

# Suboptimality of Bayes and MDL in Classification

Peter Grünwald  
CWI/EURANDOM  
[www.grunwald.nl](http://www.grunwald.nl)

*joint work with John Langford, Toyota Technological Institute,  
Chicago, [www.hunch.net/~jl](http://www.hunch.net/~jl)*

# Our Result

- Bayesian and Minimum Description Length (MDL) inference are popular methods for machine learning
- Especially suitable for dealing with **overfitting**
- Arguably, most studied problem in ML is **classification**
- We show there exist classification domains where Bayes and MDL...

***when applied in a standard manner***

...perform suboptimally even if sample size tends to infinity

# Why is this interesting?

- Practical viewpoint:
  - Bayesian methods are used a *lot* in practice
  - Bayesian methods are sometimes claimed to be ‘universally asymptotically optimal’
  - Our result shows that MDL and Bayes can ‘fail’ even with infinite data
- Theoretical viewpoint
  - How can result be reconciled with various strong Bayesian *consistency* theorems?

# Menu

1. Classification
- 2.
3. Abstract statement of main result
- 4.
5. Bayesian learning for classification
- 6.
7. Precise statement of result
- 8.
9. Discussion

# Classification

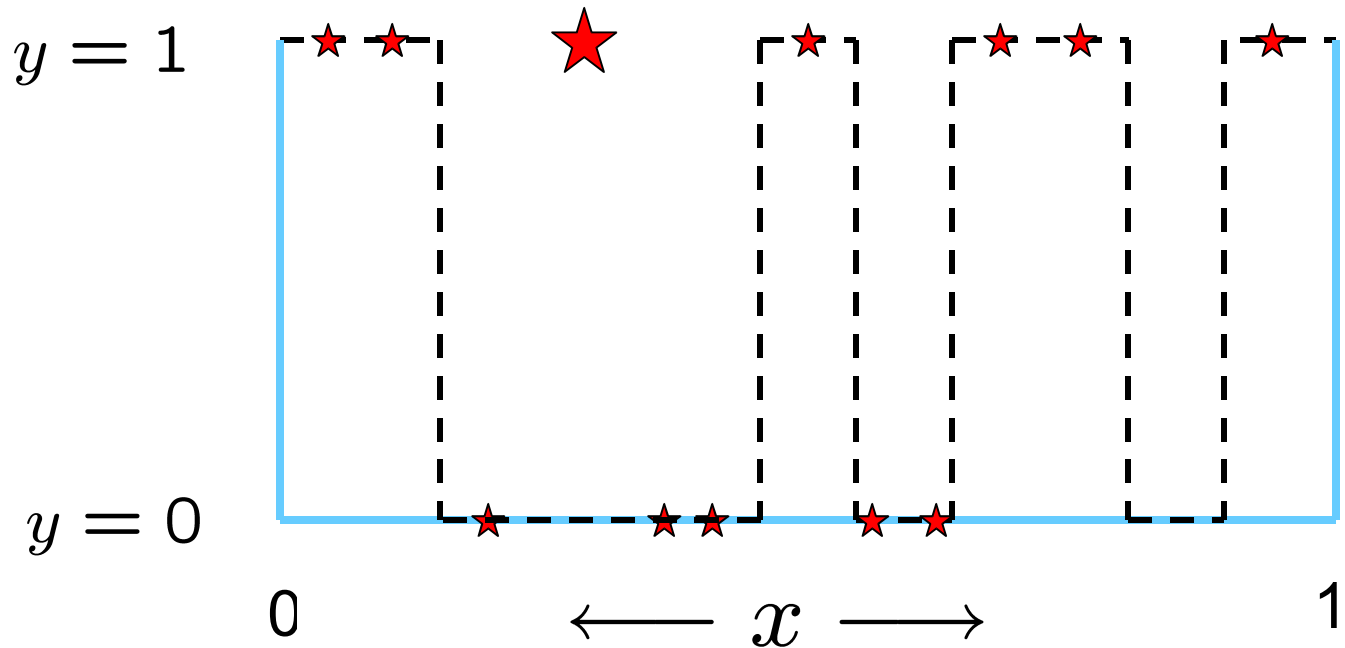
- Given:
  - Feature space  $\mathcal{X}$
  - Label space  $\mathcal{Y} = \{-1, 1\}$
  - Sample  $S = (x_1, y_1), \dots, (x_m, y_m)$
  - Set  $\mathcal{C}$  of hypotheses (**classifiers**)  $c : \mathcal{X} \rightarrow \mathcal{Y}$
- Goal: find a  $c \in \mathcal{C}$  that makes few mistakes on future data from the same source
  - We say ‘ $c$  has small **generalization error**’
  - if  $\mathcal{C}$  is ‘large’ (‘complex’), then it is **not** a good idea to adopt the  $c \in \mathcal{C}$  that minimizes nr of mistakes on the given data
  - leads to **over-fitting**

