Bayesian Learning

Complicated derivation of known things.

- **Maximum a posteriori probability hypothesis** (MAP) (nejpravděpodobnější hypotéza)
- **Maximum likelihood hypothesis** (ML) (maximálně věrohodná hypotéza)
- **Bayesian optimal prediction** (Bayes Rate)
- **Bayesian methods**, **bayesian smoothing**
  - 'complexity penalty' is our prior distribution/preference on parameters.
- **Naive Bayes model (classifier)**.
- **EM algorithm**
  - The best way to fill in missing values
  - the most common application is clustering
  - but the use is far broader, for example
  - Baum-Welch algorithm for HMM
  - variational approximation for continuous distributions.

Candy Example (Russel, Norvig: Artif. Intell. a MA)

- Our favorite candy comes in two flavors: cherry and lime, both in the same wrapper.
- They are in a bag in one of following rations of cherry candies and prior probability of bags:

hypothesis (bag type)	h_1	h_2	h_3	h_4	h_5
cherry	100%	75%	50%	25%	0%
prior probability h_i	10%	20%	40%	20%	10%

- The first candy is cherry.

  MAP Which of h_i is the most probable given first candy is cherry?

  Bayes estimate What is the probability next candy from the same bag is cherry?

Maximum Aposteriory Probability Hypothesis (MAP)

- We assume large bags of candies, the result of one missing candy in the bag is negligable.
- Recall Bayes formula:

$$P(h_i|B=c) = \frac{P(B=c|h_i)\cdot P(h_i)}{\sum_{j=1,\ldots,5} P(B=c|h_j)\cdot P(h_j)} = \frac{P(B=c|h_i)\cdot P(h_i)}{P(B=c)}$$

- We look for the MAP hypothesis **maximálně aposteriorně pravděpodobná**

$$argmax_i P(h_i|B=c) = argmax_i P(B=c|h_i)\cdot P(h_i).$$

- Aposteriory probabilities of hypotheses are in the following table.

Candy Example: Aposteriory Probability of Hypotheses

index	prior	cherry ratio	cherry AND h_i	aposteriory prob. h_i
i	$P(h_i)$	$P(B = c\|h_i)$	$P(B = c\|h_i) \cdot P(h_i)$	$P(h_i\|B = c)$
1	0.1	1	0.1	0.2
2	0.2	0.75	0.15	0.3
3	0.4	0.5	0.2	**0.4**
4	0.2	0.25	0.05	0.1
5	0.1	0	0	0

- Which hypothesis is most probable?

$$h_{MAP} = argmax_i P(data|h_i) \cdot P(h_i)$$

- What is the prediction of a new candy according the most probable hypothesis h_{MAP}?

Bayesian Learning, Bayesian Optimal Prediction

- **Bayesian optimal prediction** is weigthed average of predictions of all hypotheses:

$$\begin{aligned} P(N = c|data) &= \sum_{j=1,\ldots,5} P(N = c|h_j, data) \cdot P(h_j|data) \\ &= \sum_{j=1,\ldots,5} P(N = c|h_j) \cdot P(h_j|data) \end{aligned}$$

- If our model is correct, no prediction has smaller expected error then Bayesian optimal prediction.
- We always assume i.i.d. data, independently identically distributed.
- We assume the hypothesis fully describes the data behavior. Observations are mutually conditionally independent given the hypothesis. This allows the last equation above.
- The error of the bayesian optimal prediction is called the **bayes error rate**.

  It is analogous to **ireducible error** from 'bias variance decomposition'.

Candy Example: Bayesian Optimal Prediction

i	$P(h_i \mid B = c)$	$P(N = c \mid h_i)$	$P(N = c \mid h_i) \cdot P(h_i \mid B = c)$
1	0.2	1	0.2
2	0.3	0.75	0.225
3	0.4	0.5	0.2
4	0.1	0.25	0.02
5	0	0	0
$\sum$	1		0.645

Maximum Likelihood Estimate (ML)

- Usually, we do not know prior probabilities of hypotheses.
- Setting all prior probabilities equal leads to **Maximum Likelihood Estimate, maximálně věrohodný odhad**

$$h_{ML} = argmax_i P(data|h_i)$$

- Probability of data given hypothesis = likelihood of hypothesis given data.
- Find the ML estimate:

index	prior	cherry ratio	cherry AND h_i	Aposteriory prob. h_i
i	$P(h_i)$	$P(B = c\|h_i)$	$P(B = c\|h_i) \cdot P(h_i)$	$P(h_i\|B = c)$
1	0.1	1	0.1	0.2
2	0.2	0.75	0.15	0.3
3	0.4	0.5	0.2	0.4
4	0.2	0.25	0.05	0.1
5	0.1	0	0	0

- In this example, do you prefer ML estimate or MAP estimate?
- (Only few data, over-fitting, penalization is useful. AIC, BIC)

Maximum Likelihood: Continuous Parameter θ

- New producer on the market. We do not know the ratios of candies, any h_θ, kde $\theta \in \langle 0; 1 \rangle$ is possible, any prior probabilities h_θ are possible.
- We look for maximum likelihood estimate.
- For a given hypothesis h_θ, the probability of a cherry candy is θ, of a lime candy $1 - \theta$.
- Probability of a sequence of c cherry and l lime candies is:

$$P(data|h_\theta) = \theta^c \cdot (1-\theta)^l = \theta^{\sum_{i=1}^{N} \mathbb{1}_{Flavor_i = c}} (1-\theta)^{\sum_{i=1}^{N} \mathbb{1}_{Flavor_i = \ell}}.$$

ML Estimate of Parameter θ

- Probability of a sequence of c cherry and l lime candies is:

$$P(data|h_\theta) = \theta^c \cdot (1-\theta)^l$$

- Usual trick is to take logarithm:

$$\ell(h_\theta; data) = c \cdot \log_2 \theta + l \cdot \log_2(1-\theta)$$

- To find the maximum of ℓ (log likelihood of the hypothesis) with respect to θ we set the derivative equal to 0:

$$\begin{aligned} \frac{\partial \ell(h_\theta; data)}{\partial \theta} &= \frac{c}{\theta} - \frac{l}{1-\theta} \\ \frac{c}{\theta} &= \frac{l}{1-\theta} \\ \theta &= \frac{c}{c+l}. \end{aligned}$$

ML Estimate of Multiple Parameters

- Producer introduced two colors of wrappers - red r and green g.
- Both flavors are wrapped in both wrappers, but with different probability of the red/green wrapper.
- We need three parameters to model this situation:

$P(B = c)$	$P(W = r \mid B = c)$	$P(W = r \mid B = l)$
θ_0	θ_1	θ_2

- Following table denotes observed frequences:

wrapper\ flavor	cherry	lime
red	r_c	r_l
green	g_c	g_l

ML Estimate of Multiple Parameters

Parameters are:

$P(B=c)$	$P(W=r\|B=c)$	$P(W=r\|B=l)$
θ_0	θ_1	θ_2

Probability of data given the hypothesis $h_{\theta_0,\theta_1,\theta_2}$ is:

$$\begin{aligned} P(data|h_{\theta_0,\theta_1,\theta_2}) &= \theta_1^{r_c}\cdot(1-\theta_1)^{g_c}\cdot\theta_0^{r_c+g_c}\cdot\theta_2^{r_l}\cdot(1-\theta_2)^{g_l}\cdot(1-\theta_0)^{r_l+g_l} \\ \ell(h_{\theta_0,\theta_1,\theta_2};data) &= r_c\log_2\theta_1 + g_c\log_2(1-\theta_1) + (r_c+g_c)\log_2\theta_0 \\ &\quad + r_l\log_2\theta_2 + g_l\log_2(1-\theta_2) + (r_l+g_l)\log_2(1-\theta_0) \end{aligned}$$

We look for maximum:

$$\begin{aligned} \frac{\partial\ell(h_{\theta_0,\theta_1,\theta_2};data)}{\partial\theta_0} &= \frac{r_c+g_c}{\theta_0} - \frac{r_l+g_l}{1-\theta_0} \\ \theta_0 &= \frac{(r_c+g_c)}{r_c+g_c+r_l+g_l} \\ \frac{\partial\ell(h_{\theta_0,\theta_1,\theta_2};data)}{\partial\theta_2} &= \frac{r_l}{\theta_2} - \frac{g_l}{1-\theta_2} \\ \theta_2 &= \frac{r_l}{r_l+g_l}. \end{aligned}$$

- Maximum Likelihood estimate is the ratio of frequencies.

