} t_n \tag{2}$$

For a regression problem, the contribution of $\mathcal{R}_\tau$ to the residual sum-of-squares is:

$$Q_\tau(T) = \sum_{x_n \in \mathcal{R}_\tau} (t_n - y_n)^2 \tag{3}$$

and the pruning criterion is:

$$C(T) = \sum_{\tau=1}^{|T|} Q_\tau(T) + \lambda |T| \tag{4}$$

For classification problems $Q_\tau(T)$ is replaced by a classification loss (e.g. the cross-entropy loss).

Mixture models: Linear regression

We consider K linear models mapping ϕ vector to variable t:

- each having its own weight parameter vector $\mathrm{w}_k,\ k = 1, \ldots, K$
- all having the same noise variance, expressed by the precision parameter β
- The mixture distribution is:

$$p(t|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(t|\mathrm{w}_k^T \boldsymbol{\phi}, \beta^{-1}) \tag{5}$$

where $\pi_k,\ k = 1, \ldots, K$ are the mixing coefficients and $\boldsymbol{\theta}$ expresses the parameters of the mixture model ($\mathrm{w}_k,\ \pi_k,\ k = 1, \ldots, K$ and β)

Mixture models: Linear regression

Given a set of data points and target values $\{\phi_n, t_n\},\ n = 1, \ldots, N$ the log-likelihood functions is:

$$\ln p(\mathrm{t}|\boldsymbol{\theta}) = \sum_{n=1}^{N} \ln \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(t_n | \mathrm{w}_k^T \boldsymbol{\phi}_n, \beta^{-1}) \right) \tag{6}$$

To optimize the log-likelihood function we introduce a set of 1-of-K variables $\mathrm{Z} = \{\mathrm{z}_n\}$ (with $z_{nk} \in \{0, 1\}$) indicating which component of the mixture is responsible for generating the data point:

$$\ln p(\mathrm{t}, \mathrm{Z}, |\boldsymbol{\theta}) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \ln \left(\pi_k \mathcal{N}(t_n | \mathrm{w}_k^T \boldsymbol{\phi}_n, \beta^{-1}) \right) \tag{7}$$

We use the EM algorithm to estimate $\{\mathrm{w}_k, \pi_k\},\ k = 1, \ldots, K$ and β

Mixture models: Linear regression

Expectation Maximization algorithm:
Start by using $\boldsymbol{\theta}^{old}$

E-step: Use $\boldsymbol{\theta}^{old}$ to evaluate the posterior probabilities (or *resposibilities*)

$$\gamma_{nk} = \mathbb{E}[z_{nk}] = p(k|\boldsymbol{\phi}_n, \boldsymbol{\theta}^{old}) = \frac{\pi_k \mathcal{N}(t_n|\mathrm{w}_k^T \boldsymbol{\phi}_n, \beta^{-1})}{\sum_j \pi_j \mathcal{N}(t_n|\mathrm{w}_j^T \boldsymbol{\phi}_n, \beta^{-1})} \tag{8}$$

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) = \mathbb{E}[\ln p(\mathrm{t}, \mathrm{Z}|\boldsymbol{\theta})] = \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} \Big(\ln \pi_k + \ln \mathcal{N}(t_n|\mathrm{w}_k^T \boldsymbol{\phi}_n, \beta^{-1}) \Big) \tag{9}$$

Mixture models: Linear regression

Expectation Maximization algorithm:

M-step: Maximize $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{old})$ w.r.t. $\{\mathrm{w}_k, \pi_k\}$, $k = 1, \ldots, K$ and β keeping γ_{nk} fixed, and considering that $\sum_k \pi_k = 1$:

