10-704: Information Processing and Learning **Spring 2012**

Lecture 13: Minimum Description Length

Lecturer: Aarti Singh *Scribes: Aarti Singh*

13.1 Minimum Description Length

The minimum description length (MDL) criteria in machine learning says that the best description of the data is given by the model which compresses it the best. Put another way, learning a model for the data or predicting it is about capturing the regularities in the data and any regularity in the data can be used to compress it. Thus, the more we can compress a data, the more we have *learnt* about it and the better we can predict it.

MDL is also connected to Occam's Razor used in machine learning which states that "other things being equal, a simpler explanation is better than a more complex one." In MDL, the simplicity (or rather complexity) of a model is interpreted as the length of the code obtained when that model is used to compress the data.

The ideal version of MDL is given by the Kolmogorov Complexity, which is defined as the length of the shortest computer program that prints the sequence of observed data and halts. However, the Komogorov Complexity is uncomputable i.e. it can be shown that there exists no computer program that, for every set of data D, when given D as input, returns the shortest program that prints D). Moreover, for finite length sequences, the best program may depend on the sequence itself. For fruther reading on Kolmogorov Complexity, see Chapter 14 of Thomas and Cover.

Practical versions of MDL are either based on one stage universal codes (known as refined MDL) or two-stage codes (known as crude MDL). The refined MDL suggest that, for a given class of models, pick the model which minimizes the worst case redundancy on the data. This leads to precisely the universal models such as mixture models and normalized maximum likelihood (NML) model, which we discussed in last few lectures:

$$\text{Mixture model:} \quad q_\gamma(x^n) = \sum_{q \in Q_\gamma} \pi(q)\ q(x^n) \quad \begin{array}{l}\text{where } \pi \text{ is chosen to be (nearly)} \\ \text{minimax optimal prior for class } Q_\gamma\end{array}$$

$$\text{NML model:} \quad q_\gamma(x^n) = \arg\max_{q \in Q_\gamma} \frac{q(x^n)}{\sum_{x^n} \max_{q \in Q_\gamma} q(x^n)}$$

As we discussed earlier, the mixture models are ususally prefered as they amenable to sequential modeling and hence to design of practical arithmetic codes based on those sequential models.

The crude MDL or two-stage code approach is based on the notion that we can specify the descriptive properties of a model for data in two stages - (i) encode the model with some codelength $L(q)$, (ii) encode the data using the model with codelength $L_q(x^n)$. Now pick the model which minimizes the total codelength of the two-stage code:

$$q_\gamma(x^n) = \arg\min_{q \in Q_\gamma} \{L(q) + L_q(x^n)\}$$

While this procedure can be used for model selection within a class, we will discuss that later. First, we first look at using the two-stage coding approach to select the best model class.

13.2 MDL for Model Class Selection

Suppose we know the universal model q_γ (say mixture or NML) within a given class Q_γ. The best model class γ^* however may be unknown. How can we select the best model class γ^*?

We can use the two-stage MDL coding approach as follows. Encode the data using the universal model for each class $\{Q_\gamma\}_{\gamma\in\Gamma}$, i.e. $L_{q_\gamma}(x^n) = \log 1/q_\gamma(x^n)$. Encode the class using a prefix code for the class index γ. Now pick the model class that minimizes the length of concatenated prefix code:

$$\widehat{\gamma} = \arg\min_{\gamma\in\Gamma}\{L(\gamma) + L_{q_\gamma}(x^n)\}$$

The model selected by this procedure is $q_{\widehat{\gamma}}$, the universal model for class $\widehat{\gamma}$.

