MDL and Classification

Peter Grünwald
CWI and EURANDOM
www.grunwald.nl

Part IV of a series of Lectures on 'modern MDL' ;
Extended and Revised September 24th, 2003

Classification: Overview

1. Introduction
2. MDL for classification, basic approach
3. The Promise
   - Basic approach has some great properties!
4. The Problem
   - Basic approach shows problematic behaviour
5. Conclusions

Introduction

- MDL mostly developed and studied for probability models
- Yet often applied to models/model classes that are not (directly) interpretable as probability distributions
- Here we apply it to models that are families of classifiers
  - decision trees
  - support vector machines
  - neural networks…

Introduction - II

- 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…

Introduction - III

- Standard approach has pleasant but also unpleasant properties:
  - strange experimental results (Kearns et al. 1997 (?))
  - can be inconsistent! (Grünwald & Langford, 2003)
    - Even with infinite data, MDL does not identify the classifier with the smallest 'generalization error' (probability of making a wrong prediction) – it asymptotically overfits!
- Several adjustments exist
  - Barron (1991), Yamanishi (1998), McAllester's PAC-Bayes (1999)
  - these are provably consistent
  - but loose some of the pleasant properties of standard approach

Classification: Overview

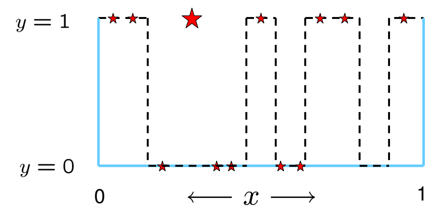
1. Introduction
2. MDL for classification, basic approach
3. The Promise
   - Basic approach has some great properties!
4. The Problem
   - Basic approach shows problematic behaviour
5. Conclusions

Classification

- Given:
  - Feature space $\mathcal{X}$
  - Label space $\mathcal{Y} = \{0, 1\}$
  - data $D = ((x_1, y_1), \ldots, (x_n, y_n))$
  - countable set $\mathcal{H}$ of hypotheses (classifiers) $h : \mathcal{X} \rightarrow \mathcal{Y}$
- Goal: find a $h \in \mathcal{H}$ that makes few mistakes on future data from the same source
  - We say 'h has small generalization error'
  - if data are noisy, then it is not a good idea to adopt the h that minimizes nr of mistakes on the given data
  - leads to over-fitting

Example: intervals (toy) domain

Kearns et al., 1995



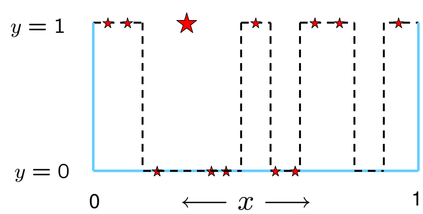
$\mathcal{X} = [0, 1]$

$\mathcal{H}_k$: set of functions $h : \mathcal{X} \rightarrow \{0, 1\}$ that switch value k times

$\mathcal{H} = \bigcup_{k=1,2,\ldots} \mathcal{H}_k$

Example: intervals domain

Kearns et al., 1995



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

$\mathcal{H}_k$: set of functions $h : \mathcal{X} \rightarrow \{0, 1\}$ that switch value k times

$\mathcal{H} = \bigcup_{k=1,2,\ldots} \mathcal{H}_k$

the h in picture is in $\mathcal{H}_6$ and makes 1 *classification error* on data D

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
  - Grünwald & Langford, 2003
- Using two-part code allows us to keep our story as simple as possible

Two-Part Code MDL

Two-part code MDL:

- Let $\mathcal{H}$ be a set of hypotheses. Given data D pick the $h \in \mathcal{H}$ that minimizes the sum of
  - the description length of the hypothesis h
  - the description length of the data D when encoded 'with the help of the hypothesis h'

Two-Part Code MDL

Pick $h \in \mathcal{H}$ minimizing

$$\mathrm{DL}(h) + \mathrm{DL}(y_1, \ldots, y_n \mid h, x_1, \ldots, x_n)$$

Two-Part Code MDL

Pick $h \in \mathcal{H}$ minimizing

$$\text{DL}(h) + \text{DL}(y_1, \ldots, y_n \mid h, x_1, \ldots, x_n)$$

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

Two-Part Code MDL

Pick $h \in \mathcal{H}$ minimizing

$$\text{DL}(h) + \text{DL}(y_1, \ldots, y_n \mid h, x_1, \ldots, x_n)$$

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

Coding Hypotheses

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

Coding Data

Pick $h \in \mathcal{H}$ minimizing

$$\text{DL}(h) + \text{DL}(y_1, \ldots, y_n \mid h, x_1, \ldots, x_n)$$

Code $y_1, \ldots, y_n$ by coding
a. number of mistakes
b. location (index) of mistakes

