授業 / 2015 /

MDL基準による仮説・モデル選択

Index

機械学習において過学習を防ぐ汎用的な枠組みであるMDL基準を紹介する。

背景

機械学習の一般的な目的は、与えられた訓練データ集合に対し適切なモデル（傾向性や構造）を「学習」することである（更に典型的には、これを利用して未知のデータ集合における予測に役立てる）。この学習の過程では、訓練データへの適合が行き過ぎる、過学習（未知のデータでは期待できないような、訓練データに特異な構造に適合してしまうことで、Overfittingともいう）が問題となる。そこで、モデルの複雑さを制御するいくつかの処方箋が考えられている。

- (k-fold, leave-one-out) Cross-validation など、汎化性能の経験的測定
- L_1 正則化 などの Regularization
- AIC, BIC など統計学由来の情報量基準の援用

どの方法にもそれぞれの理論的背景・適用範囲があるが、ここではBICなどと関係が深く、広い範囲のモデルに適用できるMDL基準を紹介する。（ここでは ad hoc なモデルの表現方法による two-part MDL を紹介する。）

動機

MDLの哲学では以下のように考える。機械学習とは、データセットの規則性を学習することに他ならない。例えば、良い機械学習手法は、次の例から規則性を読み取れるべきである。

> A. 00010001000100010001...
> B. 11011100101110111000...

では、規則性とは何か。データ内の規則性は、データの圧縮に利用できることに着目する。そこで逆に、データの圧縮可能性により規則性を定義する。圧縮とは、元のデータを復元できる、より簡潔な表現を見出すことである。最後に、（同様にデータを説明できるならば）表現は簡潔であればあるほど良い（科学的方法の公理・オッカムの剃刀）。

これらの前提から、過学習を防ぐために、「データのなるべく簡潔な表現」を探すことを目指すのがMDLの発想である。

具体的な手続きとしては、データの圧縮とは、（元のデータを表す）ビット列から（圧縮された表現である）ビット列への変換を指す。ここで大事なのは、圧縮が可逆であるためには、圧縮先の表現の中に圧縮の方法に関する情報が必要であるという事実である。そこで圧縮先のデータは、「圧縮方法の表現＋圧縮されたデータ」という二部分に分けられる。MDL基準では、これを最小化するモデルや仮説が選ばれる。実はこれが、「モデルの簡潔さ＋訓練データへの適合」というトレードオフを表現したものになっていることが分かる。

定式化

圧縮のアナロジーから、MDLでは次の二つの量を定義する。

1. $L_1(P)$: モデルを記述するために必要なビット数
2. $L_2(D|P)$:（モデルの助けを得て）データを圧縮記述するために必要なビット数

ここで、D は訓練データで、P は D を説明するためのモデル（確率分布）である。直観的には、モデルがデータを良く説明できる（モデル P のもとでの D の尤度が高い）ほど $L_2(D|P)$ は小さくなり、モデル P がシンプルなほど $L_1(P)$ は小さくなる。

これらの二項の実際の計算方法は以降で説明される。MDL基準は次のように定義される。

> MDL Principle:
> We should select a model P minimizing the quantity: $L_1(P) + L_2(D|P)$.

これが、MDL 基準が「モデルの簡潔さ＋訓練データへの適合」を同時に考慮するということの数学的表現である。

圧縮と確率

この節では、$L_2(D|P)$ の定義のために確率分布と符号化の関係について準備する。
簡単のため、データは既にビット列で表現されているとする。圧縮方法（以下コードという）を定めることは、ビット列からビット列への（単射の）関数 $C(x)$ を定めることと等価。

コードの例a

0100 0001 -> 100
0100 0011 -> 101
0100 0111 -> 110
0101 0100 -> 111
(other 8 bit) -> 0 + (identical 8 bits)

Prefix Code

コードの連結可能性を保証するために、以下では prefix code であることを要求する。コードがprefixであるとは、$C(x)$ が $C(y)$ の prefix であるような組 $(x, y)(x \neq y)$ が存在しないことをいう。

prefix code の例b

(A, C, G, T, other) -> (0, 10, 110, 1110, 1111)
(A, C, G, T) は上の例と同じくascii表現されていると見做している。このコードbと上のコードaはどちらもprefix codeであるが、圧縮効率は元のビット列に依存することに注意。

