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

Lecture 20: Redundancy Lower Bound, Sequential Prediction

Lecturer: Aarti Singh *Scribes: Salman Salamatian*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

20.1 Lower Bound on Redundancy

In the previous discussions (Lectures 10-14) we have found some upper bounds for universal codes. We basically used 2 techniques (and showed they had similar performance asymptotically):

Mixture Models

In this model, we suppose the universal code has the form :

$$Q_{dir} = \sum_{\theta \in \Theta} \pi(\theta) P_\theta$$

where the prior distribution for the parameters $\theta \in \Theta$ is $\pi \sim Dirichlet(\underbrace{\frac{1}{2}, \frac{1}{2}, \ldots, \frac{1}{2}}_{|\mathcal{X}|-1})$ is a Dirichlet distribution over $(|\mathcal{X}| - 1)-$dimensional probability simplex. Also remember that we had found that the maximum redundancy using the above mixture model for the class of *iid* processes was:

$$\max_{\theta \in \Theta} \frac{1}{n} D_n(P_\theta || Q_{dir}) = O(\frac{|\mathcal{X}| - 1}{2} \frac{\log(n)}{n})$$

Two-Stage MDL

In this method, the universal estimator was the one that minimized the sum of the length to describe the model and the length to code message using the model:

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

Again the maximum redundancy was in the order of:

$$O\left(\frac{\#\text{ parameters} \log(n)}{n}\right)$$

where for iid processes the number of parameters is $|\mathcal{X}| - 1$.

Using the redundancy-capacity theorem, we will show that these 2 methods achieve the asymptotic minimax rates. In order to do so, we will find a lower bound for the capacity of a channel (as described in previous lecture) which is same as the minimax redundancy, and show that it is of the same order as the maximum redundancy achieved by the above two methods. Once again, we will take the same format as previous lectures :

$$\theta \longrightarrow X \longrightarrow Z = \hat{\theta}(X)$$

20.1.1 Lower Bound on Redundancy Through Channel Capacity

We know that $C = \max_{\pi(\theta)} I(\theta, X)$. Consider $\pi(\theta) \sim Uniform$ on $\theta \in \Theta$. Moreover call $Z = \hat{\theta}(X)$. Then

$$\begin{aligned} I(\theta, X) &= H(\theta) - H(\theta|X) && (20.1)\\ &\geq H(\theta) - H(\theta|Z) && (20.2)\\ &= H(\theta) - H(\theta - Z|Z) && (20.3)\\ &\geq H(\theta) - H(\theta - Z) && (20.4) \end{aligned}$$

where (20.2) is by Data Processing Inequality, and (20.4) holds since conditioning does not increase entropy. Now suppose there exists an estimator $\hat{\theta} = Z$ such that $E[||Z - \theta||^2] \leq \epsilon$ for any $\theta \in \Theta$. Then

$$\begin{aligned} I(\theta, X) &\geq H(\theta) - \frac{k}{2}\log\left(2\pi e\frac{\epsilon}{k}\right) && (20.5)\\ &= vol(\Theta) - \frac{k}{2}\log\left(2\pi e\frac{\epsilon}{k}\right) && (20.6) \end{aligned}$$

where (20.5) is obtained using the fact that the maximum entropy continuous distribution subject to a 2nd moment constraint $E[||Z - \theta||^2] \leq \epsilon$ is the k-dimensionnal normal distribution, where k is the number of parameters (for *iid* process, $k = |\mathcal{X}| - 1$). Finally we have that

$$C = \max_{\pi(\theta)} I(\theta, X) \geq vol(\Theta) + \frac{k}{2}\log\left(\frac{k}{2\pi e\epsilon}\right) \qquad (20.7)$$

Note that we still have to show that there exists an estimator $\hat{\theta} = Z$ such that $E[||Z - \theta||^2] \leq \epsilon$ for any $\theta \in \Theta$ For iid processes, $\theta = (p_1, \ldots, p_{|\mathcal{X}|-1})$, the probability of each symbol. Consider $\hat{\theta} = (\hat{p}_1, \ldots, \hat{p}_{|\mathcal{X}|-1})$, where $\hat{p_j} = \frac{n_j}{n}$ for $j = 1, \ldots, |\mathcal{X}| - 1$ the simple fraction estimator. Note that $E[\hat{p}_j] = p_j$ and hence $E[\hat{\theta}] = \theta$. Also, $E[\|\hat{\theta} - \theta\|^2] = var(\hat{\theta}) = \sum_{j=1}^{|\mathcal{X}|-1} E[(\hat{p}_j - p_j)^2] = \sum_{j=1}^{|\mathcal{X}|-1} var(\hat{p}_j)$. It is easy to verify that $var(\hat{p}_j) = O(1/n)$. Thus,

$$E\left[\|\hat{\theta} - \theta\|^2\right] = O\left(\frac{|\mathcal{X}| - 1}{n}\right) := \epsilon$$

Since $vol(\Theta) < \infty$ is a constant and for iid processes $k = |\mathcal{X}| - 1$, this gives:

$$C \geq constant + \frac{|\mathcal{X}| - 1}{2}\log\left(\frac{n}{2\pi e}\right) \qquad (20.8)$$

Hence, the lower bound for the channel capacity per symbol (above expression divided by n), which is equal to minimax redundancy by the redundancy-capacity theorem, and the upper bounds (of mixture and MDL) match, those estimators are hence minimax optimal.