If the true class γ^* was known, the model with the smallest description would be q_{γ^*} and its encoding would have length $L_{q_{\gamma^*}}(x^n)$. Lets look at the length of the encoding using the selected model $q_{\widehat{\gamma}}$:

$$L_{q_{\widehat{\gamma}}}(x^n) \le L(\widehat{\gamma}) + L_{q_{\widehat{\gamma}}}(x^n) = \min_{\gamma\in\Gamma}\{L(\gamma) + L_{q_\gamma}(x^n)\}$$

Since $\gamma^* \in \Gamma$, this implies that

$$L_{q_{\widehat{\gamma}}}(x^n) \le L(\gamma^*) + L_{q_{\gamma^*}}(x^n)$$

From homework, we know that a prefix code for integers can be designed so that $L(\gamma^*) = O(\log \gamma^*)$. Thus, the overhead for choosing the model class using the MDL procedure is small - $O(\frac{\log \gamma^*}{n})$ bits/symbol.

This approach is similar to the notion of universal models for hiearchical classes as discussed in last lecture. Here the nested hierarchical models are $\cup_{\gamma\in\Gamma} Q_\gamma$.

13.2.1 Comparison of 2-stage MDL with mixture model

What if we knew the optimal weights to design an optimal mixture code for the entire union of classes $\cup_{\gamma\in\Gamma} Q_\gamma$? Could we have gained much over the two-stage MDL approach? The answer is no. In fact, the length of the two-stage code is sandwiched between the lengths of two mixture models for the union class:

$$-\log Q^L(x^n) \le L_{2-stage}(x^n) \le -\log Q^U(x^n) + \log c_2$$

where $Q^L(x^n) = \sum_{\gamma\in\Gamma} \pi^L(\gamma)\ q_\gamma(x^n)$ with $\pi^L(\gamma) = 2^{-L(\gamma)}/\sum_{\gamma\in\Gamma} 2^{-L(\gamma)}$,
and $Q^U(x^n) = \sum_{\gamma\in\Gamma} \pi^U(\gamma)\ q_\gamma(x^n)$ with $\pi^U(\gamma) = 2^{-2L(\gamma)}/\sum_{\gamma\in\Gamma} 2^{-2L(\gamma)}$.

Proof:

$$L_{2-stage}(x^n) = \min_{\gamma\in\Gamma}\{L(\gamma) - \log q_\gamma(x^n)\} = -\log \max_{\gamma\in\Gamma}[2^{-L(\gamma)} q_\gamma(x^n)].$$

Now, since $\sum_{\gamma\in\Gamma} 2^{-L(\gamma)} \le 1$ by Kraft's inequality, we have:

$$Q^L(x^n) \ge \sum_{\gamma\in\Gamma} 2^{-L(\gamma)}\ q_\gamma(x^n) \ge \max_{\gamma\in\Gamma}[2^{-L(\gamma)} q_\gamma(x^n)] \ge \sum_{\gamma\in\Gamma} 2^{-2L(\gamma)}\ q_\gamma(x^n) = Q^U(x^n)/c_2$$

The result follows by taking the $-\log$ of the above expression. ■

13.3 MDL for model selection

Now lets consider model selection within a class using two-stage MDL as an alternative to mixture model with optimal weights and NML:

$$q_\gamma(x^n) = \arg\min_{q \in Q_\gamma} \{L(q) + L_q(x^n)\}$$

where $L_q(x^n) = \log 1/q(x^n)$ and $L(q)$ is the length of a prefix code for the models. $L(q)$ is a measure of the description length of the model q and hence measures the complexity of the model. Thus, the above approach is also called *complexity penalized or regularized maximum likelihood estimation.*

A **Bayesian interpretation** of the above method is that it is akin to selecting the *maximum aposterior model* since $L(q)$ is equivalent to a $-\log$ prior. Thus, the method is seeking the model which minimizes $-[\log \text{ prior} + \log \text{likelihood}]$, or equivalently, which maximizes the prior $\times$ likelihood (which is $\propto$ posterior).