# Classification Models

- Typical classification models used in ML community:
  1. **hard** classifiers: (-1/1-output)
    - decision trees, stumps, forests
  2. **soft** classifiers (real-valued output)
    - support vector machines
    - neural networks
  3. **probabilistic** classifiers
    - Naïve Bayes/Bayesian network classifiers
    - Logistic regression

# Example: intervals (toy) domain

Kearns et al., 1995



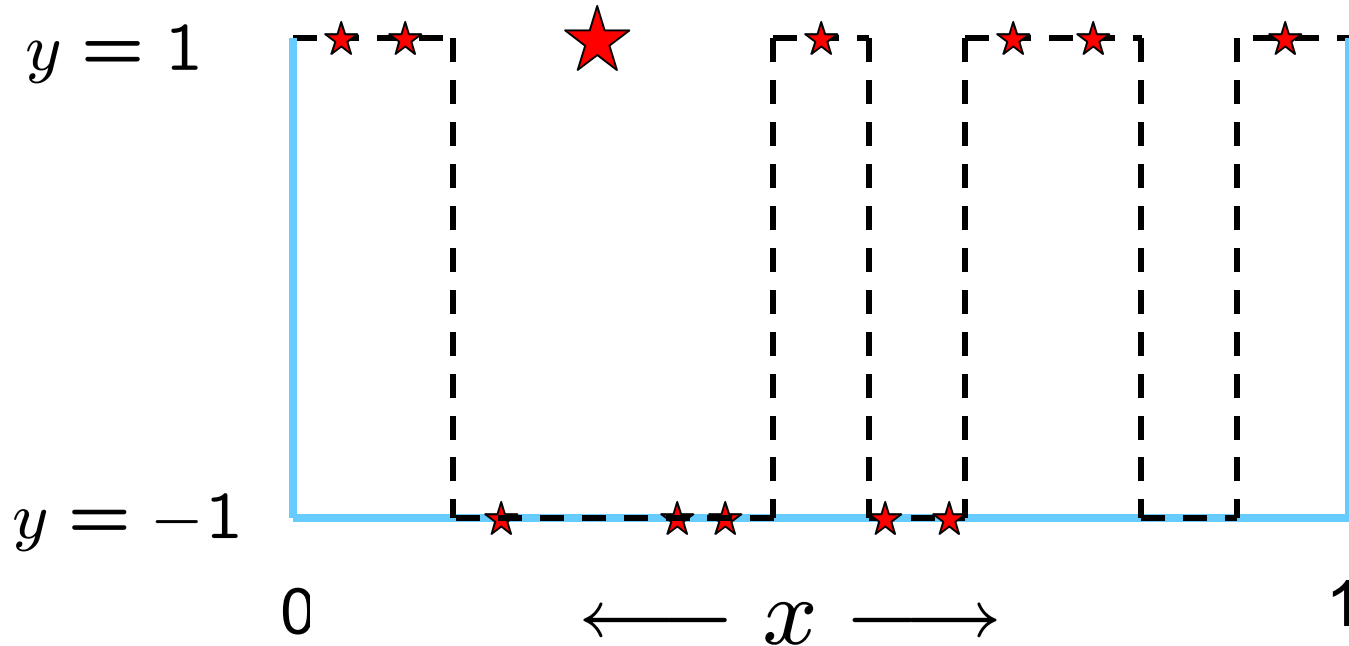
$$\mathcal{X} = [0, 1]$$

$\mathcal{C}_k$  : set of functions  $c : \mathcal{X} \rightarrow \{-1, 1\}$  that switch value  $k$  times

$$\mathcal{C} = \bigcup_{k=1,2,\dots} \mathcal{C}_k$$

# Example: intervals (toy) domain

Kearns et al., 1995



$$\mathcal{X} = [0, 1]$$

$\mathcal{C}_k$  : set of functions  $c : \mathcal{X} \rightarrow \{-1, 1\}$  that switch value  $k$  times

$$\mathcal{C} = \bigcup_{k=1,2,\dots} \mathcal{C}_k$$

The  $c$  in picture is in  $\mathcal{C}_6$  and makes 1 classification error on data  $D$

# Generalization Error

- As is customary, we analyze classification by postulating some (unknown) distribution  $D$  on joint (input,label)-space  $\mathcal{X} \times \mathcal{Y}$
- Generalization error defined as

$$\mathbf{e}_D(c) :=$$

$$\Pr_{(X,Y) \sim D}(Y \neq c(X)) = \frac{1}{2} \mathbf{E}_{(X,Y) \sim D} |Y - c(X)|.$$