Naive Bayes Model, Bayes Classifier

- **Naive Bayes Model, Bayes Classifier** assumes independent features given the class variable.
  - Calculate prior probability of classes $P(c_i)$
  - For each feature x_j, calculate for each class the probability of this feature $P(x_j|c_i)$
  - For a new observation of features f predict the most probable class $argmax_{c_i} P(x_j|c_i) \cdot P(c_i)$.
  - Maximum Likelihood estimate is the ratio of frequencies.
    - We may use smoothed estimate adding α samples to each possibility to avoid zero probabilities.
  - ML estimite of a Gaussian distribution parameters are the mean and the variance (or covariance matrix for multivariate distribution).

  Bayes factor
  - We can start with a comparison ratio of two classes $\frac{P(c_i)}{P(c_j)}$
  - after each observation x_p multiply it by the **bayes factor** $\frac{P(x_p|c_i)}{P(x_p|c_j)}$
  - that is:

$$\frac{P(c_i|x_1,\ldots,x_p)}{P(c_k|x_1,\ldots,x_p)} = \frac{P(c_i)}{P(c_k)} \cdot \frac{P(x_1|c_i)}{P(x_1|c_k)} \cdot \ldots \cdot \frac{P(x_p|c_i)}{P(x_p|c_k)}.$$

- Bayesian Networks learn more complex (in)dependencies between features.

Expected Log-Likelihood

- Recall the log-likelihood for a simple parameter θ estimate

$$\ell(h_\theta; data) = c \cdot \log_2 \theta + l \cdot \log_2(1-\theta) = (\sum_{i=1}^{N} \mathbb{1}_{Flavor_i=c}) \cdot \theta + \sum_{i=1}^{N} \mathbb{1}_{Flavor_i=\ell} \cdot (1-\theta)$$

- Let us denote the probabilities of samples $P(Z^m)$ (m for missing)
- Assume we know just the probabilities only probabilities on Z^m data
  - $P(Z_i^m = c) = w_{Z^m=c}$ and
  - $P(Z_i^m = \ell) = w_{Z^m=\ell} = 1 - w_{Z^m=c}$.
- We assume the samples to be independent.
- The expected loglikelihood with respect to $P(Z^m)$ is

$$\mathbb{E}_{P(Z^m)}[\ell_0(\theta; \mathbf{Z}^m)] \quad = \quad \mathbb{E}_{P(Z^m)} \left[(\sum_{i=1}^{N} w_{Z_i^m=c}) \cdot \theta + \sum_{i=1}^{N} w_{Z_i^m=\ell} \cdot (1-\theta) \right]$$

- and the argmax is:

$$\hat{\theta} \quad = \quad \arg\max_\theta \mathbb{E}_{P(Z^m)} \left[(\sum_{i=1}^{N} w_{Flavor_i=c}) \cdot \theta + \sum_{i=1}^{N} w_{Flavor_i=\ell} \cdot (1-\theta) \right]$$

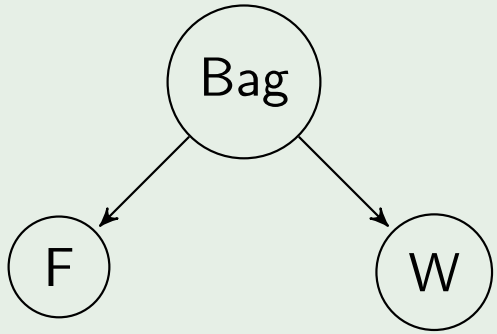
$$= \quad \frac{\sum_{i=1}^{N} w_{Flavor_i=c}}{\sum_{i=1}^{N} w_{Flavor_i=c} + \sum_{i=1}^{N} w_{Flavor_i=\ell}}.$$

Expectation Maximization Algorithm (EM Algorithm)

- EM algorithm estimates the maximum likelihood model based on the data with missing values.
  - used in HMM
  - used in clustering (Gaussian mixture model estimation)
  - but not restricted to this applications
  - It is a general approach to fill missing values based on the maximum likely model.
- Next example estimates parameters with data missing at random.

Example (EM Algorithm for Missing Data)

- Two bags of bonbons mixed together. Each bonbon has a *Wrapper* and flavor *Flavor* and may have *Holes*. Each bag had another ratio of *Wrapper* color and *Flavor*.



Bag	F	W
?	c	r
1	l	r
1	c	?
1	c	g
?	l	?

- Initialize all parameters randomly close to uniform distribution, $\theta_* \approx 0.5$.

$\mathbb{E}$ step

$w = \hat{P}(\mathbf{Z}^m \mid \theta, \mathbf{Z})$	Bag	F	W
$P_\theta(Bag = 1 \mid F = c, W = r)$	1	c	r
$P_\theta(Bag = 2 \mid F = c, W = r)$	2	c	r
1	1	l	r
$P_\theta(W = r \mid Bag = 1, F = c)$	1	c	r
$P_\theta(W = g \mid Bag = 1, F = c)$	1	c	g
1	1	c	g
$P_\theta(Bag = 1, W = r \mid F = l)$	1	l	r
$P_\theta(Bag = 1, W = g \mid F = l)$	1	l	g
$P_\theta(Bag = 0, W = r \mid F = l)$	2	l	r
$P_\theta(Bag = 0, W = g \mid F = l)$	2	l	g

M step – update θs

$$\theta_{Bag=1} \leftarrow \frac{\sum_{Bag=1} w}{\sum w}$$

$$\theta_{F=c|Bag=1} \leftarrow \frac{\sum_{Bag=1,F=c} w}{\sum_{Bag=1} w}$$

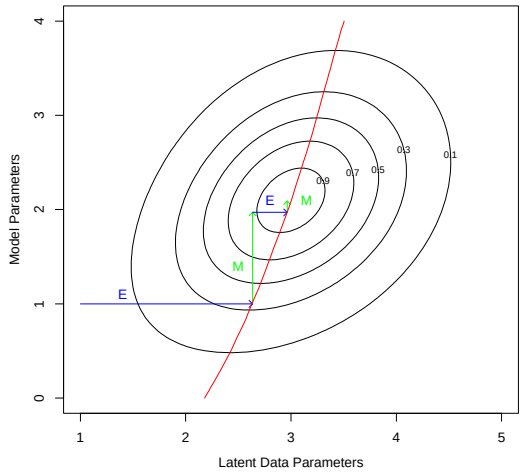
$$\theta_{F=c|Bag=2} \leftarrow \frac{\sum_{Bag=2,F=c} w}{\sum_{Bag=2} w}$$