The length of the two-stage code for the selected model is $L_{2-stage}(x^n) = \min_{q \in Q_\gamma}\{L(q) + L_q(x^n)\}$. Barron and Cover [BC91] showed a bound on the expected redundancy per symbol of such a code as follows. Consider the expected value of the length of the two-stage code with respect to the true data generating distribution p: $L(q) + E_p[\log 1/q(X^n)] = L(q) + D_n(p||q) + H_n(p)$, and define the index of resolvability as

$$R_n(p) = \min_{q \in Q_\gamma} \{L(q) + D_n(p||q)\}.$$

Then, the expected redundancy of the two-stage code

$$\frac{1}{n} E_p[L_{2-stage}(x^n) - \log 1/p(x^n)] \leq \frac{1}{n} R_n(p).$$

Proof:

$$\begin{aligned} E_p[L_{2-stage}(x^n) - \log 1/p(x^n)] &= E_p[\min_{q \in Q_\gamma} \{L(q) + \log p(x^n)/q(x^n)\}] \\ &\leq \min_{q \in Q_\gamma} E_p[L(q) + \log p(x^n)/q(x^n)] \\ &= \min_{q \in Q_\gamma} \{L(q) + D_n(p||q)\} = R_n(p). \end{aligned}$$

The inequality holds using Jensen's inequality since minimum is a concave function. ■

This bound reflects the approximation error vs estimation error (or bias vs variance) tradeoff in machine learning problems, since the more complex the model class, the larger the number of bits needed to describe the model $L(q)$, but the smaller the divergence with respect to the true distribution $D_n(p||q)$. We will investigate this in next class for some parametric and nonparametric classes Q_γ such as Markov models of order γ and histograms density estimates with γ bins. For parametric classes, this will lead to a per symbol expected redundancy bound of $O((\#\text{parameters} \log \text{n})/\text{n})$, which is precisely the same bound obtained using an optimal mixture model as we saw in last few lectures.

13.4 Classification/Regression as Maximum Likelihood estimation

So far we have focused on modeling or density estimation. However, the MDL principle can also be used for classification or regression. In these problems, the goal is essentially equivalent to modeling $p_{Y|X}$ or p_{XY}. We will show this for one concrete setup next.

A typical model for regression is $Y_i = f(X_i) + \epsilon_i$ for $i = 1, \ldots, n$ where $\epsilon_i \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$. The regression problem is equivalent to modeling $p_{Y|X}$ which in this case is $\mathcal{N}(f(X), \sigma^2)$, the Gaussian model indexed by parameter $f(X)$. The length of codeword for encoding the data using this model is

$$\log 1/p_{Y|X}(y^n|x^n) = \log\left(\sqrt{2\pi\sigma^2}\; e^{\frac{1}{2\sigma^2}\sum_{i=1}^n (y_i - f(x_i))^2}\right) \propto \sum_{i=1}^n (y_i - f(x_i))^2$$

Thus, the negative log loss under additive Gaussian noise model is simply the least squares error.

A model class now correspond to assuming a class that the parameter $f(X)$ belongs to, e.g. $F_\gamma = \{f : f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_\gamma x^\gamma\}$ is the class of polynomials of degree γ, or $F_\gamma = \{f : f(x) = \sum_{i=1}^\gamma a_i \phi_i(x)\}$ where $\phi_i(x)$ are nonlinear features or basis functions such as fourier basis, wavelet basis, splines etc.

The two-stage MDL procedure for selecting the best model in a given class would be:

$$f_\gamma = \arg\min_{f \in F_\gamma} \left[L(f) + \frac{1}{n}\sum_{i=1}^n (Y_i - f(X_i))^2\right]$$

This is precisely the *regularized least squares* framework.

We can also select the model class (e.g. order of the polynomial) in addition to selecting the best model within the class, adaptively using a three-step MDL approach:

$$\widehat{\gamma} = \arg\min_{\gamma \in \Gamma} \left\{ L(\gamma) + \min_{f \in F_\gamma} \left[L(f) + \frac{1}{n}\sum_{i=1}^n (Y_i - f(X_i))^2 \right] \right\}$$

References

[BC91] A. Barron and T. Cover, "Minimum Complexity Density Estimation," *IEEE Transactions on Information Theory*, Vol. 37, No.4, July 1991.