Remark: Notice that the above bound was derived using a uniform prior on the parameter space Θ which is not the least favorable prior. Often computing the least favorable prior is not easy, but using any other prior gives us a *lower bound* on the channel capacity and minimax redundancy. For parametric classes, the lower bound given by any prior is the same up to constants (in fact the best first-order term in capacity is given by the Dirichlet prior used in the mixture model we considered for upper bound). However, the same is not true for non-parametric classes, where the choice of the prior used to compute lower bound needs to be judicious to yield a large lower bound (if lower bound is not as large as possible, then it may not match the upper bound and may not reflect the complexity of the minimax problem).

20.2 General Loss Function

So far we have only discussed shown the connection of modeling or prediction under log loss with coding and information theory. However we would like to generalize this discussion for applications in classification and regression for example. We will consider here a general loss function $loss(\cdot,\cdot) > 0$ that is upper bounded by a constant L:

$$L = \max_{X,\hat{X}} loss(X, \hat{X})$$

The approach we will take here, will be to consider sequential prediction, that is given $X_1, X_2, \ldots, X_{t-1}$ we want to guess X_t (Note that standard regression and classification are included in this set up and correspond to $X_1, \ldots, X_{t-1}$ being the iid tranining points and X_t being the test point). We define $b_t : \mathcal{X}^{t-1} \to \mathcal{X}$. We define the bayes estimator with respect to P_θ as :

$$b_t^{Bayes(P_\theta)} = \operatorname*{argmin}_{b_t} E_\theta \left[loss(X_t, \hat{X}_t) \right]$$

where P_θ is the actual distribution of X_t and $\hat{X}_t = b_t(X^{t-1})$.
If we do not know P_θ, we might use another distribution Q. We can define the bayes estimator for an arbitrary distribution Q:

$$b_t^{Bayes(Q)} = \operatorname*{argmin}_{b_t} E_Q \left[loss(X_t, \hat{X}_t) \right]$$

Q could for example be the empirical distribution.

Lemma 20.1

$$E_\theta \left[loss(b_t^{Bayes(Q)}, x_t) - loss(b_t^{Bayes(P_\theta)}, x_t) \right] \leq L\sqrt{2D(P_\theta||Q)}$$

Since $D(P_\theta||Q)$ is the excess risk under log loss, this gives a relation between performance under log-loss and any other loss function. In other words we can use the minimax estimator for log loss and still get an upper bound on the excess risk for any other bounded loss function. (There is a way to handle unbounded but smooth loss functions, but we won't discuss that here.) In many cases, the upper bound on other loss funtions is also tight (i.e. there exist matching lower bounds). However, there are cases where it is possible to obtain a better bound on the other loss functions than what is provided by using the minimax estimator for log loss and plugging that in to obtain a predictor for other loss functions.
Proof:

$$\begin{aligned}
&E_\theta \left[loss(b_t^{Bayes(Q)}, x_t) - loss(b_t^{Bayes(P_\theta)}, x_t) \right] && (20.9)\\
&= \sum_{x_t} P_\theta(x_t) \left[loss(b_t^{Bayes(Q)}, x_t) - loss(b_t^{Bayes(P_\theta)}, x_t) \right] && (20.10)\\
&\leq \sum_{x_t} \left(|P_\theta(x_t) - Q(x_t)| + Q(x_t) \right) \left[loss(b_t^{Bayes(Q)}, x_t) - loss(b_t^{Bayes(P_\theta)}, x_t) \right] && (20.11)\\
&\leq \sum_{x_t} |P_\theta(x_t) - Q(x_t)|\, L && (20.12)\\
&= L\, TV(P_\theta, Q) && (20.13)\\
&\leq L\sqrt{2D(P_\theta||Q)} && (20.14)
\end{aligned}$$

Where (20.12) follows by definition of the Bayes estimator ($b_t^{Bayes(Q)}$ minimizes loss function if x_t is distributed as Q), TV is the total variation, and (20.14) is by Pinsker's Inequality. ■

20.3 Prediction of sequence/Online Learning

Consider a sequence $X_1, X_2, \ldots, X_{t-1}\ \ X_t$. In online learning we do not want to assume that X_t is stochastic conditioned on previous values $X_1, \ldots, X_{t-1}$. That is, we suppose that an adversary selects X_t at every iteration.

