In this lecture, we discuss a unified theoretical framework of statistics proposed by Abraham Wald, which is named statistical decision theory.  

1 Goals

1. **Evaluation:** The theoretical framework should aid fair comparisons between algorithms (e.g., maximum entropy vs. maximum likelihood vs. method of moments).

2. **Achievability:** The theoretical framework should be able to inspire the constructions of statistical algorithms that are (nearly) optimal under the optimality criteria introduced in the framework.

2 Basic Elements of Statistical Decision Theory

1. **Statistical Experiment:** A family of probability measures $\mathcal{P} = \{P_\theta : \theta \in \Theta\}$, where $\theta$ is a parameter and $P_\theta$ is a probability distribution indexed by the parameter.

2. **Data:** $X \sim P_\theta$, where $X$ is a random variable observed for some parameter value $\theta$.

3. **Objective:** $g(\theta)$, e.g., inference on the entropy of distribution $P_\theta$.

4. **Decision Rule:** $\delta(X)$. The decision rule need not be deterministic. In other words, there could be a probabilistically defined decision rule with an associated $P_{\delta|X}$.

5. **Loss Function:** $L(\theta, \delta)$. The loss function tells us how bad we feel about our decision once we find out the true value of the parameter $\theta$ chosen by nature.

**Example:** $P_\theta(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-\theta)^2}{2}}$, $g(\theta) = \theta$, and $L(\theta, \delta) = (\theta - \delta)^2$. In other words, $X$ is normally distributed with mean $\theta$ and unit variance $X \sim N(\theta, 1)$, and we are trying to estimate the mean $\theta$. We judge our success (or failure) using mean-square error.

3 Risk Function

**Definition 1** (Risk Function).

$$R(\theta, \delta) \triangleq \mathbb{E}[L(\theta, \delta(X))]$$

$$= \int L(\theta, \delta(x))P_\theta(dx)$$

$$= \iint L(\theta, \delta)P_{\delta|X}(d\delta|x)P_\theta(dx).$$

\[1\]

A risk function evaluates a decision rule’s success over a large number of experiments with fixed parameter value \( \theta \). By the law of large numbers, if we observe \( X \) many times independently, the average empirical loss of the decision rule \( \delta \) will converge to the risk \( R(\theta, \delta) \).

Even after determining risk functions of two decision rules, it may still be unclear which is better. Consider the example of Figure 1. Two different decision rules \( \delta_1 \) and \( \delta_2 \) result in two different risk functions \( R(\theta, \delta_1) \) and \( R(\theta, \delta_2) \) evaluated over different values on the parameter \( \theta \). The first decision rule \( \delta_1 \) is inferior for low and high values of the parameter \( \theta \) but is superior for the middle values. Thus, even after computing a risk function \( R \), it can still be unclear which decision rule is better. We need new ideas to enable us to compare different decision rules.

4 Optimality Criterion of Decision Rules

Given the risk function of various decision rules as a function of the parameter \( \theta \), there are various approaches to determining which decision rule is optimal.

4.1 Restrict the Competitors

This is a traditional set of methods that were overshadowed by other approaches that we introduce later. A decision rule \( \delta' \) is eliminated (or formally, is inadmissible) if there are any other decision rules \( \delta \) that are strictly better, i.e., \( R(\theta, \delta') \geq R(\theta, \delta) \) for any \( \theta \in \Theta \) and the inequality becomes strict for at least one \( \theta_0 \in \Theta \). However, the problem is that many decision rules cannot be eliminated in this way and we still lack a criterion to determine which one is better. Then to aid in selection, the rationale of the approach of restricting competitors is that only decision rules that are members of a certain decision rule class \( D \) are considered. The advantage is that, sometimes all but one decision rule in \( D \) is inadmissible, and we just use the only admissible one.

1. **Example 1**: Class of unbiased decision rules \( \mathcal{D}' = \{ \delta : \mathbb{E}[\delta(X)] = g(\theta), \forall \theta \in \Theta \} \)

2. **Example 2**: Class of invariant estimators.

However, a serious drawback of this approach is that \( \mathcal{D} \) may be an empty set for various decision theoretic problems.
4.2 Bayesian: Average Risk Optimality

The idea is to use averaging to reduce the risk function \( R(\theta, \delta) \) to a single number for any given \( \delta \).