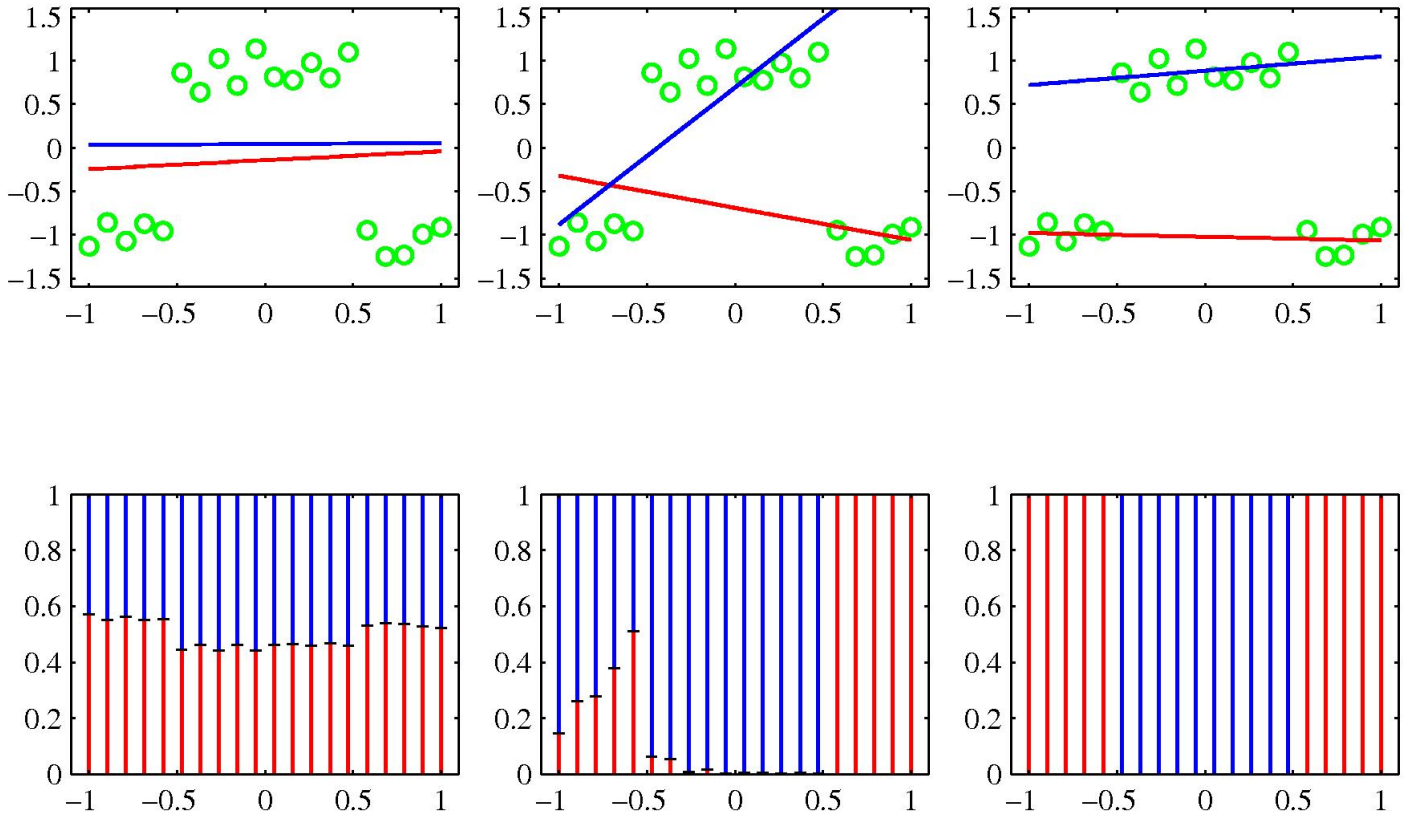
$$\pi_k = \frac{1}{N} \sum_{n=1}^{N} \gamma_{nk} \tag{10}$$

$$\mathrm{w}_k = \left(\Phi^T \mathrm{R}_k \Phi\right)^{-1} \Phi^T \mathrm{R}_k \mathrm{t} \tag{11}$$

$$\frac{1}{\beta} = \frac{1}{N} \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} (t_n - \mathrm{w}_k^T \phi_n)^2 \tag{12}$$

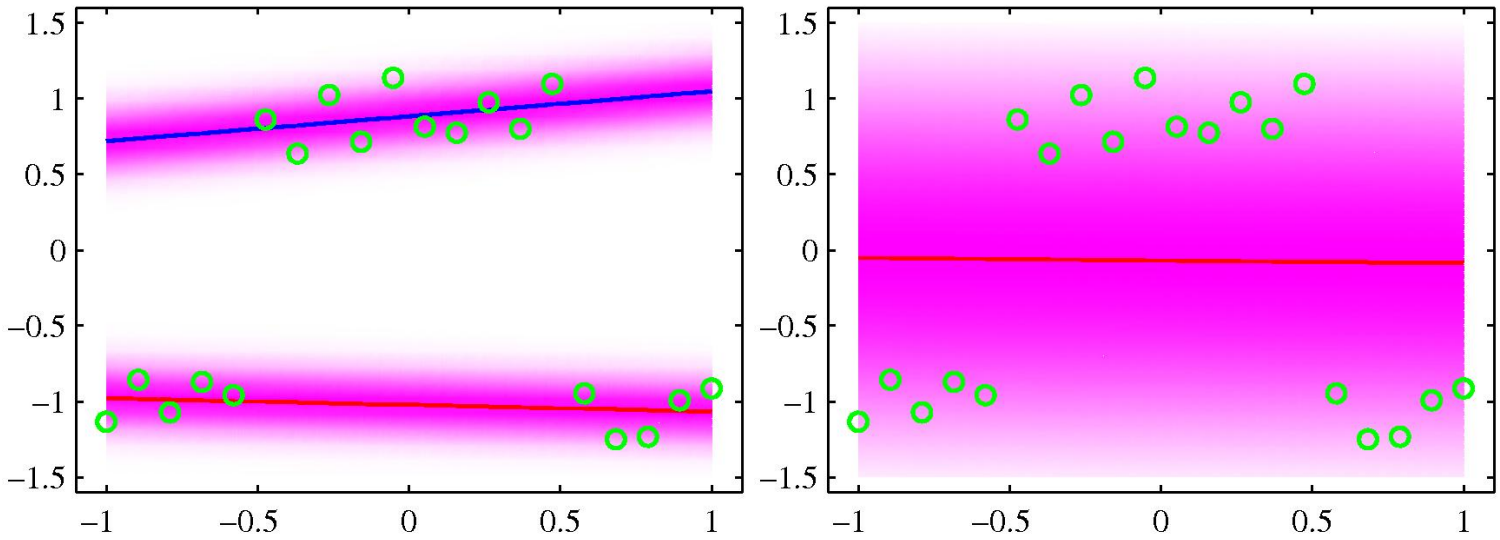
where $\mathrm{R}_k = diag(\gamma_{nk})$: each data point ϕ_n is contributing to the calculation of the parameters of the k-th model according to the corresponding responsibility value γ_{nk}.