$$\theta_{W=r|Bag=1} \leftarrow \frac{\sum_{Bag=1,W=r} w}{\sum_{Bag=1} w}$$

$$\theta_{W=r|Bag=2} \leftarrow \frac{\sum_{Bag=2,W=r} w}{\sum_{Bag=2} w}$$

EM as a Maximization-Maximization Procedure

- $\mathbf{Z}$ the observed data (the usual X with missing values)
- $\ell(\theta; \mathbf{Z})$ the log–likelihood of the model θ
- $\mathbf{Z}^m$ the latent or missing data
- $T = (\mathbf{Z}, \mathbf{Z}^m)$ the complete data with the log–likelihood $\ell_0(\theta; \mathbf{T})$.
- $\hat{P}(\mathbf{Z}^m), \hat{P}(\mathbf{Z}^m|\theta, \mathbf{Z})$ any distribution over the latent data $\mathbf{Z}^m$.
- Consider the function F



$$F(\theta', \hat{P}) = \mathbb{E}_{\hat{P}}[\ell_0(\theta'; (\mathbf{Z}, \mathbf{Z^m}))] - \mathbb{E}_{\hat{P}}[log\, \hat{P}(\mathbf{Z}^m)]$$

- for $\hat{P} = \hat{P}(\mathbf{Z}^m|\theta, \mathbf{Z})$ is F the log–likelihood of the observed data
  - $F(\theta, \hat{P}(\mathbf{Z}^m|\theta, \mathbf{Z})) = \mathbb{E}[\ell_0(\theta; (\mathbf{Z}, \mathbf{Z^m}))|\theta, \mathbf{Z}] - \mathbb{E}[\ell_1(\theta; \mathbf{Z}^m|\mathbf{Z})|\theta, \mathbf{Z}]$

EM in brief

E step fit $\mathbf{Z}^m$ (evaluate expected values)

M step fit θ' (maximize log likelihood given expectation on $\mathbf{Z}, \mathbf{Z}^m$)

The EM Algorithm in General×

$$
\begin{aligned}
P(\mathbf{Z}^m|\mathbf{Z},\theta') &= \frac{P(\mathbf{Z}^m,\mathbf{Z}|\theta')}{P(\mathbf{Z}|\theta')},\\
P(\mathbf{Z}|\theta') &= \frac{P(\mathbf{Z}^m,\mathbf{Z}|\theta')}{P(\mathbf{Z}^m|\mathbf{Z},\theta')},
\end{aligned}
$$

- In the log–likelihoods

$$\ell(\theta';\mathbf{Z}) = \ell_0(\theta';\mathbf{T}) - \ell_1(\theta';\mathbf{Z}^m|\mathbf{Z})$$

- where ℓ_1 is based on the conditional density $P(\mathbf{Z}^m|\mathbf{Z})$.
- Taking the expectation w.r.t. $\mathbf{T}|\mathbf{Z}$ governed by parameter θ gives

$$
\begin{aligned}
\ell(\theta';\mathbf{Z}) &= \mathbb{E}[\ell_0(\theta';\mathbf{T})|\theta,\mathbf{Z}] - \mathbb{E}[\ell_1(\theta';\mathbf{Z}^m|\mathbf{Z})|\theta,\mathbf{Z}]\\
&\equiv Q(\theta',\theta) - R(\theta',\theta)
\end{aligned}
$$

- $R()$ is the expectation of a density with respect the same density
  - it is maximized when $\theta' = \theta$.
- Therefore:

$$
\begin{aligned}
\ell(\theta';\mathbf{Z}) - \ell(\theta;\mathbf{Z}) &= [Q(\theta',\theta) - Q(\theta,\theta)] - [R(\theta',\theta) - R(\theta,\theta)]\\
&\geq 0.
\end{aligned}
$$

The EM Algorithm

procedure THE EM ALGORITHM:(**Z** observed data, the model(θ))
  $\hat{\theta}^{(0)} \leftarrow$ an initial guess (usually close to the uniform distribution)
  repeat
    Expectation step: at the jth step, compute

$$Q(\theta', \hat{\theta}^{(j)}) = \mathbb{E}(\ell_0(\theta'; \mathbf{T})|Z, \hat{\theta}^{(j)})$$

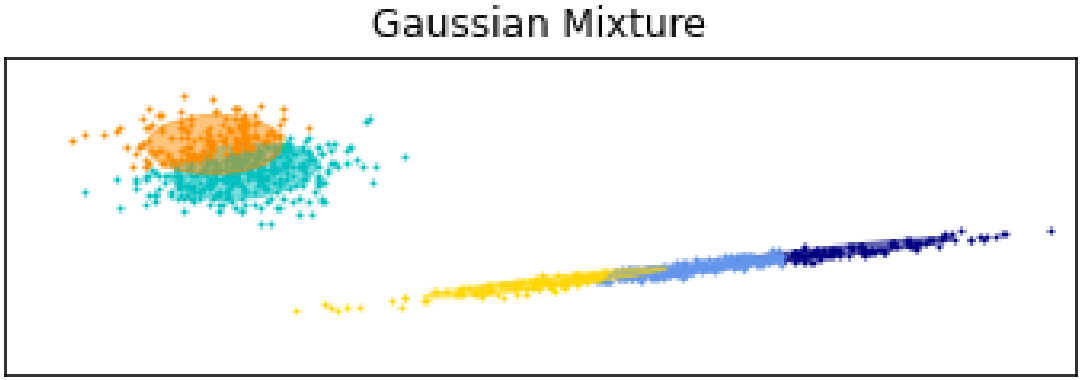
      as a function of the dummy argument θ'.
      usually 'weights' on samples with the 'fill-in' values.
    Maximization step: determine the new estimate $\hat{\theta}^{(j+1)}$
      as the maximizer of $Q(\theta', \hat{\theta}^{(j)})$ over θ'.
  until convergence
  return $\hat{\theta}$
end procedure

- Full maximization is not necessary.
- We need to find a value $\hat{\theta}^{(j+1)}$ so that $Q(\hat{\theta}^{(j+1)}, \hat{\theta}^{(j)}) > Q(\hat{\theta}^{(j)}, \hat{\theta}^{(j)})$.
- Such procedures are called **generalized EM algorithms (GEM)**.

Gaussian Mixture Model for Clustering

- We assume the Gaussian Mixture Model
  - like a Naive Bayes Model
  - but the 'Class' variable represents the cluster and is latent, 'missing'
- We use EM algorithm to estimate the 'Cluster' variable.
- sklearn example

```
from sklearn.mixture import BayesianGaussianMixture
```



ML Estimate of Gaussian Distribution Parameters

- Assume x to have Gaussian distribution with unknown parameters μ a σ.
- Our hypotheses are $h_{\mu,\sigma} = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$.
- We have observed $x_1, \ldots, x_n$.
- Log likelihood is:

$$\begin{aligned} LL &= \sum_{j=1}^{N} log \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \\ &= N \cdot \left(log \frac{1}{\sqrt{2\pi}\sigma} \right) - \sum_{j=1}^{N} \frac{(x_j - \mu)^2}{2\sigma^2} \end{aligned}$$

- Find the maximum.