**Definition 2** (Average risk under prior \( \Lambda(d\theta) \)).

\[
r(\Lambda, \delta) = \int R(\theta, \delta) \Lambda(d\theta)
\]

Here \( \Lambda \) is the is the prior distribution, a probability measure on \( \Theta \). The Bayesians and the frequentists disagree about \( \Lambda \); namely, the frequentists do not believe the existence of the prior. However, there do exist more justifications of the Bayesian approach than the interpretation of \( \Lambda \) as prior belief: indeed, the complete class theorem in statistical decision theory asserts that in various decision theoretic problems, all the admissible decision rules can be approximated by Bayes estimators. ²

**Definition 3** (Bayes estimator).

\[
\delta_\Lambda = \arg \min_\delta r(\Lambda, \delta)
\]

The Bayes estimator \( \delta_\Lambda \) can usually be found using the principle of computing posterior distributions. Note that

\[
r(\Lambda, \delta) = \iint L(\theta, \delta) P_{\theta|X}(d\delta|x) P_{\theta}(dx) \Lambda(d\theta)
\]

\[
= \int \left( \iiint L(\theta, \delta) P_{\theta|X}(d\delta|x) P_{\theta}(d\theta|x) \right) P_X(dx)
\]

where \( P_X(dx) \) is the marginal distribution of \( X \) and \( P_{\theta|X}(d\theta|x) \) is the posterior distribution of \( \theta \) given \( X \). In Equation 6, \( P_{\theta}(dx) \Lambda(d\theta) \) is the joint distribution of \( \theta \) and \( X \). In Equation 7, we only have to minimize the portion in parentheses to minimize \( r(\Lambda, \delta) \) because \( P_X(dx) \) doesn’t depend on \( \delta \).

**Theorem 4.** ³ Under mild conditions,

\[
\delta_\Lambda(x) = \arg \min_\delta \mathbb{E}[L(\theta, \delta)|X = x]
\]

\[
= \arg \min_{\delta P_{\theta|X}} \iint L(\theta, \delta) P_{\theta|X}(d\delta|x) P_{\theta}(d\theta|x)
\]

**Lemma 5.** If \( L(\theta, \delta) \) is convex in \( \delta \), it suffices to consider deterministic rules \( \delta(x) \).

**Proof** Jensen’s inequality:

\[
\iint L(\theta, \delta) P_{\theta|X}(d\delta|x) P_{\theta}(d\theta|x) \geq \int L(\theta, \delta P_{\theta|X}(d\delta|x)) P_{\theta}(d\theta|x).
\]

**Examples**

1. \( L(\theta, \delta) = (g(\theta) - \delta)^2 \Rightarrow \delta_\Lambda(x) = \mathbb{E}[g(\theta)|X = x] \). In other words, the Bayes estimator under squared error loss is the conditional expectation of \( g(\theta) \) given \( x \).

2. \( L(\theta, \delta) = |g(\theta) - \delta| \Rightarrow \delta_\Lambda(x) \) is any median of the posterior distribution \( P_{\theta|X=x} \).

3. \( L(\theta, \delta) = 1(g(\theta) \neq \delta) \Rightarrow \delta_\Lambda(x) = \arg \max_{\theta} P_{\theta|X}(\theta|x) = \arg \max_{\theta} P_{\theta}(x) \Lambda(\theta) \). In other words, an indicator loss function results in a maximum a posteriori (MAP) estimator decision rule⁴.


³See Theorem 1.1, Chapter 4 of Lehmann EL, Casella G. Theory of point estimation. Springer Science & Business Media; 1998

⁴Next week, we will cover special cases of \( P_{\theta} \) and how to solve Bayes estimator in a computationally efficient way. In the general case, however, computing the posterior distribution may be difficult.
4.3 Frequentist: Worst-Case Optimality (Minimax)

**Definition 6** (Minimax estimator). The decision rule $\delta^*$ is minimax among all decision rules in $D$ iff

$$\sup_{\theta \in \Theta} R(\theta, \delta^*) = \inf_{\delta \in D} \sup_{\theta \in \Theta} R(\theta, \delta).$$

(11)

4.3.1 First observation

Since

$$R(\theta, \delta) = \iint L(\theta, \delta) P_{\delta|X} (d\delta|x) P_{\theta}(dx)$$

(12)

is linear in $P_{\theta|X}$, this is a convex function in $\delta$. The supremum of a convex function is convex, so finding the optimal decision rule is a convex optimization problem. However, solving this convex optimization problem may be computationally intractable. For example, it may not even be computationally tractable to compute the supremum of the risk function $R(\theta, \delta)$ over $\theta \in \Theta$. Hence, finding the exact minimax estimator is usually hard.