# Consistent Learning Algorithms

- A learning algorithm LA based on set of candidate classifiers  $\mathcal{C}$  is a function that, for each sample of arbitrary length, outputs classifier  $c \in \mathcal{C}$

$$LA : \bigcup_{m \geq 0} (\mathcal{X} \times \mathcal{Y})^m \rightarrow \mathcal{C}$$

- A learning algorithm is **consistent** or **asymptotically optimal** if, *no matter what the ‘true’ distribution D is,*

$$\mathbf{e}_D(LA(S)) \rightarrow \min_{c \in \mathcal{C}} \mathbf{e}_D(c)$$

in D – probability, as  $m \rightarrow \infty$  .

# Consistent Learning Algorithms

- A learning algorithm LA based on set of candidate classifiers  $\mathcal{C}$  is a function that, for each sample of arbitrary length, outputs classifier  $c \in \mathcal{C}$

$$LA : \bigcup_{m \geq 0} (\mathcal{X} \times \mathcal{Y})^m \rightarrow \mathcal{C}$$

- A learning algorithm is **consistent** or **asymptotically optimal** if, *no matter what the 'true' distribution D is,*

$$\mathbf{e}_D(LA(S)) \rightarrow \min_{c \in \mathcal{C}} \mathbf{e}_D(c)$$

'learned' classifier

=  $\mathbf{e}_D(\tilde{c})$  where  $\tilde{c}$  is 'best' classifier

# Main Result

- There exists

- input domain  $\mathcal{X}$
- prior  $P$ , non-zero on a countable set of classifiers  $\mathcal{C}$
- ‘true’ distribution  $D$
- a constant  $K > 0$

such that the Bayesian learning algorithm  $\text{Bayes}(S, P)$  is **asymptotically K-suboptimal**:

$$\lim_{m \rightarrow \infty} \Pr_{S \sim D^m} \left( \mathbf{e}_D(\text{Bayes}(S, P)) > K + \min_{c \in \mathcal{C}} \mathbf{e}_D(c) \right) = 1$$

# Main Result

- There exists
  - input domain  $\mathcal{X}$
  - prior  $P$ , non-zero on a countable set of classifiers  $\mathcal{C}$
  - ‘true’ distribution  $D$
  - a constant  $K > 0$

such that the Bayesian learning algorithm  $\text{Bayes}(S, P)$  is **asymptotically  $K$ -suboptimal**:

$$\lim_{m \rightarrow \infty} \Pr_{S \sim D^m} \left( \mathbf{e}_D(\text{Bayes}(S, P)) > K + \min_{c \in \mathcal{C}} \mathbf{e}_D(c) \right) = 1$$

- Same holds for MDL learning algorithm

# Remainder of Talk

1. How is “Bayes learning algorithm” defined?
2. What is scenario?
  - how do  $\mathcal{X}, \mathcal{C}$ , ‘true’ distr.  $D$  and prior  $P$  look like?
3. How dramatic is result?
  - How large is  $K$ ?
  - How strange are choices for  $\mathcal{X}, \mathcal{C}, D, P$  ?
4. Why is result (un-) surprising?
  - is consistency too much to ask for?
  - can it be reconciled with Bayesian *consistency* results?
5. What about MDL?

# Bayesian Learning

- Let
  - $\mathcal{Z}$  be the set of possible outcomes (e.g.  $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$  )
  - $S = (z_1, \dots, z_m) \in \mathcal{Z}^m$  be the observed sample
  - $\mathcal{P}$  be a family of distributions on  $\mathcal{Z}^m$
  - $\pi$  be a distribution on the set of distribution  $\mathcal{P}$ .  $\pi$  is called the **prior** distribution
- Define a distribution  $P_{\text{Bayes}}$  on  $\mathcal{Z}^m \times \mathcal{P}$  by setting, for  $\theta \in \mathcal{P}$ :

$$P_{\text{Bayes}}(z_1, \dots, z_m \mid \theta) := P_{\theta}(z_1, \dots, z_m)$$

$$P_{\text{Bayes}}(\theta) := \pi(\theta)$$

# Bayesian Learning - II

- $P_{\text{Bayes}}$  defined by  $P_{\text{Bayes}}(\theta) := \pi(\theta)$   
 $P_{\text{Bayes}}(z_1, \dots, z_m | \theta) := P_\theta(z_1, \dots, z_m)$
- Given sample data  $S = (z_1, \dots, z_m)$ , calculate **posterior distribution**  $P(\theta | S)$  using **Bayes' rule** as

$$P(\theta | S) = \frac{P(S | \theta)P(\theta)}{P(S)} = \frac{P(S | \theta)P(\theta)}{\int P(S | \theta)P(\theta)d\theta}$$