Linear Gaussian Distribution

- Assume random variable (feature) X.
- Assume goal variable Y with linear Gaussian distribution where $\mu = b \cdot x + b_0$ and fixed variance σ^2 $p(Y|X = x) = N(b \cdot x + b_0; \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(y-((b \cdot x + b_0))^2}{2\sigma^2}}$.
- Find maximum likelihood estimate of b, b_0 given a set of observations $data = \{\langle x_1, y_1 \rangle, \ldots, \langle x_N, y_N \rangle\}$.
- (Look for maximum of the logarithm of it; change the max to min with the opostite sign. Do you know this formula?)

$$argmax_{b,b_0}(log_e(\Pi_{i=1}^{N}(e^{-(y_i-(b \cdot x_i+b_0))^2}))) = argmin_{b,b_0}(?)$$

EM learning of Mixture of K Gaussians !

- Model parameters $\pi_1, \ldots, \pi_k, \mu_1, \ldots, \mu_k, \Sigma_1, \ldots, \Sigma_k$ such that $\sum_{k=1}^{K} \pi_k = 1$.
- **E**xpectation: weights of unobserved 'fill–ins' k of variable C:

$$\begin{aligned} p_{ik} &= P(C = k|x_i) = \alpha \cdot P(x_i|C_i = k) \cdot P(C_i = k) \\ &= \frac{\pi_k \phi_{\theta_k}(x_i)}{\sum_{l=1}^{K} \pi_l \phi_{\theta_l}(x_i)} \\ p_k &= \sum_{i=1}^{N} p_{ik} \end{aligned}$$

- **M**aximize: mean, variance and cluster 'prior' for each cluster k:

$$\begin{aligned} \mu_k &\leftarrow \sum_i \frac{p_{ik}}{p_k} x_i \\ \Sigma_k &\leftarrow \sum_i \frac{p_{ik}}{p_k} (x_i - \mu_k)(x_i - \mu_k)^T \\ \pi_k &\leftarrow \frac{p_k}{\sum_{l=1}^{K} p_l}. \end{aligned}$$

BN example of EM algorithm (Russel, Norvig) - can be omitted

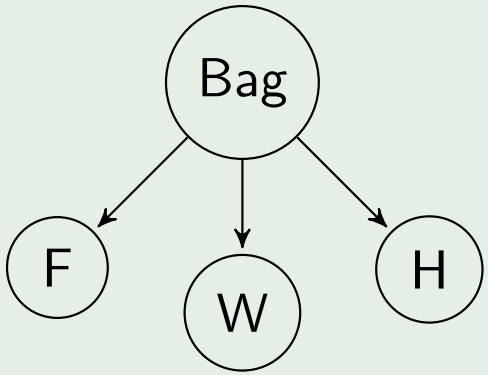
- Two bags of bonbons mixed together. Each bonbon has a *Wrapper* and flavor *Flavor* and may have *Holes*. Each bag had another ratio of *Wrapper* color, *Flavor* and *Holes*.

We can model the situation by a naive bayes model, *Bag* as the class variable.

Example

Example We have tested 1000 bonbones and observed:

	W=red		W=green	
	H=1	H=0	H=1	H=0
F=cherry	273	93	104	90
F=lime	79	100	94	167



We choose the initial parameters

$$\theta^{(0)} = 0.6,\ \theta^{(0)}_{F1} = \theta^{(0)}_{W1} = \theta^{(0)}_{H1} = 0.6,\ \theta^{(0)}_{F2} = \theta^{(0)}_{W2} = \theta^{(0)}_{H2} = 0.4$$

EM example - can be omitted

- Expectation of θ is the ratio of the expected counts

$$\theta^{(1)} = \frac{1}{N}\sum_{j=1}^{N} \frac{P(flavor_j|Bag=1)P(wrapper_j|Bag=1)P(holes_j|Bag=1)P(Bag=1)}{\sum_{i=1}^{2} P(flavor_j|Bag=i)P(wrapper_j|Bag=i)P(holes_j|Bag=i)P(Bag=i)}$$

(normalization constant **depends** on parameter values).

For the type *red*, *cherry*, *holes* we get:

$$\frac{\theta_{F1}^{(0)}\theta_{W1}^{(0)}\theta_{H1}^{(0)}\theta^{(0)}}{\theta_{F1}^{(0)}\theta_{W1}^{(0)}\theta_{H1}^{(0)}\theta^{(0)} + \theta_{F2}^{(0)}\theta_{W2}^{(0)}\theta_{H2}^{(0)}\theta^{(0)}} \approx 0.835055$$

we have 273 bonbons of this type, therefore we add $\frac{273}{N} \cdot 0.835055$.
Similarly for all seven other types and we get

$$\theta^{(1)} = 0.6124$$

EM example continued - can be omitted

- The estimate of θ_{F1} for fully observed data is $\frac{\#(Bag{=}1, Flavor{=}cherry)}{\#(Flavor{=}cherry)}$
- We have to use expected counts $Bag = 1 \& F = cherry$ and $Bag = 1$,

$$\theta_{F1}^{(1)} = \frac{\sum_{j;Flavor_j=cherry} P(Bag = 1 | Flavor_j = cherry, wrapper_j, holes_j)}{\sum_j P(Bag = 1 | cherry_j, wrapper_j, holes_j)}$$

- Similarly we get:

$$\theta^{(1)} = 0.6124,\ \theta_{F1}^{(1)} = 0.6684, \theta_{W1}^{(1)} = 0.6483, \theta_{H1}^{(1)} = 0.6558,$$

$$\theta_{F2}^{(1)} = 0.3887, \theta_{W2}^{(1)} = 0.3817, \theta_{H2}^{(1)} = 0.3827.$$

Summary

Complicated derivation of known things.

- **Maximum a posteriori probability hypothesis** (MAP) (nejpravděpodobnější hypotéza)
- **Maximum likelihood hypothesis** (ML) (maximálně věrohodná hypotéza)
- **Bayesian optimal prediction** (Bayes Rate)
- **Bayesian methods**, **bayesian smoothing**
  - 'complexity penalty' is our prior distribution/preference on parameters.
- **Naive Bayes model (classifier)**.
- **EM algorithm**
  - The best way to fill in missing values
  - the most common application is clustering
  - but the use is far broader, for example
  - Baum-Welch algorithm for HMM
  - variational approximation for continuous distributions.

MAP, Bayesian Smoothing and Penalized Methods

- The complexity penalty (Lasso, Ridge, ...) can be explained as our prior distribution on parameters (hypotheses) $P(h)$ with a higher probability for more simple models.
- MAP hypothesis maximizes:

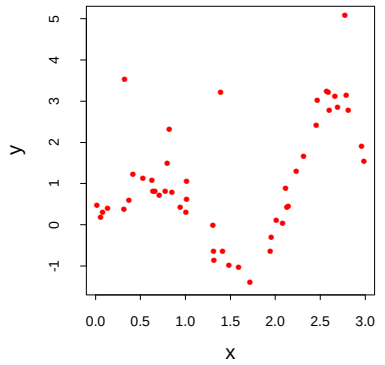
$$h_{MAP} = argmax_i P(data|h_i) \cdot P(h_i)$$

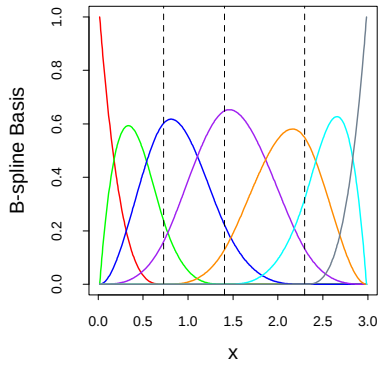
- therefore minimizes:

$$\begin{aligned} h_{MAP} &= argmax_h P(data|h)P(h) \\ &= argmin_h[-log_2 P(data|h) - log_2 P(h)] \\ &= argmin_h[-loglik + \text{complexity penalty}] \\ &= argmin_h[RSS + \text{complexity penalty}] \text{ Gaussian models} \\ &= argmax_h[loglik - \text{complexity penalty}] \text{ Categorical models} \end{aligned}$$