4.3.2 Second observation

Due to the previous difficulty of finding the exact minimax estimator, we turn to another goal: we wish to find an estimator $\delta'$ such that

$$\inf_{\delta} \sup_{\theta} R(\theta, \delta) \leq \sup_{\theta} R(\theta, \delta') \leq c \cdot \inf_{\delta} \sup_{\theta} R(\theta, \delta)$$

(13)

where $c > 1$ is a constant. The left inequality is trivially true. For the right inequality, in practice one can usually choose some specific $\delta'$ and evaluate an upper bound of $\sup_{\theta} R(\theta, \delta')$ explicitly. However, it remains to find a lower bound of $\inf_{\delta} \sup_{\theta} R(\theta, \delta)$. To solve the problem (and save the world), we can use the minimax theorem.

**Theorem 7** (Minimax Theorem (Sion-Kakutani)). Let $\Lambda$, $X$ be two compact, convex sets in some topologically vector spaces. Let function $H(\lambda, x) : \Lambda \times X \to \mathbb{R}$ be a continuous function such that:

1. $H(\lambda, \cdot)$ is convex for any fixed $\lambda \in \Lambda$
2. $H(\cdot, x)$ is concave for any fixed $x \in X$.

Then

1. **Strong duality**: $\max_{\lambda} \min_{x} H(\lambda, x) = \min_{x} \max_{\lambda} H(\lambda, x)$
2. **Existence of Saddle point**:

$$\exists (\lambda^*, x^*) : \quad H(\lambda^*, x^*) \leq H(\lambda, x^*) \leq H(\lambda^*, x) \quad \forall \lambda \in \Lambda, x \in X.$$

The existence of saddle point implies the strong duality.

We note that other than the strong duality, the following weak duality is always true without assumptions on $H$:

$$\sup_{\lambda} \inf_{x} H(\lambda, x) \leq \inf_{x} \sup_{\lambda} H(\lambda, x)$$

(14)

We define the quantity $r_{\Lambda} \equiv \inf_{\delta} r(\Lambda, \delta)$ as the Bayes risk under prior distribution $\Lambda$. We have the following lines of arguments using weak duality:

$$\inf_{\delta} \sup_{\theta} R(\theta, \delta) = \inf_{\delta} \sup_{\theta} r(\Lambda, \delta)$$

(15)

$$\geq \sup_{\Lambda} \inf_{\delta} r(\Lambda, \delta)$$

(16)

$$= \sup_{\Lambda} r_{\Lambda}$$

(17)
Equation (17) gives us a strong tool for lower bounding the minimax risk: for any prior distribution $\Lambda$, the Bayes risk under $\Lambda$ is a lower bound of the corresponding minimax risk. When the condition of the minimax theorem is satisfied (which may be expected due to the bilinearity of $r(\Lambda, \delta)$ in the pair $(\Lambda, \delta)$), equation (16) achieves equality, which shows that there exists a sequence of priors such that the corresponding Bayes risk sequence converges to the minimax risk.

In practice, it suffices to choose some appropriate prior distribution $\Lambda$ in order to solve the (nearly) minimax estimator.

5 Decision Theoretic Interpretation of Maximum Entropy

Definition 8 (Logarithmic Loss). For a distribution $P$ over $\mathcal{X}$ and $x \in \mathcal{X}$,

$$L(x, P) \triangleq \log \frac{1}{P(x)},$$

(18)

Note that $P(x)$ is the discrete mass on $x$ in the discrete case, and is the density at $x$ in the continuous case.

By definition, if $P(x)$ is small, then the loss is large. Now we see how the maximum likelihood estimator reduces to the so-called maximum entropy under the logarithmic loss function. Specifically, suppose we have functions $r_1(x), \cdots, r_k(x)$ and compute the following empirical averages:

$$\beta_i \triangleq \frac{1}{n} \sum_{j=1}^{n} r_i(x_j) = \mathbb{E}_{P_n}[r_i(X)], \quad i = 1, \cdots, k$$

(19)

where $P_n$ denotes the empirical distribution based on $n$ samples $x_1, \cdots, x_n$. By the law of large numbers, it is expected that $\beta_i$ is close to the true expectation under $P$, so we may restrict the true distribution $P$ in the following set:

$$\mathcal{P} = \{ P : \mathbb{E}_P[r_i(X)] = \beta_i, 1 \leq i \leq k \}.$$  

(20)

Also, denote by $\mathcal{Q}$ the set of all probability distributions which belong to the exponential family with central statistics $r_1(x), \cdots, r_k(x)$, i.e.,

$$\mathcal{Q} = \{ Q : Q(x) = \frac{1}{Z(\lambda)} e^{\sum_{i=1}^{k} \lambda_i r_i(x)}, \quad \forall x \in \mathcal{X} \}$$

(21)

where $Z(\lambda)$ is the normalization factor.