Mixture models: Linear regression



Regression using mixture of two linear regression models
Top: After 30 (left) and 50 iteration (center), final result (right)
Bottom: The responsibility values $p(k|\phi_n, \theta)$, $k = 1, 2$

Mixture models: Linear regression



Left: Predictive distribution for a mixture of two linear regression models
Right: Predictive density for a single linear regression model

Mixture models: Logistic regression

For a mixture of K logistic regression models:

$$p(t|\phi, \theta) = \sum_{k=1}^{K} \pi_k y_k^t \left[1 - y_k\right]^{1-t} \tag{13}$$

where:

- ϕ is the feature vector
- $y_k = \sigma(\mathrm{w}_k^T \phi)$
- θ denotes the parameters $\{\pi_k, \mathrm{w}_k\}$, $k = 1, \ldots, K$

Mixture models: Logistic regression

Given a set of data points and target values $\{\phi_n, t_n\},\ n = 1, \ldots, N$ the likelihood functions is:

$$p(\mathrm{t}, \boldsymbol{\theta}) = \prod_{n=1}^{N} \left(\sum_{k=1}^{K} \pi_k y_{nk}^{t_n} [1 - y_{nk}]^{1-t_n} \right) \tag{14}$$

with $y_{nk} = \sigma(\mathrm{w}_k^T \phi_n)$ and $\mathrm{t} = [t_1, \ldots, t_N]^T$.

To optimize the likelihood function we introduce a set 1-of-K variables $\mathrm{Z} = \{\mathrm{z}_n\}$ indicating which component of the mixture is responsible for generating the data point:

$$p(\mathrm{t}, \mathrm{Z}, |\boldsymbol{\theta}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \left(\pi_k y_{nk}^{t_n} [1 - y_{nk}]^{1-t_n} \right)^{z_{nk}} \tag{15}$$

We use the EM algorithm to estimate $\{\mathrm{w}_k, \pi_k\},\ k = 1, \ldots, K$ and β

Mixture models: Logistic regression

Expectation Maximization algorithm:
Start by using $\boldsymbol{\theta}^{old}$

E-step: Use $\boldsymbol{\theta}^{old}$ to evaluate the posterior probabilities (or *resposibilities*)

$$\gamma_{nk} = \mathbb{E}[z_{nk}] = p(k|\phi_n, \boldsymbol{\theta}^{old}) = \frac{\pi_k y_{nk}^{t_n}[1-y_{nk}]^{1-t_n}}{\sum_j \pi_j y_{nj}^{t_n}[1-y_{nj}]^{1-t_n}} \tag{16}$$

$$\begin{aligned} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) &= \mathbb{E}_z[\ln p(\mathrm{t}, \mathrm{Z}|\boldsymbol{\theta})] \\ &= \sum_{n=1}^{N}\sum_{k=1}^{K}\gamma_{nk}\Big(\ln \pi_k + t_n \ln y_{nk} + (1-t_n)\ln(1-y_{nk})\Big) \end{aligned} \tag{17}$$

Mixture models: Logistic regression

Expectation Maximization algorithm:

M-step: Maximize $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{old})$ w.r.t. $\{\mathrm{w}_k, \pi_k\}$, $k = 1, \ldots, K$ keeping γ_{nk} fixed

$$\pi_k = \frac{1}{N} \sum_{n=1}^{N} \gamma_{nk} \tag{18}$$

$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{old})$ does not have a closed form solution for w_k and they need to be solved iteratively using the iterative reweighted least squares (IRLS) algorithm.