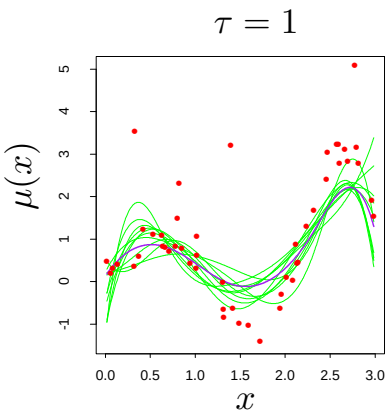
Bayesian smoothing example

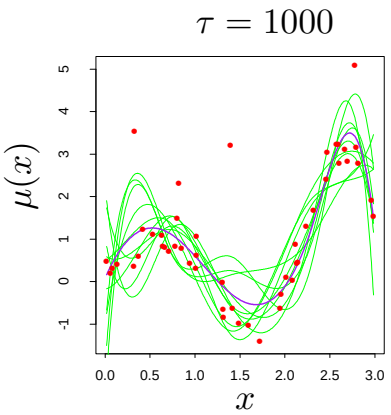
- Training data $\mathbf{Z} = \{z_i, \ldots, z_N\}$, $z_i = (x_i, y_i)$, $i = 1, \ldots, N$.
- We look for a cubic spline with three knots in quartiles of the X values. It corresponds to B-spline basis $h_j(x)$, $j = 1, \ldots, 7$.





- We estimate the conditional mean $\mathbb{E}(Y|X = x)$: $\mu(x) = \sum_{j=1}^{7} \beta_j h_j(x)$
- Let $\mathbf{H}$ be the $N \times 7$ matrix $h_j(x_i)$.
- RSS β estimate is $\hat{\beta} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{y}$.





We assume to know σ^2, fixed x_i, we specifying prior on $\beta \sim N(0, \tau\Sigma)$.

$$\mathbb{E}(\beta|\mathbf{Z}) = (\mathbf{H}^T\mathbf{H} + \frac{\sigma^2}{\tau}\Sigma^{-1})^{-1}\mathbf{H}^T\mathbf{y}$$

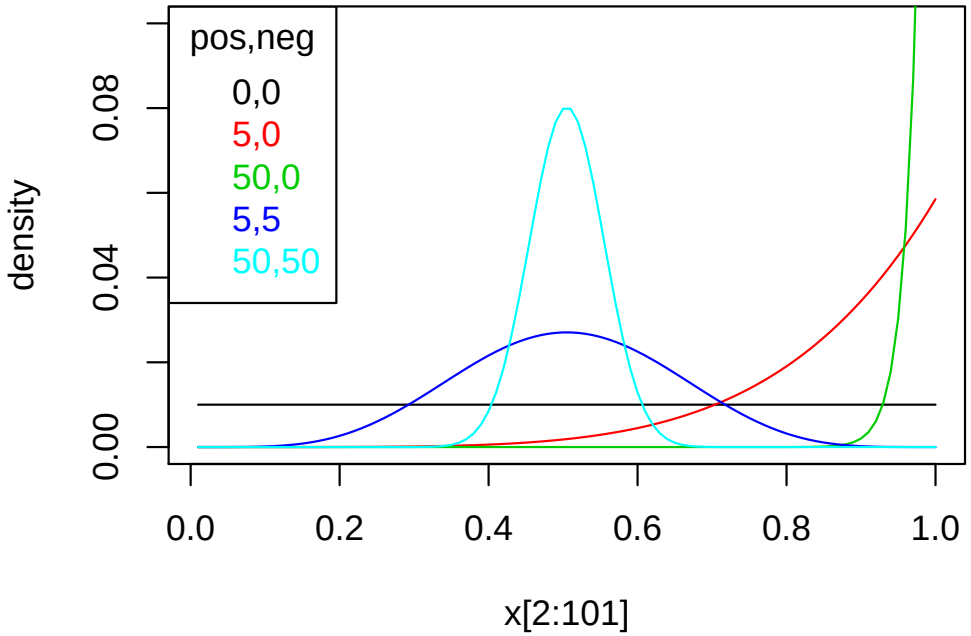
$$\mathbb{E}(\mu(x)|\mathbf{Z}) = h(x)^T(\mathbf{H}^T\mathbf{H} + \frac{\sigma^2}{\tau}\Sigma^{-1})^{-1}\mathbf{H}^T\mathbf{y}.$$

Parameter Estimate as a Probability Distribution

- For binary features, Beta function is used, $(a-1)$ is the number of positive examples, $(b-1)$ the number of negative examples.

$$beta[a,b](\theta) = \alpha\theta^{a-1}(1-\theta)^{b-1}$$

- Beta Function:



- For categorical features, Dirichlet priors and multinomial distribution is used. (Dirichlet-multinomial distribution).
- For Gaussian, μ has Gaussian prior, $\frac{1}{\sigma}$ has gamma prior (to stay in exponential family).

Bayesian Methods

- We specify a sampling model $P(\mathbf{Z}|\theta)$
- and a prior distribution for parameters $P(\theta)$
- then we compute

$$P(\theta|\mathbf{Z}) = \frac{P(\mathbf{Z}|\theta) \cdot P(\theta)}{\int P(\mathbf{Z}|\theta) \cdot P(\theta) d\theta},$$

  - we may draw samples
  - or summarize by the mean or mode.
  - it provides the **Bayesian optimal predictive distribution**:

$$P(z^{new}|\mathbf{Z}) = \int P(z^{new}|\theta) \cdot P(\theta|\mathbf{Z}) d\theta.$$

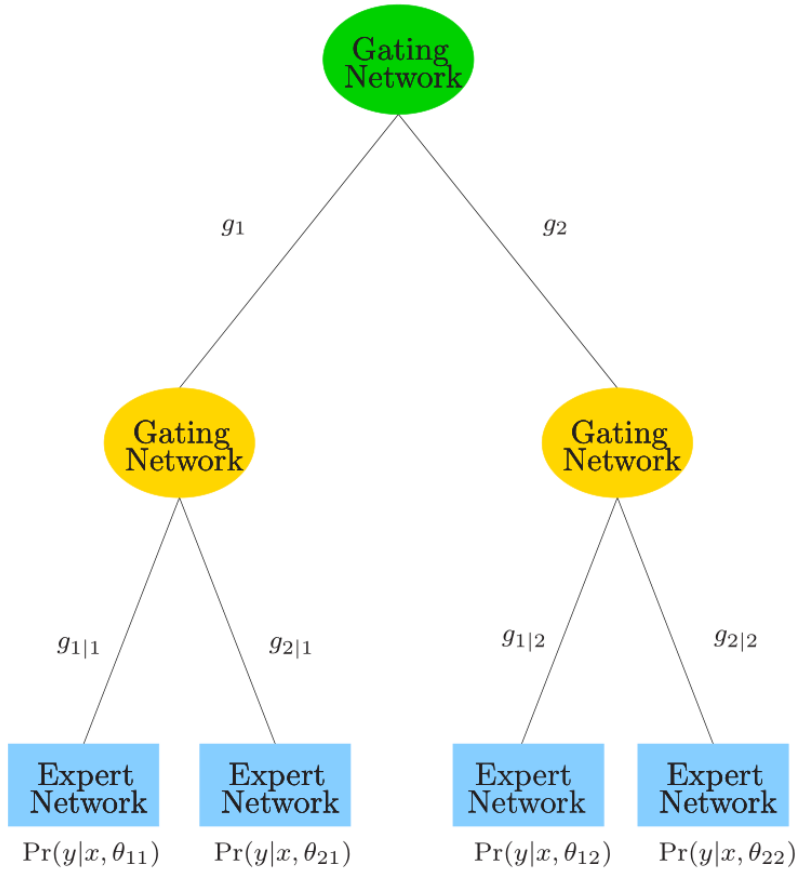
Example (Previous slide)

Tossing a biased coin

- $P(Z = head|\theta) = \theta$
- $p(\theta)$ =uniform
- $P(\theta|\mathbf{Z})$ follows the Beta distribution.