Now we consider two estimators of $P$. The first one is the maximum likelihood estimator over $\mathcal{Q}$, i.e.,

$$P_{ML} = \arg\max_{P \in \mathcal{Q}} \frac{1}{n} \sum_{j=1}^{n} \log P(x_j).$$

(22)

The second one is the maximum entropy estimator over $\mathcal{P}$, i.e.,

$$P_{ME} = \arg\max_{P \in \mathcal{P}} H(P)$$

(23)

where $H(P)$ is the entropy of $P$, which is $H(P) = \sum_{x \in \mathcal{X}} -P(x) \log P(x)$ in the discrete case, and $H(P) = \int_{\mathcal{X}} -P(x) \log P(x) dx$ in the continuous case.

The key result is as follows:

Theorem 9. In the above setting, we have $P_{ML} = P_{ME}$.

Proof We begin with the following lemma, which is the key motivation for the use of the cross-entropy loss in machine learning.

**Lemma 10.** The true distribution minimizes the expected logarithmic loss:

\[ P = \arg \min_Q \mathbb{E}_P L(x, Q) \]  

and the corresponding minimum \( \mathbb{E}_P L(x, P) \) is the entropy of \( P \).

**Proof** The assertion follows from the non-negativity of the KL divergence \( D(P \parallel Q) \geq 0 \), where \( D(P \parallel Q) = \mathbb{E}_P \log \frac{dP}{dQ} \).

Equipped with this lemma, and noting that \( \mathbb{E}_P [\log \frac{1}{Q(x)}] \) is linear in \( P \) and convex in \( Q \), by the minimax theorem we have

\[ \max_{P \in \mathcal{P}} H(P) = \max_{P \in \mathcal{P}} \min_Q \mathbb{E}_P [\log \frac{1}{Q(x)}] = \min_Q \max_{P \in \mathcal{P}} \mathbb{E}_P [\log \frac{1}{Q(x)}] \]  

In particular, we know that \((P_{\text{ME}}, P_{\text{ME}})\) is a saddle point in \( \mathcal{P} \times \mathcal{M} \), where \( \mathcal{M} \) denotes the set of all probability measures on \( \mathcal{X} \). We first take a look at \( P_{\text{ME}} \) which maximizes the LHS. By variational methods, it is easy to see that the entropy-maximizing distribution \( P_{\text{ME}} \) belongs to the exponential family \( Q \), i.e.,

\[ P_{\text{ME}}(x) = \frac{1}{Z(\lambda)} \sum_{i=1}^k \lambda_i r_i(x) \]  

for some parameters \( \lambda_1, \ldots, \lambda_k \).

Now by the definition of saddle points, for any \( Q \in Q \subset \mathcal{M} \) we have

\[ \mathbb{E}_{P_{\text{ME}}} [\log \frac{1}{P_{\text{ME}}(x)}] \leq \mathbb{E}_{P_{\text{ME}}} [\log \frac{1}{Q(x)}] \]  

while for any \( Q \in Q \) with possibly different coefficients \( \lambda' \),

\[ \mathbb{E}_{P_{\text{ME}}} [\log \frac{1}{Q(x)}] = \mathbb{E}_{P_{\text{ME}}} [\log Z(\lambda') - \sum_i \lambda'_i r_i(x)] \]  

[since \( Q \in Q \)]

\[ = \log Z(\lambda') - \sum_i \lambda'_i \beta_i \]  

[since \( P_{\text{ME}} \in \mathcal{P} \)]

\[ = \mathbb{E}_{P_n} [\log Z(\lambda') - \sum_i \lambda'_i r_i(x)] \]  

[since \( P_n \in \mathcal{P} \)]

\[ = \mathbb{E}_{P_n} [\log \frac{1}{Q(x)}] \]  

\[ = \frac{1}{n} \sum_{j=1}^n \log \frac{1}{Q(x_j)} \]  

As a result, the saddle point condition reduces to

\[ \frac{1}{n} \sum_{j=1}^n \log \frac{1}{P_{\text{ME}}(x_j)} \leq \frac{1}{n} \sum_{j=1}^n \log \frac{1}{Q(x_j)}, \quad \forall Q \in Q \]  

establishing the fact that \( P_{\text{ME}} = P_{\text{ML}} \). \( \square \)

In this lecture, we define the Hidden Markov Processes (HMP) as well as the forward-backward recursion method that efficiently estimates the posteriors in HMPs.