# Bayesian Learning - II

- $P_{\text{Bayes}}$  defined by  $P_{\text{Bayes}}(\theta) := \pi(\theta)$   
 $P_{\text{Bayes}}(z_1, \dots, z_m | \theta) := P_\theta(z_1, \dots, z_m)$

- Given sample data  $S = (z_1, \dots, z_m)$ , calculate **posterior distribution**  $P(\theta | S)$  using **Bayes' rule** as

$$P(\theta | S) = \frac{P(S | \theta)P(\theta)}{P(S)} = \frac{P(S | \theta)P(\theta)}{\int P(S | \theta)P(\theta)d\theta}$$

- Given enough data, the posterior typically concentrates on a small subset  $\mathcal{P}' \subset \mathcal{P}$ 
  - this  $\mathcal{P}'$  is the set of distributions ‘learned’ by Bayes

# Bayesian Learning of Classifiers

- Problem: Bayesian inference defined for models  $\mathcal{P}$  that are **sets of probability distributions**
- In our scenario, models are **sets of classifiers**  $\mathcal{C}$ , i.e. functions  $c : \mathcal{X} \rightarrow \mathbb{R}$
- How can we find a posterior over classifiers using Bayes rule?
- Standard answer: convert each  $c \in \mathcal{C}$  to a **corresponding distribution**  $P(\cdot | c)$  and apply Bayes to the set  $\mathcal{P}$  of distributions thus obtained

# classifiers $\rightarrow$ probability distrs.

- Standard conversion method from  $\mathcal{C}$  to  $\mathcal{P}$ :  
**logistic (sigmoid) transformation**

- For each  $c \in \mathcal{C}$  and  $\beta \in \mathbb{R}$ , set

$$P_{\text{Bayes}}(Y = 1 \mid X = x, (c, \beta)) := \frac{e^{\beta c(x)}}{1 + e^{\beta c(x)}}$$

$$P_{\text{Bayes}}(y_1, \dots, y_m \mid x_1, \dots, x_m, (c, \beta)) := \prod_{i=1}^m P_{\text{Bayes}}(y_i \mid x_i, (c, \beta))$$

- Define priors  $\pi$  on  $\mathcal{C}$  and  $\pi'$  on  $\mathbb{R}$  and set

$$P_{\text{Bayes}}((c, \beta)) := \pi(c)\pi'(\beta)$$

# classifiers $\longrightarrow$ probability distrs.

- We transformed  $\mathcal{C}$  into corresponding (conditional) probabilistic model  $\mathcal{P}$ , and defined a prior on  $\mathcal{P}$ 
  - Note: model  $\mathcal{P}$  has 1 extra parameter  $\beta \in \mathbb{R}$
- All ingredients for Bayesian learning are now present: Given sample  $S = (X_1, Y_1), \dots, (X_m, Y_m)$  use Bayes' rule to get posterior over **(classifier, confidence)-pairs**  $(c, \beta)$  :

$$P_{\text{Bayes}}(c, \beta | S) = \frac{P_{\text{Bayes}}(y^m | x^m, (c, \beta)) P_{\text{Bayes}}(c, \beta)}{P_{\text{Bayes}}(y^m | x^m)}$$

# Logistic transformation - intuition

- Consider 'hard' classifiers  $c : \mathcal{X} \rightarrow \{-1, 1\}$
- For each  $(c, \beta)$ ,

$$\log P(y^m | x^m, (c, \beta)) = \beta m \hat{e}(c) + m \ln(1 + e^{-\beta})$$

- Here

$$\hat{e}(c) = 0.5 \frac{1}{m} \sum_{i=1}^m |y_i - c(x_i)|$$

is **empirical error** that  $c$  makes on data,

and  $m\hat{e}(c)$  is **number of mistakes**  $c$  makes on data

# Logistic transformation - intuition

$$\log P(y^m | x^m, (c, \beta)) = \beta m \hat{e}(c) + m \ln(1 + e^{-\beta})$$

- where  $m\hat{e}(c)$  is number of mistakes  $c$  makes on data
- For fixed  $\beta > 0$ 
  - log-likelihood is linear function of number of mistakes  $c$  makes on data
  - maximized for  $c$  that is optimal for observed data
- For fixed  $c$ ,
  - log-likelihood maximized for  $\hat{\beta} := \ln \hat{e}(c) - \ln(1 - \hat{e}(c))$
  - $\hat{\beta}$  encodes estimate of quality of  $c$
  - large beta indicates  $c$  made few mistakes on training data

# Logistic transformation - intuition

- The distribution  $P(Y|X, (\hat{c}, \hat{\beta})) \in \mathcal{P}$  that maximizes the likelihood of  $S$  is such that
  - $\hat{c} \in \mathcal{C}$  minimizes number of mistakes on  $S$
  - $\hat{\beta}$  encodes how well  $\hat{c}$  performs on  $S$

A classifier  $c$  achieves small error on sample  $S$  iff for some  $\beta$  the corresponding distribution  $P(Y|X, (c, \beta))$  assigns high probability to  $S$ .