Hierarchical Mixture of Experts

- a hierarchical extension of naive Bayes (latent class model)
- a decision tree with 'soft splits'
- splits are probabilistic functions of a linear combination of inputs (not a single input as in CART)
- terminal nodes called 'experts'
- non–terminal nodes are called gating network
- may be extended to multilevel.

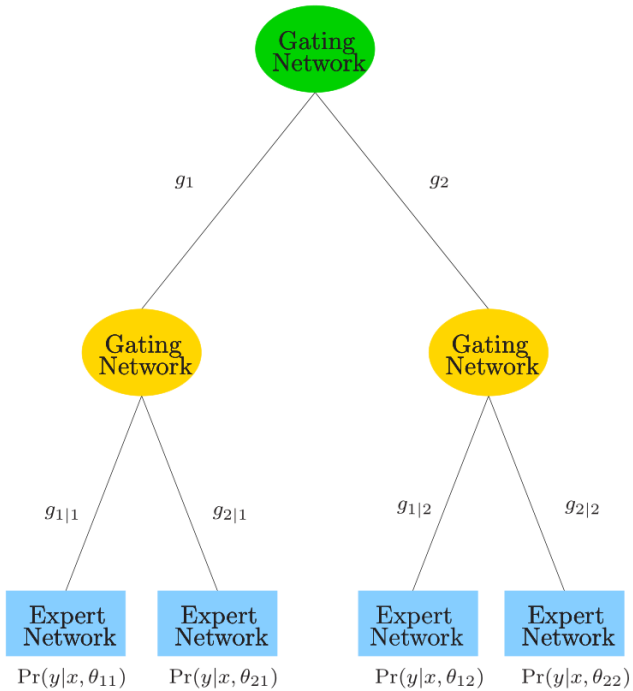


Hierarchical Mixture of Experts

- data (x_i, y_i), $i = 1, \ldots, N$, y_i continuous or categorical, first $x_i \equiv 1$ for intercepts.
- $g_i(x, \gamma_j) = \frac{e^{\gamma_j^T x}}{\sum_{k=1}^{K} e^{\gamma_k^T x}}$, $j = 1, \ldots, K$ children of the root,
- $g_{\ell|j}(x, \gamma_{j\ell}) = \frac{e^{\gamma_{j\ell}^T x}}{\sum_{k=1}^{K} e^{\gamma_{jk}^T x}}$, $\ell = 1, \ldots, K$ children of the root,
- Terminals (Experts)

Regression Gaussian linear reg. model,
$\theta_{j\ell} = (\beta_{j\ell}, \sigma^2_{j\ell})$, $Y = \beta^T_{j\ell} + \epsilon$

Classification The linear logistic reg. model:
$Pr(Y = 1|x, \theta_{j\ell}) = \frac{1}{1+e^{-\theta^T_{j\ell} x}}$



- EM algorithm
- Δ_i, $\Delta_{\ell|j}$ 0–1 latent variables – branching

E step expectations for Δ's

M step estimate parameters HME by a version of multiple logistic

Missing data (T.D. Nielsen)

Die tossed N times. Result reported via noisy telephone line. When transmission not clearly audible, record missing value:

$$4, 2, ?, 6, 5, 4, ?, 3, 4, 1, \ldots$$

"2" and "3" sound similar, therefore:

$$P(Y_i = ?|X_i = k) = P(M_i = 1|X_i = k) = \begin{cases} 1/4 & \text{k} = 2,3 \\ 1/8 & \text{k} = 1,4,5,6 \end{cases}$$

Distribution of the Y is (for fair die):

?	$\frac{1}{3}\frac{1}{4} + \frac{2}{3}\frac{1}{8} = \frac{1}{6}$
2,3	$\frac{1}{6}\frac{3}{4} = \frac{1}{8}$
1,4,5,6	$\frac{1}{6}\frac{7}{8} = \frac{7}{48}$

If we simply ignore the missing data items, we obtain as the maximum likelihood estimate for the parameters of the die:

$$\theta^* = (\frac{7}{48}, \frac{1}{8}, \frac{1}{8}, \frac{7}{48}, \frac{7}{48}, \frac{7}{48}) * \frac{6}{5} = (0.175, 0.15, 0.15, 0.175, 0.175, 0.175)$$

Incomplete data

How do we handle cases with missing values:

- Faulty sensor readings.
- Values have been intentionally removed.
- Some variables may be unobservable.

How is the data missing?
We need to take into account how the data is missing:

- **Missing completely at random** The probability that a value is missing is independent of both the observed and unobserved values (a monitoring system that is not completely stable and where some sensor values are not stored properly).
- **Missing at random** The probability that a value is missing depends only on the observed values (a database containing the results of two tests, where the second test has only performed (as a "backup test") when the result of the first test was negative).
- **Non-ignorable** Neither MAR nor MCAR (an exit poll, where an extreme right-wing party is running for parlament).

Decision Rules from Decision Trees

- We can represent a tree as a set of rules
  - one rule for each leaf.
- These rules may be improved by testing each attribute in each rule
  - Has the rule without this test a better precision than with the test?
  - Use validation data
  - May be time consuming.
- These rules are sorted by (decreasing) precision.

Patient Rule Induction Method PRIM = Bump Hunting

- Rule induction method
- We iteratively search regions with the high Y values
  - for each region, a rule is created.
- CART runs of data after aproximately $\log_2(N) - 1$ cuts.
- PRIM can affort $-\frac{\log(N)}{\log(1-\alpha)}$.
  For $N = 128$ data samples and $\alpha = 0.1$ it is 6 and 46 respectively 29, since the number of observations must be a whole number.

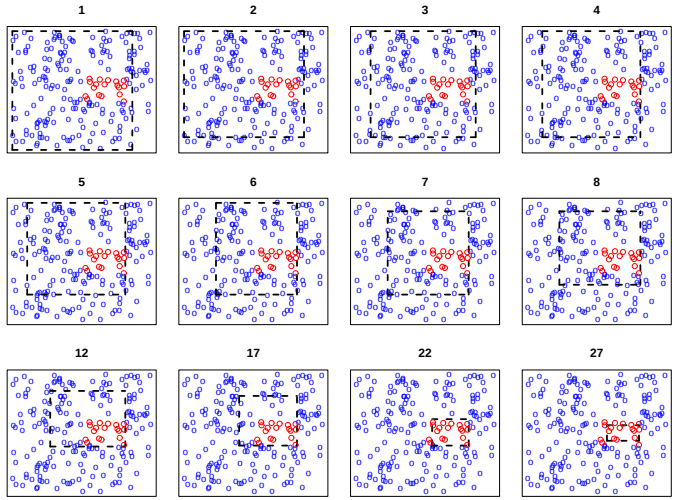


FIGURE 9.7. *Illustration of PRIM algorithm. There are two classes, indicated by the blue (class 0) and red (class 1) points. The procedure starts with a rectangle (broken black lines) surrounding all of the data, and then peels away points along one edge by a prespecified amount in order to maximize the mean of the points remaining in the box. Starting at the top left panel, the sequence of peelings is shown, until a pure red region is isolated in the bottom right panel. The iteration number is indicated at the top of each panel.*

PRIM Patient Rule induction Algorithm

PRIM

- Consider the whole space and all data. Set $\alpha = 0.05$ or 0.10.
- Find X_j and its upper or lower boundary such that the cut of $\alpha \cdot 100\%$ observations leads to the maximal mean of the remaining data.
- Repeat until less then 10 observations left.
- Enlarge the region in any direction that increases the mean value.
- Select the number of regions by the crossvalidation. All regions generated 1-4 are considered.
- Denote the best region B_1.
- Create a rule that describes B_1.
- Remove all data in B_1 from the dataset.
- Repeat 2-5, create B_2 continue until final condition met.