1 Markov Chain

We begin by introducing the Markov triplet, which is useful in the design and analysis of algorithms for state estimation in Hidden Markov Processes.

Let $X,Y,Z$ denote discrete random variables. We say that $X,Y,Z$ form a Markov triplet (which we denote as $X \sim Y \sim Z$) given the following relationships:

$$X \sim Y \sim Z \iff p(x,z|y) = p(x|y)p(z|y)$$

$$\iff p(z|x,y) = p(z|y)$$

If $X,Y,Z$ form a Markov triplet, their joint distribution enjoys the important property that it can be factored into the product of two functions $\phi_1$ and $\phi_2$.

Lemma 1. $X \sim Y \sim Z \iff \exists \phi_1, \phi_2 \text{ s.t. } p(x,y,z) = \phi_1(x,y)\phi_2(y,z)$.

Proof (Proof of necessity) Assume that $X \sim Y \sim Z$. Then by definition of the Markov chain, we have

$$p(x,y,z) = p(x|y,z)p(y,z) = p(x|y)p(y,z).$$

Now taking $\phi_1(x,y) = p(x,y)$ and $\phi_2(y,z) = p(y,z)$ proves this direction.

(Proof of sufficiency) Now, the functions $\phi_1, \phi_2$ exist as mention in the statement. Then we have:

$$p(x,y,z) = \frac{p(x,y,z)}{p(y,z)} = \frac{p(x,y,z)}{\sum_{\tilde{x}} p(\tilde{x},y,z)} = \frac{\phi_1(x,y)\phi_2(y,z)}{\phi_2(y,z)\sum_{\tilde{x}} \phi_1(\tilde{x},y)} = \frac{\phi_1(x,y)}{\sum_{\tilde{x}} \phi_1(\tilde{x},y)}.$$

The final equality shows that $p(x|y,z)$ is not a function of $z$ and hence, $X \sim Y \sim Z$ follows by definition.

2 Hidden Markov Processes

Based on our understanding of the Markov Triplet, we define the Hidden Markov Process (HMP) as follows:

$\textit{Reading: Ephraim & Merhav 2002, lectures 9,10 of 2016 notes}$
Definition 2 (Hidden Markov Process). The process \( \{(X_n, Y_n)\}_{n \geq 1} \) is a Hidden Markov Process if

- \( \{X_n\}_{n \geq 1} \) is a Markov process, i.e.,
  \[ \forall \ n \geq 1 : \ p(x^n) = \prod_{t=1}^{n} p(x_t|x_{t-1}), \]
  where by convention, we take \( X_0 \equiv 0 \) and hence \( p(x_1|x_0) = p(x_1) \).

- \( \{Y_n\}_{n \geq 1} \) is determined by a memoryless channel; i.e.,
  \[ \forall \ n \geq 1 : \ p(y^n|x^n) = \prod_{t=1}^{n} p(y_t|x_t). \]

The processes \( \{X_n\}_{n \geq 1} \) and \( \{Y_n\}_{n \geq 1} \) are called the state process and the noisy observations, respectively.

In view of the above definition, we can derive the joint distribution of the state and the observation as follows:

\[ p(x^n, y^n) = p(x^n)p(y^n|x^n) = \prod_{t=1}^{n} p(x_t|x_{t-1})p(y_t|x_t). \]

The main problem in the Hidden Markov Models is to compute the the posterior probability of the state at any time, given all the observations up to that time, i.e. \( p(x_t|y^t) \). The naive approach to do this is to simply use the the definition of the conditional probability:

\[ p(x_t|y^t) = \frac{p(x_t, y^t)}{p(y^t)} = \frac{\sum_{x_t} p(y^t|x_t, x^{t-1})p(x_t, x^{t-1})}{\sum_{\tilde{x}_t} \sum_{x_t} p(y^t|\tilde{x}_t, x^{t-1})p(\tilde{x}_t, x^{t-1})}. \]

The above approach needs exponentially many computations both for the numerator and the denominator. To avoid this problem, we develop an efficient method for computing the posterior probabilities using forward recursion. Before getting to the algorithm, we establish some conditional independence relations.