# Logistic transformation - intuition

- In case of real-valued classifiers, other intuitions can be given
- In Bayesian practice, logistic transformation is a standard tool, nowadays performed without giving any motivation or explanation
- Comparable to turning set of predictors with squared error into conditional distributions with normally distributed noise

## 2 Bayesian learning algorithms

- Posterior distribution still needs to be turned into actual learning/prediction algorithm.
- Two standard ways: given  $S = (x_1, y_1), \dots, (x_m, y_m)$

### 1. Bayesian MAP (Maximum A Posteriori):

pick a single  $c \in \mathcal{C}$  that has maximum posterior probability and use it to classify new input value  $x_{m+1}$

### 2. 'Full' Bayesian classifier (should work better!):

$$P_{\text{Bayes}}(Y_{m+1} = 1 \mid X_{m+1} = x, S) =$$

$$\int_{c \in \mathcal{C}; \theta \in \mathbb{R}} P(Y = 1 \mid X_{m+1} = x, (c, \theta)) P_{\text{Bayes}}(c, \theta \mid S) dc d\theta$$

Predict 1 iff  $P_{\text{Bayes}}(Y_{m+1} = 1 \mid X_{m+1} = x, S) > 0.5$

# Main Result

Grünwald & Langford, COLT 2004

- There exists
  - input domain  $\mathcal{X}$
  - prior  $P$  on a countable set of classifiers  $\mathcal{C} : \mathcal{X} \rightarrow \{-1, 1\}$
  - ‘true’ distribution  $D$
  - a constant  $K > 0$

such that the Bayesian learning algorithm  $\text{Bayes}(S, P)$  is **asymptotically K-suboptimal**:

$$\lim_{m \rightarrow \infty} \Pr_{S \sim D^m} \left( \mathbf{e}_D(\text{Bayes}(S, P)) > K + \min_{c \in \mathcal{C}} \mathbf{e}_D(c) \right) = 1$$



holds both for **full Bayes** and for **Bayes MAP**

# Generalization Error

- Generalization error defined as

$$\mathbf{e}_D(h) :=$$

$$\mathbf{Pr}_{(X,Y) \sim D}(Y \neq h(X)) = \frac{1}{2} \mathbf{E}_{(X,Y) \sim D} |Y - h(X)|.$$

# Issues/Remainder of Talk

1. How is “Bayes learning algorithm” defined?
2. What is scenario?
  - how do  $\mathcal{X}, \mathcal{C}$ , ‘true’ distr.  $D$  and prior  $P$  look like?
3. How dramatic is result?
  - How large is  $K$ ?
  - How strange are choices for  $\mathcal{X}, \mathcal{C}, D, P$  ?
4. Why is result (un-) surprising?
  - is consistency too much to ask for?
  - can it be reconciled with Bayesian *consistency* results?
5. What about MDL?

# Scenario

- Definition of  $Y$ ,  $X$  and  $\mathcal{C}$ :

$$Y \in \{-1, 1\}$$

$$X \equiv (X_0, X_1, X_2, \dots) \quad \text{for all } j \geq 0: X_j \in \{-1, 1\}$$

$$\mathcal{C} = (c_0, c_1, c_2, \dots)$$

$$\text{For all } j \geq 0: c_j(X) := x_j$$

- Definition of prior:

- for some small  $\alpha > 0$ , for all large  $n$ ,

$$P_{\text{Bayes}}(c_n) > \frac{1}{n^{1+\alpha}}$$

- $P_{\text{Bayes}}(\beta)$  can be just about any smooth prior

# Scenario - II

- Definition of 'true' distribution D:
2. Toss fair coin to determine value of  $Y$ .
  3. Toss coin  $Z$  with bias  $\Pr(Z = 1) = 0.6$
  4. If  $Z = 0$  (**easy** example) then for all  $j \geq 0$ , set  
 $X_j := Y$
  6. If  $Z = 1$  (**hard** example) then set  
 $X_0 := Y$  with probability  $\frac{2}{3}$ ;  $X_0 := -Y$  otherwise  
and for all  $j > 0$ , independently set  
 $X_j := Y$  with probability 0.5;  $X_j := -Y$  otherwise