Kraft の不等式

この不等式は、prefix code の簡潔さの限界を与える。任意の（prefix な）コード $C(x)$ に対し、コード長関数 $L_C(x) := length(C(x))$ を定義すると、以下が成り立つ。

$$\sum_x 2^{-L_C(x)} \leq 1$$

MDLでは、モデル選択に関わるのがコード長関数のみであることから、具体的なコードそのものよりもコード長関数が本質的な存在である。そこで、Kraft の不等式を満たすコード長関数が得られれば、それをコードと同一視する。

コードと確率の一対一対応

$2^{-L_C(x)}$ を$P_C(x)$ とかくと、Kraft の不等式は以下の形になる。

$$\sum_x P_C(x) \leq 1$$

このように、コードが与えられるとそれに defective probability, $P_C(x)$ を対応させることができる。ここで重要な抽象化を行う。**コード長が整数であるという制約を忘れることで、x に関するコード長関数と確率分布が一対一対応する。つまり、任意の確率分布$P(x)$に対し、コード長関数が $L_C(x) = -\log P(x)$ （対数の底は本質的ではないが、2と約束する）となるコード C が存在することが言える。**しかも、このコードは、x が $P(x)$ に従って出現するという仮定のもとでは、期待コード長が最小となる、という意味で最適である。

最後の言明を確かめよう。x が $P(x)$に従って現れるとき、コード C' の期待コード長は、

$$\sum_x P(x) L_{C'}(x) = \sum_x P(x) \log P_{C'}(x) \geq \sum_x P(x) \log P(x) = \sum_x P(x) L_C(x)$$

中央の不等式は、Kullback-Leibler divergence が常に非負の値をとることと同義である。

L_2 の定義

以上の考察より、「モデルPの助けを得てデータを圧縮した場合の記述長」の関数系は自然に定まる。モデルPが正しいと考えるならば、最善の圧縮はPに対応するコード長関数(これはコードと同一視されていることに注意)を利用した場合に得られ、記述長は以下となる。

$L_2(D|P) = -\log P(D)$

L_1 の定義（マルコフ連鎖での例）

$L_2(D|P)$ の 定義は一般的で理論的にも修正の余地の少ないものだが、$L_1(P)$は ad hoc な選択が簡便である。Universal coding をきちんと定義すれば、より厳密な記述長が定義できるようだが、ここではマルコフ連鎖を例に $L_1(P)$の定義を概観する。

具体的に、{0,1}* のビット列を$\log K$ 次のマルコフ連鎖でモデル化することを考える。
例えば $K = 4$ (2次のマルコフ連鎖)なら、モデルのパラメタは $P(0|00), P(0|01), P(0|10), P(0|11)$ の4個あり（HMMで出力確率を1とした場合と思う）、これを $\Theta^{(4)} := [0,1]^4$ などと表す。

さて、$\Theta^{(k)} = [0,1]^k$ からの選択を記述するためには、空間を離散化する必要がある。パラメタの空間を幅 $w = 2^{-d}$ の小超立方体へ分割して、その重心だけをパラメタの候補としたものを、離散化したパラメタ空間 $\Theta_d^{(k)}$ とかく。この集合の大きさは $(1/w)^k$ なので、これを一様なコードで符号化するのに $\log (1/w)^k = kd$ bits 要する。（これは、各パラメタを d bits の精度で符号化したことに等しい。）

最後に、k, d を符号化するための $O(\log k), O(\log d)$ を合わせて（これに関しElias codingを参照）、パラメタの記述に要する記述長は
$kd + 2\log k + 2\log d + 2$
となる。

最終形

まとめると、マルコフ連鎖の次数、パラメタは、MDL基準では次の最小値を実現するものが選ばれる。

$$\min_d \min_k \min_{\theta \in \Theta_d^{(k)}} kd + 2\log k + 2\log d - 2\log P_\theta(D)$$

Comments 2

yk_tanigawa ★ 0 15/12/22 11:31

- 情報源符号化定理の下限側との対応があるのは面白いですね。
- 細かいことですが，logの底が2であることをどこかに書いたほうが良いと思います。

yuta_suzuki_hacone ★ 0 15/12/22 12:32

ありがとう、改善していきます。
そんな名前の定理でしたね。参考にした MDL の本 (Grunwald 2007)では Information Inequality と呼ばれてました。（証明はCover & Thomas 2001 を参照とある）
情報源の確率分布を仮定した場合の最適な符号化（と符号長関数）の存在を言えるので、ad hoc な L_1 と違って L_2 の定義はすでに Theoretically sound らしいです。