3 Conditional Independence Relations

We introduce some conditional independence relations which help us find more efficient ways of calculating the posterior.

1. \( (X^{t-1}, Y^t) - X_t - (X_{t+1}^{n}, Y_{t+1}^{n}) \):

   **Proof** We have:

   \[ p(x^n, y^n) = \prod_{t=1}^{n} p(x_t|x_{t-1})p(y_t|x_t) \]

   \[ = \left[ \prod_{t=1}^{t} p(x_t|x_{t-1})p(y_t|x_t) \right] \left[ \prod_{t=t+1}^{n} p(x_t|x_{t-1})p(y_t|x_t) \right] \]

   \[ = \phi_1(x_{t-1}, (x^t, y^t)) \phi_2(x_{t+1}, (x_{t+1}, y_{t+1})) \]

   The statement then follows from Lemma 1. \( \square \)
2. \((X^{t-1}, Y^{t-1}) - X_t - (X^{n}_{t+1}, Y^{n}_{t}):\)

**Proof** Note that if \(\{X_i, Y_i\}_{i=1}^{n}\) is a Hidden Markov Process, then \(\{X_{n-i}, Y_{n-i}\}_{i=1}^{n}\) will also be a Hidden Markov Process. Applying (a) to this reversed process proves (b). □

3. \(X^{t-1} - (X_t, Y^{t-1}) - (X^{n}_{t+1}, Y^{n}_{t}):\)

**Proof** Left as an exercise. □

### 4 Causal Inference via Forward Recursion

We now derive the forward recursion algorithm as an efficient method to sequentially compute the causal posterior probabilities.

Note that we have

\[
p(x_t|y^t) = \frac{p(x_t, y^t)}{\sum_{\tilde{x}_t} p(\tilde{x}_t, y^t)}
\]

\[
= \frac{p(x_t, y^{t-1})p(y_t|x_t, y^{t-1})}{\sum_{\tilde{x}_t} p(\tilde{x}_t, y^{t-1})p(y_t|x_t, y^{t-1})}
\]

\[
= \frac{p(y^{t-1})p(x_t|y^{t-1})p(y_t|x_t)}{\sum_{\tilde{x}_t} p(\tilde{x}_t, y^{t-1})p(y_t|x_t, y^{t-1})} \quad \text{(by (b))}
\]

\[
= \frac{p(y^{t-1})p(x_t|y^{t-1})p(y_t|x_t)}{\sum_{\tilde{x}_t} p(\tilde{x}_t, y^{t-1})p(y_t|x_t, y^{t-1})}
\]

Define \(\alpha_t(x_t) = p(x_t|y^t)\) and \(\beta_t(x_t) = p(x_t|y^{t-1})\), then the above computation can be summarized as

\[
\alpha_t(x_t) = \frac{\beta_t(x_t)p(y_t|x_t)}{\sum_{\tilde{x}_t} \beta_t(\tilde{x}_t)p(y_t|x_t, y^{t-1})}.
\]  (3)

Similarly, we can write \(\beta_t\) as

\[
\beta_{t+1}(x_{t+1}) = p(x_{t+1}|y^t)
\]

\[
= \sum_{x_t} p(x_{t+1}, x_t|y^t)
\]

\[
= \sum_{x_t} p(x_t|y^t)p(x_{t+1}|y^t, x_t)
\]

\[
= \sum_{x_t} p(x_t|y^t)p(x_{t+1}|x_t) \quad \text{(by (a))}.
\]

The above equation can be summarized as

\[
\beta_{t+1}(x_{t+1}) = \sum_{x_t} \alpha_t(x_t)p(x_{t+1}|x_t).
\]  (4)

Equations (3) and (4) indicate that \(\alpha_t\) and \(\beta_t\) can be sequentially computed based on each other for \(t = 1, \cdots, n\) with the initialization \(\beta(x_1) = p(x_1)\). Hence, with this simple algorithm, the causal inference can be done efficiently in terms of both computation and memory. This is called the *forward recursion*. 