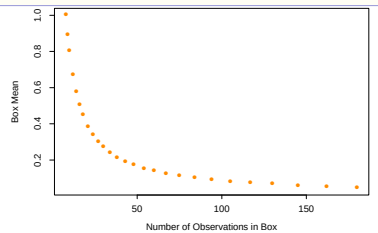


FIGURE 9.8. *Box mean as a function of number o observations in the box.*

CART Weaknesses

- the high variance
  - the tree may be very different for very similar datasets
  - ensemble learning addresses this issue
- the cuts are perpendicular to the axis
- the result is not smooth but stepwise.
  - MARS (Multivariate Adaptive Regression Splines) addresses this issue.
- it is difficult to capture an additive structure

$$Y = c_1 I(X_1 < t_1) + c_2 I(X_2 < t_2) + \ldots + c_k I(X_k < t_k) + \epsilon$$

  - MARS (Multivariate Adaptive Regression Splines) addresses this issue.

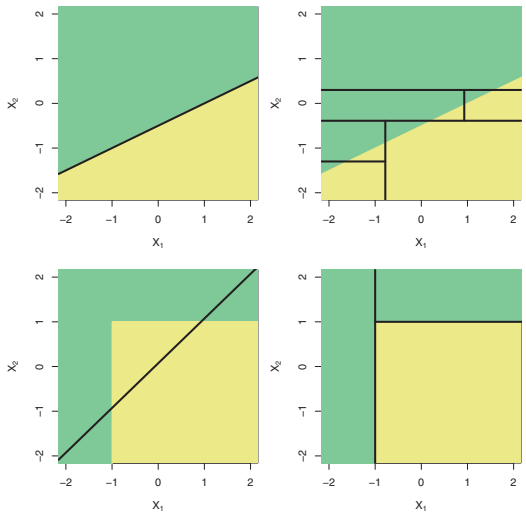


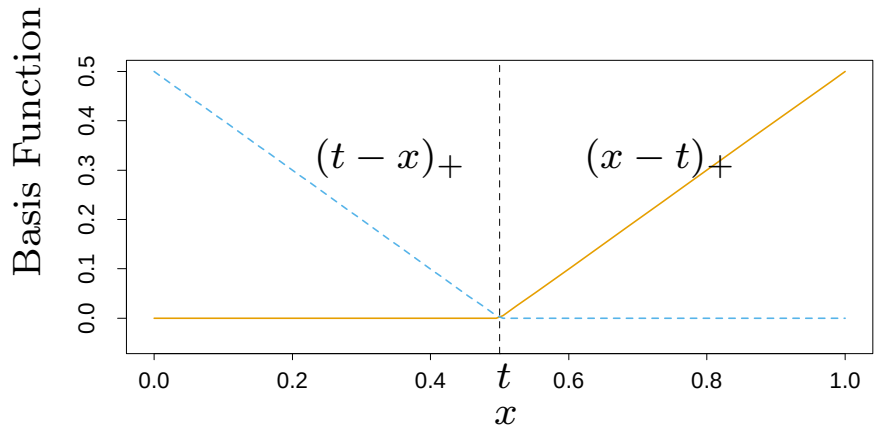
FIGURE 8.7. Top Row: *A two-dimensional classification example in which the true decision boundary is linear, and is indicated by the shaded regions. A classical approach that assumes a linear boundary (left) will outperform a de-*

MARS Multivariate Adaptive Regression Splines

- generalization of linear regression and decision trees CART
- for each feature and each data point we create a **reflected pair** of basis functions
- $(x - t)_+$ and $(t - x)_+$ where $+$ denotes non–negative part, minimum is zero.
- we have the set of functions

$$\mathcal{C} = \{(X_j - t)_+, (t - X_j)_+\}_{t \in \{x_{1,j}, x_{2,j}, \ldots, x_{N,j}\}, j=1,2,\ldots,p}$$

- that is $2Np$ functions for non–duplicated data points.

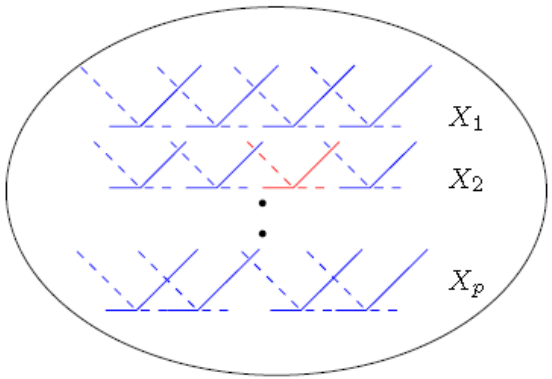


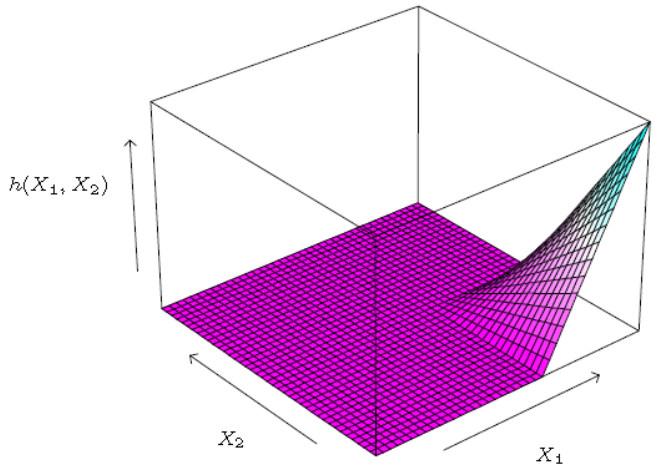
MARS – continuation

- our model is in the form

$$f(X) = \beta_0 + \sum_{m=1}^{M} \beta_m h_m(X)$$

  where $h_m(X)$ is a function from $\mathcal{C}$ or a product of any amount of functions from $\mathcal{C}$
- for a fixed set of h_m's we calculate coefficients β_m by usual linear regression (minimizing RSS)
- the set of functions h_m is selected iteratively.