Coding Data: $\text{DL}(y^n \mid x^n, h)$

- Define:
  – mistake count M_h
  number of mistakes h makes on D

  – 0/1-loss: for $y, \hat{y} \in \{0, 1\}$:
  $L_{01}(y, \hat{y}) := |y - \hat{y}|$
- Formally, $M_h := \sum_{i=1}^{n} L_{01}(y_i, h(x_i))$

Standard approach to coding data

$$\text{DL}^*(y_1, \ldots, y_n \mid h, x_1, \ldots, x_n) =$$
$$= \log(n+1) + \log \binom{n}{M_h}$$

nr of bits needed to encode total nr of mistakes

nr of bits needed to encode location of mistakes

2p-code length intervals domain

$$\min_{h \in \mathcal{H}} \left\{ \text{DL}(y^n \mid x^n, h) + \text{DL}(h) \right\} =$$

$$\min_{h_{k,d} \in \mathcal{H}} \left\{ \log \binom{n}{M_h} + \log k + \log d + \log \binom{d}{k} \right\}$$

error term complexity term

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

MDL Version C0

- We call the coding scheme for 'coding data with the help of hypothesis' MDL Version C0.
- (slight variations of) MDL C0 used by
  - Quinlan and Rivest (1989),
  - Rissanen & Wax (1989),
  - Kearns et al. (1997) ;
  - even Wallace & Boulton (1968)
- But is it the 'right' way to do things?

Potential Problems:

1. Many different coding schemes of data given hypothesis $\text{DL}(y^n | h, x^n)$ possible
   - Comparison strongly indicates that MDL C0 is basically the 'right' coding scheme.
2. Theoretical results on MDL C0
   - (in sharp constrast to probabilistic MDL), analysis strongly indicates that nevertheless something's wrong with MDL C0

Classification: Overview

1. Introduction
2. MDL for classification, basic approach
3. The Promise
   - Basic approach has some great properties!
4. The Problem
   - Basic approach shows problematic behaviour
5. Conclusions

1. Alternative coding schemes

- Two other coding schemes have been proposed in the literature.
  - seemingly very different, they both lead to same hypothesis selection criterion as MDL C0
  - shows that MDL C0 is special case of general procedure, applicable to arbitrary loss functions
- Evidence that what we're doing is o.k.!

MDL C1: *entropification*

Rissanen 1989, implicit in Vovk 1990
Meir and Merhav 1995, Yamanishi 1998
Grünwald 1998

- Suppose we have a code such that for all h, all (x^n, y^n), the code length is an increasing affine function of the loss:

$$\text{DL}(y^n \mid x^n, h) = \beta \sum_{i=1}^{n} L_{01}(y_i; h(x_i)) + \alpha$$
$$= \beta M_h + \alpha$$

- Here $\beta > 0$; α may depend on n, but not on h

MDL C1: *entropification*

Rissanen 1989, Meir and Merhav 1995, Yamanishi 1998, Grünwald 1998, implicit in Vovk 1990 and others

- Suppose we have a code such that for all h, all (x^n, y^n), the code length is an increasing affine function of the loss:

$$\mathrm{DL}(y^n \mid x^n, h) = \beta \sum_{i=1}^{n} L_{01}(y_i; h(x_i)) + \alpha$$
$$= \beta M_h + \alpha$$

- then 'error term' in $\mathrm{DL}(y^n|x^n, h) + \mathrm{DL}(h)$ expresses exactly the error function we are interested in!

entropification

- We can construct a code satisfying $\mathrm{DL}(y^n \mid x^n, h) = \beta \sum_{i=1}^{n} L_{01}(y_i; h(x_i)) + \alpha$

  by first constructing a conditional probability distribution:

$$P(y|x, h, \beta) := \frac{1}{Z(\beta)} e^{-\beta L_{01}(y;h(x))}$$

$$Z(\beta) := \sum_{y \in \{0,1\}} e^{-\beta L_{01}(y;h(x))}$$

Note: $Z(\beta)$ does not depend on h or X !