---

3
5 Non-causal Inference via Backward Recursion

Now suppose that we are interested in finding $p(x_t|y^n)$ for $t = 1, 2, \cdots, n$ in a non-causal manner. For this purpose, we can derive a backward recursion algorithm as follows.

$$p(x_t|y^n) = \sum_{x_{t+1}} p(x_t, x_{t+1}|y^n)$$

$$= \sum_{x_{t+1}} p(x_{t+1}|y^n) p(x_t|x_{t+1}, y^t) \quad \text{(by (c))}$$

$$= \sum_{x_{t+1}} p(x_{t+1}|y^n) \frac{p(x_t|y^t)p(x_{t+1}|x_t)}{p(x_{t+1}|y^t)} \quad \text{(by (a))}$$

Now, let $\gamma_t(x_t) = p(x_t|y^n)$. Then, the above equation can be summarized as

$$\gamma_t(x_t) = \sum_{x_{t+1}} \gamma_{t+1}(x_{t+1}) \frac{\alpha_t(x_t)p(x_{t+1}|x_t)}{\beta_{t+1}(x_{t+1})}. \quad (5)$$

Equation (5) indicates that $\gamma_t$ can be recursively computed based on $\alpha_t$’s and $\beta_t$’s for $t = n-1, n-2, \cdots, 1$ with the initialization of $\gamma_n(x_n) = \alpha_n(x_n)$. This is called the backward recursion.
In this lecture we build on previous analysis of state estimation for Hidden Markov Models to create an algorithm for estimating the posterior distribution of the states given the observations.

1 Notation

A quick summary of the notation:

1. **Random Variables (objects):** used more “loosely”, i.e. $X, Y, U, V$

2. **Alphabets:** $X, Y, U, V$

3. **Specific Realizations of Random Variables:** $x, y, u, v$

4. **Vector Notation:** For a vector $(x_1, \cdots, x_n)$, we denote the vector of values from the $i$-th component to the $j$-th component (inclusive) as $x_{ij}$, with the convention $x_i \triangleq x_{i1}$.

5. **Markov Chains:** We use the standard Markov Chain notation $X - Y - Z$ to denote that $X, Y, Z$ form a Markov triplet.

   For discrete random variable (object), we write $p(x)$ for the pmf $P(X = x)$, and similarly $p(x, y)$ for $p_{X,Y}(x, y)$ and $p(y|x)$ for $p_{Y|X}(y|x)$, etc.

2 Recap of HMP

For a set of unknown states $\{X_n\}_{n \geq 1}$ of a *Markov Process* and noisy observations $\{Y_n\}_{n \geq 1}$ on those states taken through some memoryless channel, we can express the joint distribution of the states and outputs:

$$ p(x^n, y^n) = \prod_{t=1}^{n} p(x_t|x_{t-1}) \prod_{t=1}^{n} p(y_t|x_t) $$ (1)

where $x_0$ is taken to be some fixed arbitrary value (e.g., 0). Additionally we have the following principles for this setting:

$$ (X^{t-1}, Y^t) - X_t - (X_{t+1}^n, Y_{t+1}^n) \quad (a) $$

$$ (X^{t-1}, Y^{t-1}) - X_t - (X_{t+1}^n, Y_t^n) \quad (b) $$

$$ X^{t-1} - (X_t, Y^{t-1}) - (X_{t+1}^n, Y_t^n) \quad (c) $$

In the last lecture, through the forward-backward recursion we have the following algorithms for computing various posteriors:

$$ \alpha_t(x_t) = p_{X_t|Y^t}(x_t|y^t) = \frac{\beta_t(x_t)p_{Y_t|X_t}(y_t|x_t)}{\sum_{x_t} \beta_t(x_t)p_{Y_t|X_t}(y_t|x_t)} \quad \text{(Measurement Update)} $$

$$ \beta_{t+1}(x_{t+1}) = p_{X_{t+1}|Y^{t+1}}(x_{t+1}|y^{t+1}) = \sum_{x_t} \alpha_t(x_t)p_{X_{t+1}|X_t}(x_{t+1}|x_t) \quad \text{(Time Update)} $$

$$ \gamma_t(x_t) = p_{X_t|Y^n}(x_t|y^n) = \sum_{x_{t+1}} \gamma_{t+1}(x_{t+1}) \frac{\alpha_t(x_t)p_{X_{t+1}|X_t}(x_{t+1}|x_t)}{\beta_{t+1}(x_{t+1})} \quad \text{(Backward Recursion)} $$
3 Factor Graphs

Let us derive a graphical representation of the joint distribution expressed in (1) as follows:

1. Create nodes on the graph representing each random variable: \(X_1, ..., X_n, Y_1, ..., Y_n\).
2. For each pairing of nodes, create an edge if there is some factor in 1 which contains both variables.