# Result:

- All features  $X_j$  are informative of  $Y$ , but  $X_0$  is more informative than all the others, so  $c_0$  is best classifier:

$$\mathbf{e}_D(c_0) = 0.2 \quad \text{while for all } j > 0, \mathbf{e}_D(c_j) = 0.3$$

- Nevertheless, with 'true' D- probability 1, as  $m \rightarrow \infty$

$$\arg \max_j P(c_j | S) \rightarrow \infty$$

$$\frac{P(c_0 | S)}{\max_j P(c_j | S)} < e^{-\text{constant} \cdot (\sqrt{m})}$$

# Idea of proof

- For all **fixed**  $n$ , with probability 1, as  $m \rightarrow \infty$ ,

$$P_{\text{Bayes}}(c_0 \mid S, C \in \{c_0, \dots, c_n\}) \rightarrow 1$$

- However, since

1. all classifiers err **independently**,
2. Prior of  $c_n$  decreases only slowly with  $n$ , ...

...for each  $m$  there will be some classifier  $c_n$  that has 0 error (mistakes) on sample, and has 'relatively large' prior  $P_{\text{Bayes}}(c_n)$

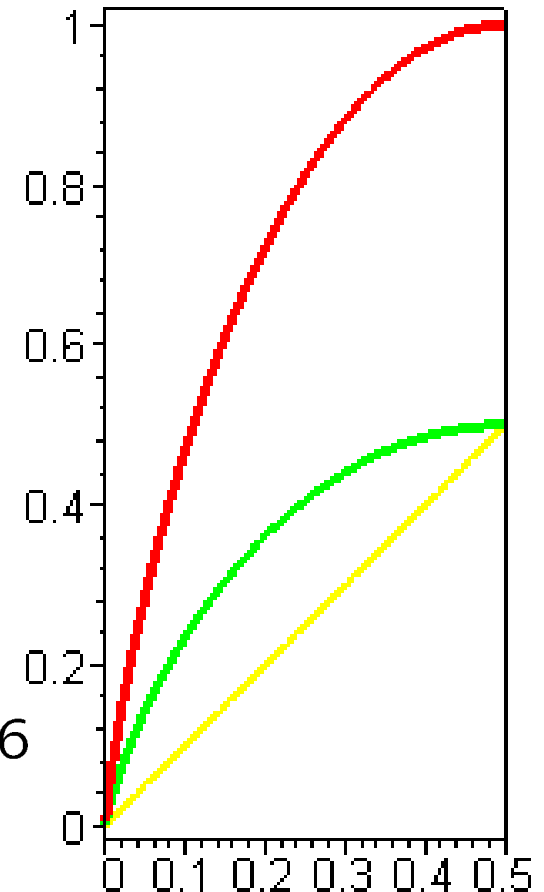
- This classifier has exponentially larger posterior than  $c_0$
- UPSHOT: Bayes avoids overfitting, **but not enough!**

# Issues/Remainder of Talk

1. How is “Bayes learning algorithm” defined?
2. What is scenario?
  - how do  $\mathcal{X}, \mathcal{C}$ , ‘true’ distr.  $D$  and prior  $P$  look like?
3. How dramatic is result?
  - How large is  $K$ ?
  - How strange are choices for  $\mathcal{X}, \mathcal{C}, D, P$  ?
4. Why is result (un-) surprising?
  - is consistency too much to ask for?
  - can it be reconciled with Bayesian *consistency* results?
5. What about MDL?

# How wrong can Bayes go?

- X-axis:  $e_D(c_0)$
- **—** = maximum  $e_D(\text{Bayes}(S, P))$   
that we can prove to be achieved by  
appropriate settings of data generating  
procedure:  
 $\alpha \downarrow 0$  ;  $P(\text{hard example}) = \text{large}$
- **—** = general upper bound on  
 $e_D(\text{Bayes}(S, P))$
- Maximum provable difference  $K \approx 0.16$   
, achieved at  $e_D(c_0) = 0.2$



# How 'natural' is scenario?

- Basic scenario is quite unnatural
- We chose it because we could prove something about it! Two possibilities:
  1. similar result holds in more 'natural' settings but this is harder to prove
  2. a 'really strange' scenario is needed for results of our type
- Not clear what is the case!

# How 'natural' is scenario?

- Priors are natural!
- Rissanen's 'universal prior for the integers',

$$-\log P(c_j) \approx \log j + 2 \log \log j$$

is often used in practice

- All of the green line can be achieved with Rissanen's prior

# Issues/Remainder of Talk

1. How is “Bayes learning algorithm” defined?
2. What is scenario?
  - how do  $\mathcal{X}$ ,  $\mathcal{H}$ , ‘true’ distr.  $D$  and prior  $P$  look like?
3. How dramatic is result?
  - How large is  $K$ ?
  - How strange are choices for  $\mathcal{X}$ ,  $\mathcal{H}$ ,  $D$ ,  $P$  ?
4. Why is result (un-) surprising?
  - is consistency too much to ask for?
  - can it be reconciled with Bayesian *consistency* results?
5. What about MDL?