entropification

- We can construct a code satisfying $\mathrm{DL}(y^n \mid x^n, h) = \beta \sum_{i=1}^{n} L_{01}(y_i; h(x_i)) + \alpha$

  by first constructing a conditional probability distribution:

$$P(y|x, h, \beta) := \frac{1}{Z(\beta)} e^{-\beta L_{01}(y;h(x))}$$

$$Z(\beta) := \sum_{y \in \{0,1\}} e^{-\beta L_{01}(y;h(x))}$$

$$P(y^n|x^n, h, \beta) := \prod_{i=1}^{n} \frac{1}{Z(\beta)} e^{-\beta L_{01}(y_i;h(x_i))}$$

- then

$$-\ln P(y^n|x^n, h, \beta) = \beta \sum_{i=1}^{n} L_{01}(y_i; h(x_i)) + n \ln Z(\beta)$$

entropification

- For each h, β we constructed a corresponding conditional probability distribution satisfying , for all $D = (x^n, y^n)$,

$$-\ln P(y^n|x^n, h, \beta) = \beta \sum_{i=1}^{n} L_{01}(y_i; h(x_i)) + n \ln Z(\beta)$$

- By Kraft inequality, there must also exist a (conditional) code defined on data sequences of length n , satisfying

$$\mathrm{DL}(y^n|x^n, h, \beta) = \beta \sum_{i=1}^{n} L_{01}(y_i; h(x_i)) + n \ln Z(\beta)$$

- This is the code we'll use!

entropification

- For each h, β we constructed a corresponding conditional probability distribution satisfying , for all $D = (x^n, y^n)$,

$$-\ln P(y^n|x^n, h, \beta) = \beta \sum_{i=1}^{n} L_{01}(y_i; h(x_i)) + n \ln Z(\beta)$$

- By Kraft inequality, there must also exist a (conditional) code defined on data sequences of length n , satisfying

$$\mathrm{DL}(y^n|x^n, h, \beta) = \beta \sum_{i=1}^{n} L_{01}(y_i; h(x_i)) + n \ln Z(\beta)$$

  - Code length measured in nats
  - Important: *no* claim that $P(\cdot \mid \cdot, h, \beta)$ generates the data; purely artificial construction to make sure that code length of data given h = linear function of loss h makes on data

entropification

- MDL now becomes: select $h \in \mathcal{H}$ minimizing

$$\beta \sum_{i=1}^{n} L_{01}(y_i; h(x_i)) + n \ln Z(\beta) + \mathrm{DL}(h)$$

- Problem: how to choose β ?
  - different β lead to different choices of h
  - β measures how strongly the 0/1-error should be weighted compared to the 'complexity' of h
    - β viewed as learning rate, inverse 'temperature'

entropification

- MDL now becomes: select $h \in \mathcal{H}$ minimizing
  $$\beta \sum_{i=1}^{n} L_{01}(y_i; h(x_i)) + n \ln Z(\beta) + \mathrm{DL}(h)$$
- Problem: how to choose β ?
  - different β lead to different choices of h
  - β measures how strongly the 0/1-error should be weighted compared to the \`complexity' of h
- Intuitive Solution
  - **learn** not just h , but also β from the data

entropification

- MDL now becomes: select $h \in \mathcal{H}$ achieving
  $$\min_{h \in \mathcal{H}, \beta \in [0,\infty]} \left\{ \beta \sum_{i=1}^{n} L_{01}(y_i; h(x_i)) + n \ln Z(\beta) + \mathrm{DL}(h) + \mathrm{DL}(\beta) \right\}$$
- We'll see in a minute that this does (almost) exactly the same as MDL C0 …

Don't worry about $\mathrm{DL}(\beta)$ for now!

entropification

- MDL now becomes: select $h \in \mathcal{H}$ achieving
  $$\min_{h \in \mathcal{H}, \beta \in [0,\infty]} \left\{ \beta \sum_{i=1}^{n} L_{01}(y_i; h(x_i)) + n \ln Z(\beta) + \mathrm{DL}(h) + \mathrm{DL}(\beta) \right\}$$
- We'll see in a minute that this does (almost) exactly the same as MDL C0 …
- …we do this by giving a *third* coding scheme easily shown to be equivalent with MDL C0 and MDL C1 ('entropification')