To make the problem well-posed, the goal in online learning is to compare the performance of the predictor Q we come up with to some predictors (or oracles or experts) $\{P_\theta\}_{\theta\in\Theta}$. A deterministic predictor would not work in an adversarial setting and hence the predictor Q has to a randomized predictor.

Definition 20.2 *We define the regret as the redundancy we suffer compared to the best oracle:*

$$\textit{regret(Q)} \equiv \log\left(\frac{1}{Q(x^n)}\right) - \min_{\theta\in\Theta}\log\left(\frac{1}{P_\theta(x^n)}\right)$$

where $Q(X^n) = \prod_{t=1}^{n} Q(X_t|X^{t-1})$.

We also define the *worst-case* regret:

Definition 20.3 *The* worst-case *regret is defined as*

$$\begin{aligned}\bar{\textit{regret}}(Q) &= \max_{x^n}\left[\log\left(\frac{1}{Q(x^n)}\right) - \min_{\theta\in\Theta}\log\left(\frac{1}{P_\theta(x^n)}\right)\right] \\ &= \max_{\theta\in\Theta}\max_{x^n}\left[\log(\frac{1}{Q(x^n)}) - \log(\frac{1}{P_\theta(x^n)})\right]\end{aligned}$$

Note that replacing the $\max_{x^n}$ with $E_{x^n\sim P_\theta}[.]$ in the last line gives us the usual KL-divergence $D(P_\theta||Q)$. Thus, the online learning setting is same as considering the worst case redundancy instead of expected redundancy. Recall that we discussed worst-case redundancy in Lecture 10 and showed that the optimal solution is given by the normalized maximum likelihood distribution Q_{NML}. However, since Q_{NML} does not have the nice consistency property required in a sequential setting (refer to Lecture 10), we instead use Q is a mixture of the predictors $\{P_\theta\}_{\theta\in\Theta}$:

$$Q(x^t) = \sum_\theta \pi(\theta)P_\theta(x^t)$$

In this case we have that the conditional probability of X_t given the observed sequence X^{t-1} is :

$$\begin{aligned}Q(x_t|x^{t-1}) = \frac{Q(x^t)}{Q(x^{t-1})} &= \frac{\sum_\theta \pi(\theta)P_\theta(x^t)}{\sum_{\theta'}\pi(\theta')P_{\theta'}(x^{t-1})} &(20.15)\\ &= \frac{\sum_\theta \pi(\theta)P_\theta(x_t|x^{t-1})P_\theta(x^{t-1})}{\sum_{\theta'}\pi(\theta')P_{\theta'}(x^{t-1})} &(20.16)\\ &= \sum \omega(\theta)P_\theta(x_t|x^{t-1}) &(20.17)\end{aligned}$$

where $\omega(\theta) = \frac{\pi(\theta)P_\theta(x^{t-1})}{\sum_{\theta'}\pi(\theta')P_{\theta'}(x^{t-1})}$.

The optimal solution is thus a mixture of the conditional oracle/expert distributions where the mixing weights are changed from iteration to iteration based on how well oracle distributions have done on the sequence thus far i.e. $P_\theta(x^{t-1})$ - likelihood of x^{t-1}. This "adaptation" of the mixing weights is implemented through a Weighted Majority Algorithm proposed independently in Vovk'90 and Littlestone-Warmuth'94, which also works not only for log loss but any general loss function.

Weighted Majority Algorithm

Initialize : $w_0(\theta) = \frac{1}{|\Theta|}, \ \forall\theta\in\Theta$
For $t \geq 1$:

1. Predict using $\sum_{\theta \in \Theta} w_t(\theta) \exp^{-\nu \, loss(b_t^\theta, \mathcal{X})}$

   Remark 1: Note that $\nu = 1$ and loss= -ve log likelihood gives the previous mixture model.
   Remark 2: In classification and regression problems, an oracle b_t^θ is drawn from the collection of all oracles with distribution $w_t(\theta)$ instead of using the mixture of oracles specified above.

2. We update the priors $w_t(\theta)$ using :

$$w_{t+1}(\theta) = \frac{w_t(\theta) \exp^{-\nu \, loss(b_t^\theta, x_t)}}{\sum_{\theta \in \Theta} w_t(\theta) \exp^{-\nu \, loss(b_t^\theta, x_t)}}$$

Remark: Boosting is also an example of this framework where the oracles are data points, and the weights corresponding to weights on the data points are updated exponentially based on the loss incurred at that data point at each iteration. However, there is a key difference that the ν is not fixed but is also updated at each iteration.