$$\nabla_k Q = \sum_{n=1}^{N} \gamma_{nk}(t_n - y_{nk})\boldsymbol{\phi}_n \tag{19}$$

$$\mathrm{H}_k = -\nabla_k \nabla_k Q = \sum_{n=1}^{N} \gamma_{nk} y_{nk}(1 - y_{nk})\boldsymbol{\phi}_n \boldsymbol{\phi}_n^T \tag{20}$$

Mixture models: Logistic regression

IRLS algorithm:

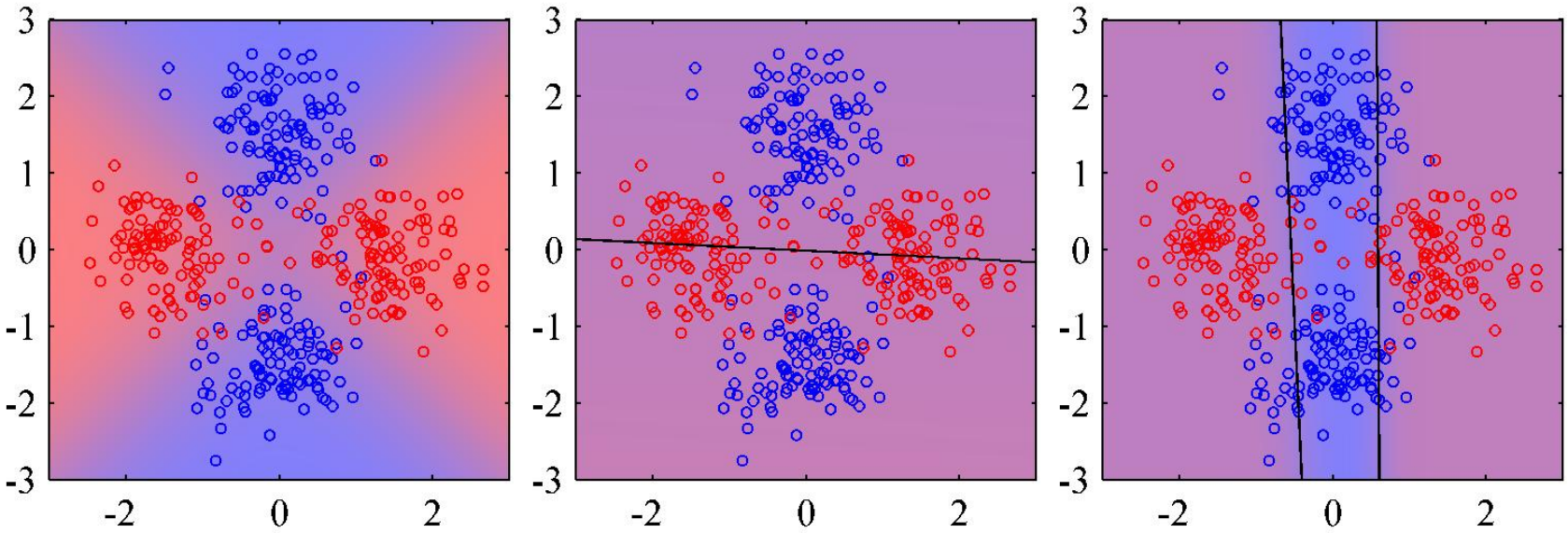
$$\nabla_k Q = \sum_{n=1}^{N} \gamma_{nk}(t_n - y_{nk})\phi_n \tag{21}$$

$$\mathrm{H}_k = -\nabla_k \nabla_k Q = \sum_{n=1}^{N} \gamma_{nk} y_{nk}(1 - y_{nk})\phi_n \phi_n^T \tag{22}$$

where ∇_k denotes the derivative w.r.t. w_k

For fixed γ_{nk} the above equations are independent of $\{\mathrm{w}_j\}$, $j \neq k$. Thus, we can solve for each w_k independently using the IRLS algorithm.

Mixture models: Logistic regression



Left: true probability of each class
Center: Result of fitting a single logistic regression model
Right: Result of fitting a mixture of two logistic regression models

Mixture of experts

We can increase the capability of mixture models by allowing the mixing coefficients be functions of the input variable too:

$$p(\mathrm{t}|\mathrm{x}) = \sum_{k=1}^{K} \pi_k(\mathrm{x}) p_k(\mathrm{t}|\mathrm{x}). \tag{23}$$

The mixing coefficients $\pi_k(\mathrm{x})$ are known as *gating functions* and the component densities $p_k(\mathrm{t}|\mathrm{x})$ are known as *experts*.

The gating functions need to satisfy the constraints:

$$0 \leq \pi_k(\mathrm{x}) \leq 1 \quad \text{and} \quad \sum_{k=1}^{K} \pi_k(\mathrm{x}) = 1. \tag{24}$$