MDL C2: Probabilistic coding

- Original two-part code MDL (Rissanen '78) was really designed for probability models:
  - Let $\mathcal{P}$ be a countable set of (conditional) distributions on $\mathcal{Y}$ given $\mathcal{X}$
  - Then *probabilistic* two-part code MDL tells us to select the $P \in \mathcal{P}$ achieving
  $$\min_{P \in \mathcal{P}} - \log P(y^n \mid x^n) + \mathrm{DL}(P)$$

MDL Version C2: Probabilistic coding

- Original two-part code MDL (Rissanen '78) was designed for probability models only:
  - Let $\mathcal{P}$ be a countable set of (conditional) distributions on $\mathcal{Y}$ given $\mathcal{X}$
  - Then *probabilistic* two-part code MDL tells us to select the $P \in \mathcal{P}$ achieving
  $$\min_{P \in \mathcal{P}} - \ln P(y^n \mid x^n) + \mathrm{DL}(P)$$
- We'll recast classification in probabilistic terms

MDL C2

- Define for each $h \in \mathcal{H}$ and 'noise level' $\theta \in [0,1]$ associated Boolean regression model, i.e.
  $$Y_i = h(X_i) \text{ xor } Z_i$$
  where
  $$Z_i \in \{0,1\}, P(Z_i = 1) = \theta, X_i, Y_i, Z_i \text{ i.i.d.}$$
- Let $P(\cdot, \mid \cdot, h, \theta)$ be the associated conditional distribution:
  $$P(y^n \mid x^n, h, \theta) = \theta^{M_h}(1-\theta)^{n-M_h}$$

MDL Version C2

- MDL C2 tells us to pick (h, θ) minimizing

$$-\ln P(y^n|x^n, h, \theta) + \text{DL}(h) + \text{DL}(\theta) =$$
$$-M_h \ln \theta - (n - M_h) \ln(1 - \theta) + \text{DL}(h) + \text{DL}(\theta)$$

MDL C2

- MDL C2 tells us to pick (h, θ) minimizing

$$-\ln P(y^n|x^n, h, \theta) + \text{DL}(h) + \text{DL}(\theta) =$$
$$-M_h \ln \theta - (n - M_h) \ln(1 - \theta) + \text{DL}(h) + \text{DL}(\theta)$$

- MDL C1 tells us to pick (h, β) minimizing

$$\beta \sum_{i=1}^{n} L_{01}(y_i; h(x_i)) + n \ln Z(\beta) + \text{DL}(h) + \text{DL}(\beta) =$$
$$\beta M_h + n \ln(1 + e^{-\beta}) + \text{DL}(h) + \text{DL}(\beta)$$

- substituting $\beta_\theta := \ln \frac{1-\theta}{\theta}$ shows this is the same!

MDL C2 = MDL C1

- Conclusion:
  - MDL C1 and C2 yield exactly the same hypothesis for the same data,even though codes were motivated differently:
    - version 1: code length of data linear function of loss
    - version 2: probabilistic assumption that data generated by some deterministic process + noise
  - Can encode β by encoding corresponding θ, using $\ln(n+1)$ nits

MDL C2 vs MDL C0

- MDL C2 tells us to pick (h, θ) minimizing

$$-\ln P(y^n|x^n, h, \theta) + \text{DL}(h) + \text{DL}(\theta) =$$
$$-M_h \ln \theta - (n - M_h) \ln(1 - \theta) + \text{DL}(h) + \text{DL}(\theta)$$

- $\min_{\theta \in [0,1]} \{-M_h \ln \theta - (n - M_h) \ln(1 - \theta)\}$

is achieved for maximum likelihood $\hat{\theta}$:

$$\hat{\theta} := \frac{M_h}{n} = \frac{1}{n} \sum_{i=1}^{n} L(y_i, h(x_i))$$

so that

$$\min_{\theta} \{-M_h \ln \theta - (n - M_h) \ln(1 - \theta)\} =$$
$$n[-\hat{\theta} \ln \hat{\theta} - (1 - \hat{\theta}) \ln(1 - \hat{\theta})] = n\mathbf{H}(\hat{\theta})$$

where $\mathbf{H}(\theta)$ is the binary entropy of a coin with bias θ

MDL C2 vs MDL C0

- MDL C2 tells us to pick h minimizing

$$-\ln P(y^n|x^n, h, \theta) + \text{DL}(h) + \text{DL}(\theta) =$$
$$n\mathbf{H}(\hat{\theta}) + \text{DL}(h) + \ln(n+1)$$