# Is consistency **relevant**?

- “Among all ‘optimality properties’ of statistical procedures, consistency may be the one whose relevance is the least disputed”

(Kleijn and van der Vaart 2004, others)

# Is consistency **relevant**?

- Methods for avoiding overfitting proposed in statistical and computational Learning theory literature *are* consistent (asymptotically optimal)
  - **Vapnik**'s methods (based on VC-dimension etc.)
  - McAllester's **PAC-Bayes** methods
- These methods invariably punish 'complex' (low prior) classifiers much more than ordinary Bayes – in the simplest version of PAC-Bayes,

$$P_{\text{PAC-Bayes}}(c_j) \approx \left( P_{\text{Bayes}}(c_j) \right)^{\sqrt{m}}$$

# Bayesian **Consistency** Results

- Celebrated results due to Doob (1949), Blackwell and Dubins (1962) and others:

Bayesian inference is consistent under almost no conditions on prior  $P$ , or set of distributions  $\mathcal{P}$ , in sense that

Posterior predictive distribution  $\longrightarrow$  'true' distribution

- In particular, this holds if  $\mathcal{P}$  is complex ('infinite dimensional'). For example,  $\mathcal{P}$  can be
  - All Markov chains of each order ; or
  - All Gaussian mixtures with arbitrary number of components

# Bayesian **Consistency** Results

- For our situation, Doob (1949) implies:

Suppose  $\mathcal{P}$  contains 'true' conditional distribution  $\Pr_D(Y|X)$ . Then with D-probability 1, the Bayesian posterior predictive distribution defined by

$$P_{\text{Bayes}}(Y_{m+1} = 1 \mid X_{m+1} = x, S) = \int_{c \in \mathcal{C}; \theta \in \mathbb{R}} P(Y = 1 \mid X_{m+1} = x, (c, \theta)) P_{\text{Bayes}}(c, \theta \mid S) dc d\theta$$

weakly converges to  $\Pr_D(Y|X)$ .

# Bayesian Consistency Results

- If  $P_{\text{Bayes}}(Y_{m+1} | X_{m+1}, S) \rightarrow \mathbf{Pr}_D(Y|X)$   
...then we must also have

$$\mathbf{e}_D(\text{Bayes}(S, P)) \rightarrow \min_{\text{all classifiers!}} \mathbf{e}_D(c)$$

- Our result says that this does not happen in our scenario. Hence **the  $\mathcal{P}$  we constructed must be misspecified:**

$$\mathbf{Pr}_D(Y|X) \notin \{P(Y|X, (c, \beta) | c \in \mathcal{C}, \beta \in \mathbb{R}\}$$

# Bayesian consistency under misspecification

- Suppose we use Bayesian inference based on ‘model’  $\mathcal{P}$
- If  $\Pr_D(Y|X) \notin \mathcal{P}$ , then under ‘mild’ generality conditions, Bayes still converges to distribution  $\tilde{P}(Y|X) \in \mathcal{P}$  that is closest to  $\Pr_D(Y|X)$  in KL-divergence (relative entropy), defined as

$$\text{KL}(\Pr_D(Y|X) \| P(Y|X, (c, \beta))) = E_{(X,Y) \sim D} \left[ \log \frac{\Pr_D(Y|X)}{P(Y|X, (c, \beta))} \right]$$

# Bayesian consistency under misspecification

- Suppose we use Bayesian inference based on ‘model’

$\mathcal{P}$

- If  $\Pr_D(Y|X) \notin \mathcal{P}$ , then under ‘mild’ generality conditions, Bayes still converges to distribution  $\tilde{P}(Y|X) \in \mathcal{P}$  that is closest to  $\Pr_D(Y|X)$  in KL-divergence.

- By the logistic transformation, for all  $c$ ,

$$\min_{\beta} \mathbf{KL}(\Pr_D(Y|X) \| P(Y|X, (c, \beta))) = -\mathbf{e}_D(c) \log \mathbf{e}_D(c) - (1 - \mathbf{e}_D(c)) \log(1 - \mathbf{e}_D(c)) + \text{const.}$$

which is increasing in  $\mathbf{e}_D(c)$

# Bayesian consistency under misspecification

- In our scenario, Bayesian posterior does not converge to distribution with smallest classification generalization error, so it also does not converge to distribution that closest to 'true'  $D$  in KL-divergence
- Apparently, 'mild' generality conditions for 'Bayesian consistency under misspecification' are violated!
- Conditions for 'consistency under misspecification' are much stronger than conditions for 'consistency if true distribution has positive prior mass.'