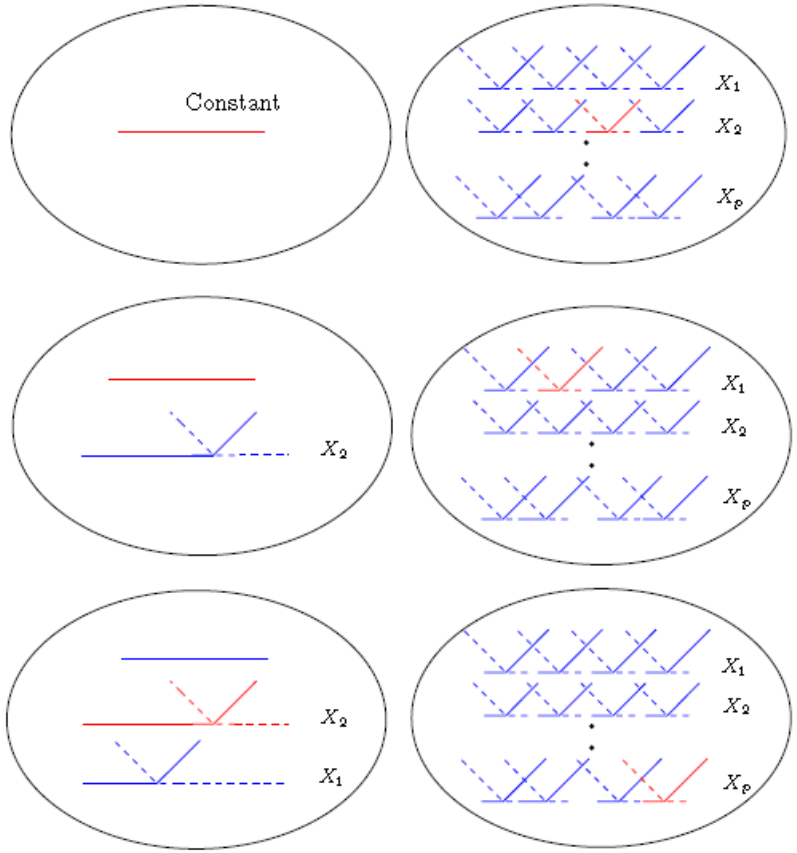


MARS – basis selections

- We start with $h_0 = 1$, we put this function into the model $\mathcal{M} = \{h_0\}$.
- We consider the product of any member $h_\ell \in \mathcal{M}$ with any pair from $\mathcal{C}$,

  $$\hat{\beta}_{M+1} h_\ell(X) \cdot (X_j - t)_+ + \hat{\beta}_{M+2} h_\ell(X) \cdot (t - X_j)_+$$

  we select the one minimizing training error RSS (for any product candidate, we estimate $\hat{\beta}$).
- Repeat until predefined number of functions in $\mathcal{M}$



MARS – model pruning

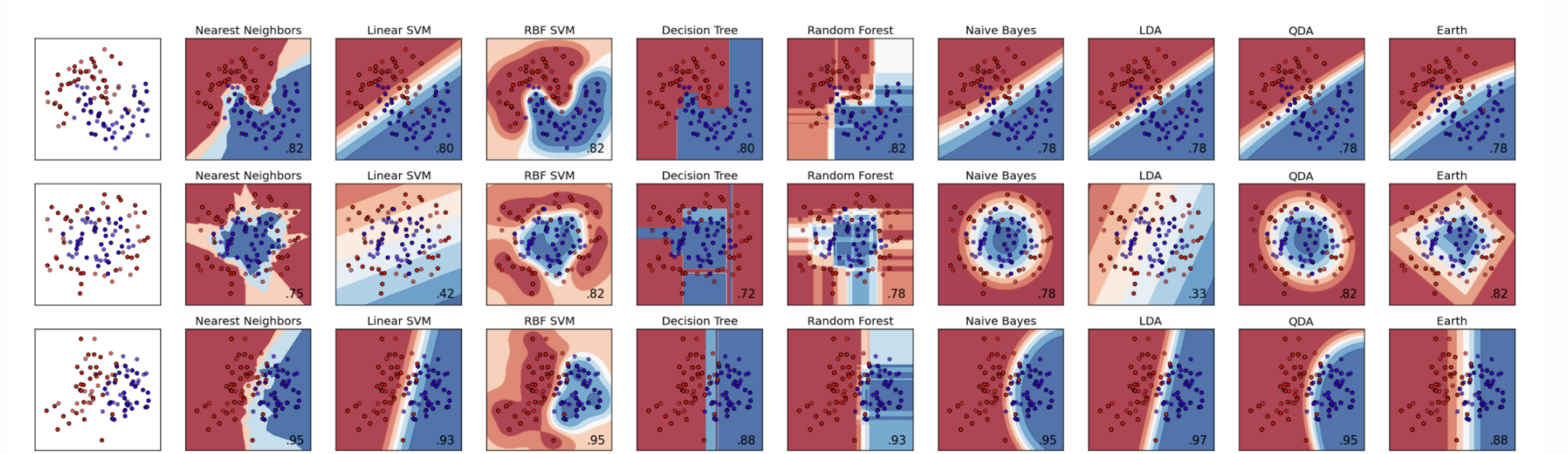
- The model is usually overfitted. We select (remove) iteratively the one minimizing the increase of training RSS. We have a sequence of models $\hat{f}_\lambda$ for different numbers of parameters λ.
- (we want to speed–up cross-validation for computational reasons)
- we select λ (and the model) minimizing **generalized cross-validation**

$$GCV(\lambda) = \frac{\sum_{i=1}^{N}(y_i - \hat{f}_\lambda(x_i))^2}{(1 - M(\lambda)/N)^2}.$$

- where $M(\lambda)$ is the number of effective parameters, the number of function h_m (denoted r) plus the number of knots K, the authors suggest to multiply K by 3: $M(\lambda) = r + 3K$.

MARS is a generalization of CART

- We select piecewise constant functions $I(x - t > 0)$ and $I(x - t \leq 0)$
- If h_m uses multiplication we remove this function from the candidate list. It cannot be used any more.
  - This guarantees binary split.
- Its CART.



https://contrib.scikit-learn.org/py-earth/auto_examples/plot_classifier_comp.html

https://contrib.scikit-learn.org/py-earth/auto_examples/index.html

List of topics

1. Linear, ridge, lasso regression, k-nearest neighbors, overfitting, curse of dimensionality, (LARS)
2. Splines - the base, natural splines, smoothing splines; kernel smoothing: kernel average, Epanechnikov kernel.
3. Logistic regression, Linear discriminant analysis, (generalized additive models)
4. Train/test error and data split, square error, 0-1, cross-entropy, AIC, BIC, cross-validation, one-leave-out CV, biased estimate example
5. decision trees, information gain/entropy/gini, CART α pruning
6. random forest (+bagging), OOB error, Variable importance, boosting (Adaboost and gradient boosting), stacking, (MARS),
7. Bayesian learning: MAP, ML hypothesis, Bayesian optimal prediction !formulas!, EM algorithm
8. Clustering: k-means, Silhouette, k-medoids, hierarchical
9. Apriori algorithm, Association rules, support, confidence, lift
10. Inductive logic programming basic: hypothesis space search, background knowledge, necessity, sufficiency and consistency of a hypothesis, Aleph
11. Undirected graphical models, Graphical Lasso procedure, (deviance, MRF)
12. Gaussian processes: estimation of the function and its variance !figures, ideas!
13. Support Vector Machines (optimal separating hyperplane, slacks, kernels).

Table of Contens

1. Basic Notions and Linear Models for Regression
2. Kernel Methods, Basis Expansion and regularization
3. Linear Methods for Classification
4. Model Assessment and Selection
5. Additive Models, Trees, and Related Methods
6. Ensamble Methods
7. Bayesian learning, EM algorithm
8. Clustering
9. Association Rules, Apriori
10. Inductive Logic Programming
11. Undirected Graphical Models
12. Gaussian Processes
13. (Support Vector Machines)
14. Further Models