MDL C2 vs MDL C0

Recall: for MDL Version C0 we had

$$\begin{aligned} & \text{DL}(y_1, \ldots, y_n \mid h, x_1, \ldots, x_n) = \\ = \; & \ln(n+1) + \ln \binom{n}{M_h} = \\ = \; & \ln(n+1) + \mathbf{H}\left(\frac{M_h}{n}\right) - \frac{1}{2} \ln n + O(1) = \\ = \; & \mathbf{H}(\hat{\theta}) + \frac{1}{2} \ln n + O(1) \end{aligned}$$

- standard application of Stirling's approximation

MDL C2 vs MDL C1

- MDL C2 tells us to pick h minimizing

$$-\ln P(y^n|x^n, h, \theta) + \mathrm{DL}(h) + \mathrm{DL}(\theta) =$$
$$n\mathbf{H}(\hat{\theta}) + \mathrm{DL}(h) + \ln(n+1)$$

- MDL C0 tells us to pick h minimizing

$$n\mathbf{H}(\hat{\theta}) + \mathrm{DL}(h) + \frac{1}{2}\ln n \quad [+O(1)]$$

- (almost) the same!

MDL C0 = MDL C1 = MDL C2

- Conclusion: all three versions essentially the same!
- Henceforth take MDL C1 as canonical since
  1. it suggests how to extend the approach to different settings (predictors, loss functions)
  2. useful to learn not just h, but also β

More on β .

- MDL C1 tells us to minimize

$$\beta \sum_{i=1}^{n} L_{01}(y_i; h(x_i)) + n \ln Z(\beta) + \mathrm{DL}(h) + \mathrm{DL}(\beta)$$

- Keeping h fixed and minimizing only over β, min is achieved for $\hat{\beta} = \ln(1-\hat{\theta}) - \ln\hat{\theta}$ with $\hat{\theta} = M_h/n$
  - $\hat{\beta}$ implicitly represents loss that h makes on data
  - Maybe can be used as estimate of h's loss on future data?

- $\hat{\beta}_h < 0$ corresponds to h that makes $> 50\%$ mistakes
  - then $\overline{h}$ is a better predictor than h for given data

Extensions

- Approach can be generalized to (quite) arbitrary symmetric loss fns (Grünwald 98)
  - Example: for the squared error, an analogous story has been known for many years
- Recently, shown that approach can even be generalized to non-symmetric loss functions
  - e.g. $L(1;1) = L(0;0) = 0$; $L(1;0) = 1$; $L(0;1) = 10^6$
  - considerably more complicated

Does it 'work'?

- Would like to show some consistency or rate-of-convergence results, saying that
  'assuming that data are distributed according to some distribution P^*, then with high P^* probability, the hypothesis inferred by MDL C0 converges to the 'best' hypothesis in (closure of) $\mathcal{H}$'

Does it 'work'?

Baby-Theorem (Grünwald 1998, others) Suppose data $(X_1, Y_1), (X_2, Y_2), \ldots,$ are independently and identically distributed according to some distribution P^* on $\mathcal{X} \times \mathcal{Y}$.

Let $\tilde{\theta} := \inf_{h \in \mathcal{H}} E_{P^*}[L_{01}(Y; h(X))] = \inf_{h \in \mathcal{H}} P^*(Y \neq h(X))$.
Let $\tilde{\beta} := \ln(1-\tilde{\theta}) - \ln\tilde{\theta}$.

Let $\mathcal{H}$ be **finite**, let DL be a code length function such that $\mathrm{DL}(h)$ is finite for all $h \in \mathcal{H}$. Let $(\hat{h}_n, \hat{\beta}_n)$ be the hypothesis inferred by *MDL-CS* based on the first n outcomes. Then with P^*-probability 1,

$$E_{P^*}[L(Y; \hat{h}_n(X))] \to \inf_{h \in \mathcal{H}} E_{P^*}[L(Y; h(X))] \text{ as } n \to \infty.$$
$$\hat{\beta}_n \to \tilde{\beta} \text{ as } n \to \infty.$$

Does it 'work'?