# Conclusion

- Bayesian may argue that the Bayesian machinery was never intended for misspecified models
  - After all, the ‘prior’ on  $\mathcal{P}' \subset \mathcal{P}$  indicates your subjective degree of belief that  $\mathcal{P}'$  contains true state of nature;
  - if you know a priori that  $\mathcal{P}'$  does not contain true state of nature, you should assign it prior 0 !
- Yet, computational resources and human imagination being limited, **in practice Bayesian inference is applied to misspecified models all the time.**
- Our result says that in this case, Bayes may overfit even in the limit for an infinite amount of data

**Thank you for your attention!**

# MDL and classificaton

- There is no unique definition of ‘the’ MDL Principle for classification
- Yet there is a certain standard approach that has been employed by most authors:
  - Quinlan and Rivest (1989),
  - Rissanen & Wax (1989),
  - Kearns et al. (1997) ;
  - several others...

# Two-part code MDL

- We use the **oldest, crudest** version of MDL (two-part code MDL, Rissanen '78)
- Problematic aspects of MDL for classification are **not** solved by using modern versions of MDL such as normalized maximum likelihood
- Using two-part code allows us to keep our story as simple as possible

# Two-Part Code MDL

- Two-part code MDL:

Let  $\mathcal{C}$  be a set of hypotheses. Given data sample  $S$ , pick the  $c \in \mathcal{C}$  that minimizes the sum of

- the description length of the hypothesis  $c$
- the description length of the data  $S$  when encoded ‘with the help of the hypothesis  $c$ ’

# Two-Part Code MDL

- Pick  $c \in \mathcal{C}$  minimizing

$$DL(c) + DL(y_1, \dots, y_n \mid c, x_1, \dots, x_n)$$

# Two-Part Code MDL

- Pick  $c \in \mathcal{C}$  minimizing

$$DL(c) + DL(y_1, \dots, y_n \mid c, x_1, \dots, x_n)$$

Encoding of  $x_1, \dots, x_n$  takes  $DL(x_1, \dots, x_n)$  bits; this term does not involve  $c$ . Therefore it plays no role in minimization and can be dropped!

# Two-Part Code MDL

- Pick  $c \in \mathcal{C}$  minimizing

$$DL(c) + DL(y_1, \dots, y_n \mid c, x_1, \dots, x_n)$$



Any function on  $\mathcal{C}$  satisfying Kraft inequality

# Coding Hypotheses

- $DL(c) = -\log W(c)$  ,  $W$  can be thought of as 'prior' ; many reasonable possibilities
- example code for intervals domain:
- encode  $c \in \mathcal{C}$  in three steps:
  1. Encode number of switches  $k$
  2. Encode 'granularity'  $d$
  3. Code location of  $k$  switches within  $\{0, \frac{1}{d}, \frac{2}{d}, \dots, \frac{d-1}{d}\}$

# Two-Part Code MDL

- Pick  $c \in \mathcal{C}$  minimizing

$$DL(c) + DL(y_1, \dots, y_n \mid c, x_1, \dots, x_n)$$



~~Code~~

Code by ending  
a. number of mistakes  
b. location (index) of mistakes

# Coding Data: $DL(y^n | x^n, c)$

- Define:
- **mistake count** :  $M_c$ 
  - number of mistakes  $h$  makes on  $D$

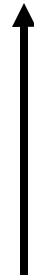
- Formally, 
$$M_c = 0.5 \sum_{i=1}^m |y_i - c(x_i)|$$

# Standard approach to coding data

$$\begin{aligned} \text{DL}^*(y_1, \dots, y_n \mid c, x_1, \dots, x_n) &= \\ &= \log(n + 1) + \log \binom{n}{M_c} \end{aligned}$$




nr of bits needed to  
encode total nr of  
mistakes



nr of bits needed to  
encode location of  
mistakes

# 2p-code length intervals domain

$$\min_{c \in \mathcal{C}} \left\{ \text{DL}(y^n | x^n, c) + \text{DL}(c) \right\} =$$
$$\min_{c_{k,d} \in \mathcal{C}} \left\{ \log \binom{n}{M_c} + \log gk + \log gd + \log \binom{d}{k} \right\}$$



error term complexity term

- familiar trade-off between error and complexity
- we can and did leave out  $\log(n + 1)$  term

# The Upshot

- The version of MDL we have just described can also be asymptotically inconsistent
- The same discrepancies between  $e_D(\text{MDL}(S))$  and  $\min_{c \in \mathcal{C}} e_D(c)$  occur as in the Bayesian algorithm,
  - in the same scenario
  - for the same priors

**Thank you for your attention!**