Baby-Theorem (Grünwald 1998, others) Suppose data $(X_1, Y_1), (X_2, Y_2), \ldots,$ are independently and identically distributed according to some distribution P^* on $\mathcal{X} \times \mathcal{Y}$.

Let $\tilde{\theta} := \inf_{h \in \mathcal{H}} E_{P^*}[L_{01}(Y; h(X))] = \inf_{h \in \mathcal{H}} P^*(Y \neq h(X))$.
Let $\tilde{\beta} := \ln(1 - \tilde{\theta}) - \ln \tilde{\theta}$.

Let $\mathcal{H}$ be **finite**, let DL be a code length function such that $\mathrm{DL}(h)$ is finite for all $h \in \mathcal{H}$. Let $(\hat{h}_n, \hat{\beta}_n)$ be the hypothesis inferred by *MDL-C0* based on the first n outcomes. Then with P^*-probability 1,

$$E_{P^*}[L(Y; \hat{h}_n(X))] \rightarrow \inf_{h \in \mathcal{H}} E_{P^*}[L(Y; h(X))] \text{ as } n \rightarrow \infty.$$

MDL is asymptotically **optimal**

$$\hat{\beta}_n \rightarrow \tilde{\beta} \text{ as } n \rightarrow \infty.$$

MDL is asymptotically **reliable**

Does it 'work'?

- In words, MDL-C0 is 'consistent':
  - MDL-C0 is capable of finding the 'best' hypothesis, with smallest 'generalization error' (optimality)
  - $\hat{\beta}_n$ can be interpreted as consistent estimator of $P^*(Y \neq \hat{h}_n(X))$, the generalization error of the hypothesis output by MDL-C0 (reliability).

Does it work?

- Baby-theorem can be extended to infinite $\mathcal{H}$ with finite VC-dimension, or to various forms of 'parametric' $\mathcal{H}$
- More generally, theorem holds for any type of $\mathcal{H}$ satisfying uniform law of large numbers
- But these are typically ***not*** the type of $\mathcal{H}$ we want to apply MDL to!
  - Example: intervals domain/decision trees: $\mathcal{H}$ has infinite VC-dimension

Part IV: Overview

1. Introduction
2. MDL for classification, basic approach
3. The Promise
   - Basic approach has some great properties!
4. The Problem
   - Basic approach shows problematic behaviour
5. Conclusions

Problems for MDL-CS

- What about grown-up versions of our baby-theorem for arbitrary countable $\mathcal{H}$ with arbitrary codes $\mathrm{DL}(h)$?
- For probabilistic MDL, general consistency/rate of convergence results exist
  - e.g., Barron and Cover 1991
  - related to Bayesian consistency proofs
- For MDL-C0, no such results exist
- …and in fact, they do not hold!

The Problem

- MDL C1 may be interpreted as applying MDL to a set of countable conditional probability distributions….so it may seem that Barron and Cover's results are still applicable…
- …but they aren't!

The Problem

- Why aren't standard consistency results applicable?
  - These all assume that the 'true' distribution P^* is in (the information closure of) $\mathcal{P}$
  - Our constructed probability distributions implicitly assume that misclassification probability is independent of X :
  - We have, for all $\mathcal{R}_1, \mathcal{R}_2 \subset \mathcal{X}$ with $P^*(X \in \mathcal{R}_i) > 0$

$$P(Y \neq \tilde{h}(X) \mid X \in \mathcal{R}_1, \tilde{h}, \tilde{\beta}) = P(Y \neq \tilde{h}(X) \mid X \in \mathcal{R}_2, \tilde{h}, \tilde{\beta})$$

- Only if this also holds for 'true' distribution, i.e. if

$$P^*(Y \neq \tilde{h}(X) \mid X \in \mathcal{R}_1) = P^*(Y \neq \tilde{h}(X) \mid X \in \mathcal{R}_2)$$

can B&C's result be applied

- But this is a very strong and unrealistic assumption!

The Problem

- In fact, none of the existing proofs of consistency of MDL or Bayesian procedures for countable models (sets of prob. distributions) can be applied without making unreasonable assumptions on P^*
- Very recently, we showed that in fact, two-part code MDL can indeed be **inconsistent!**
  - Grunwald & Langford, 2003 (under submission/revision)
- Problem not just for MDL but also for 'Bayesian classification under misspecification'

The Problem - II

- We strongly suspect that also more sophisticated versions of MDL (based on normalized maximum likelihood, Bayesian marginal likelihood) can be inconsistent
- …but no proof yet.

Adjusting MDL-C0

- Barron (1991) and Yamanishi (1998) consider adjustments of the MDL-complexity penalty that are provably consistent for inference of predictors for a given loss function
  - classification as special case
- PAC-Bayes: McAllester (1998, 1999, 2001) considers adjustments of Bayesian inference for classification that are provably consistent 'under misspecification'
- Freund, Mansour, Shapire (2003) – another pseudo-Bayesian, provably consistent inference method for classification

Previous Solutions

- All these adjustments typically punish complexity of hypothesis much more heavily than ordinary MDL
- Advantage:
  - this ensures that no asymptotic overfitting takes place…
- Disadvantages:
  - no (straightforward) coding interpretation
  - learning 'slow' compared to ordinary MDL…perhaps slower than necessary?

cf Tsybakov 1999

Example: Yamanishi's **MLC**

Yamanishi 1998

- MDL-CS:

$$\min_{\beta \in \mathbb{R}, h \in \mathcal{H}} \beta L_{01}(D; h) + n\psi(\beta) + \mathrm{DL}(h) =$$

$$\min_{h \in \mathcal{H}} \hat{\beta}_h L_{01}(D; h) + n\psi(\hat{\beta}_h) + \mathrm{DL}(h)$$

where $\hat{\beta}_h = \ln(1 - \hat{\theta}_h) - \ln \hat{\theta}_h$ $\boxed{\text{stays away from 0!}}$

- Yamanishi's MLC:

$$\min_{h \in \mathcal{H}} \beta_n L_{01}(D; h) + n\psi(\beta_n) + \mathrm{DL}(h)$$

where $\beta_n = \Theta(\sqrt{\frac{\ln n}{n}})$ $\boxed{\text{goes to 0!}}$

Example: Yamanishi's MLC

Yamanishi 1998

- Yamanishi's MLC:

$$\min_{h \in \mathcal{H}} \beta_n L_{01}(D;h) + n\psi(\beta_n) + \mathrm{DL}(h)$$

$$\beta_n = \Theta\left(\sqrt{\frac{\ln n}{n}}\right)$$

- Equivalently,

$$\min_{h \in \mathcal{H}} L_{01}(D;h) + \frac{1}{\beta_n}\mathrm{DL}(h)$$

- Compare to Barron's (1991) regularization:

$$\min_{h \in \mathcal{H}} L_{01}(D;h) + \lambda\sqrt{n\mathrm{DL}(h)}$$

where λ is some positive constant

Ubiquitous $\sqrt{n}$!

- McAllester's PAC-Bayes also leads to a model selection criterion with $\sqrt{n}$ factor in front of complexity term
  - some important refinements though
- $\sqrt{n}$ also hidden in Freund, Mansour, Shapire's work

Problems

- Approaches that are provably consistent have $\beta_n \to 0$ as n increases. Problems (in my view):
  1. There is no clear coding interpretation any more (following Rissanen, I would like to keep the coding interpretation if at all possible)
  2. β_n cannot be interpreted as an estimator of the loss h will make on future data any more (following intuition, I would like to keep this interpretation if at all possible!)
  3. Complexity penalties may (?) sometimes be larger than necessary (viz Tsybakov's recent work)
     - Smaller penalties may give better rates of convergence for certain classes of 'true' P^*

Classification – Conclusion I

- Two-part code MDL can fail for classification
- More sophisticated versions of MDL/Bayes can fail as well (did not discuss this in detail)
- In practice though, MDL often slightly underfits rather than overfits!
  - Possible reason: code length based on local rather than global optima in error surface (?)

Classification – Conclusion II

- 'raw' MDL suited and designed for probability models
  - typically consistent if well-specified, i.e. if 'true' data-generating distribution in (closure) of model $\mathcal{M}$
  - Consistent under misspecification under certain conditions, e.g. if $\mathcal{M}$ is a convex set of distributions
- MDL turns non-probability models (e.g. classifiers) into codes (probability distributions) first; the resulting model is typically misspecified and, unfortunately not convex…so that we may get inconsistency

Thank you for your attention!