This form of graph is called a **Factor Graph**.

**Example:** Figure 1 shows the factor graph for the hidden Markov process in our setting.

![Factor Graph](image)

**Figure 1:** The factor graph for our hidden Markov process

For any partitioning of the graph into three node sets \(S_1, S_2, S_3\) such that any path from \(S_1\) to \(S_3\) passes some node in \(S_2\), these sets form a Markov process \(S_1 - S_2 - S_3\) (to see this, simply express the joint pmf into factors). Hence, for any \(s_i \in S_i, i = 1, 2, 3\), we have

\[
p(\mathbf{s}_1, \mathbf{s}_3 | \mathbf{s}_2) = \phi_1(\mathbf{s}_1, \mathbf{s}_2)\phi_2(\mathbf{s}_2, \mathbf{s}_3)
\]

for some functions \(\phi_1, \phi_2\) which consist of the factors which were used to form the factor graph. Using the factor graph, it is quite easy to establish principles (a)-(c) via graphs. For example, principles (a) and (b) can be proved via Figure 2 and 3 shown below.

![Factor Graph](image)

**Figure 2:** The factor graph partition for (a)

![Factor Graph](image)

**Figure 3:** The factor graph partition for (b)
We can also construct an additional principle from the factor graph partitioning presented in Figure 4:

\[ X_{t-1} - (X_t, Y^n) - X_{t+1}^n \] (d)

![Figure 4: The graphical proof of (d)]

### 4 General Forward–Backward Recursion: Module Functions

Let us consider a generic discrete random variable \( U \sim p_U \) passed through a memoryless channel characterized by \( p_{V|U} \) to get a discrete random variable \( V \). By Bayes rule we know

\[
p_{U|V}(u|v) = \frac{p_U(u)p_{V|U}(v|u)}{\sum_{\tilde{u}} p_U(\tilde{u})p_{V|U}(v|\tilde{u})} \tag{3}
\]

If we create the probability vector of \( p_{U|V} \) for each value of \( \mathcal{U} \), we can define a vector function \( F \) s.t.

\[
\{p_{U|V}(u|v)\}_u = \left\{ \frac{p_U(u)p_{V|U}(v|u)}{\sum_{\tilde{u}} p_U(\tilde{u})p_{V|U}(v|\tilde{u})} \right\}_u \triangleq F(p_U, p_{V|U}, v) \tag{4}
\]

Additionally, overloading the notation of \( F \), we can define the matrix function \( F \) as

\[
F(p_U, p_{V|U}) = \{ F(p_U, p_{V|U}, v) \}_v \tag{5}
\]

This function \( F \) acts as a module function for calculating the posterior distribution of a random variable given a prior distribution and a channel distribution.

In addition to \( F \), we can write a module function for calculating the marginal distribution given a prior distribution on the input and a channel distribution. We label this vector function \( G \) and define it as follows:

\[
p_V = \{ p_V(v) \}_v = \left\{ \sum_{\tilde{u}} p_U(\tilde{u})p_{V|U}(v|\tilde{u}) \right\}_v \triangleq G(p_U, p_{V|U}) \tag{6}
\]

With these module functions in mind, we can express our recursion functions as follows

\[
\alpha_t = F(\beta_t, p_{Y_t|X_t}, y_t) \quad \text{(Measurement Update)}
\]
\[
\beta_{t+1} = G(\alpha_t, p_{X_{t+1}|X_t}) \quad \text{(Time Update)}
\]
\[
\gamma_t = G(\gamma_{t+1}, F(\alpha_t, p_{X_{t+1}|X_t})) \quad \text{(Backward Recursion)}
\]
5 State Estimation: the Viterbi Algorithm

Now we are ready to derive the joint posterior distribution \( p(x^n|y^n) \):

\[
p(x^n|y^n) = p(x_n|y^n) \prod_{t=1}^{n-1} p(x_t|x_{t+1}, y^n) \\
= p(x_n|y^n) \prod_{t=1}^{n-1} p(x_t|x_{t+1}, y^n) \quad \text{(By principle (d))} \\
= p(x_n|y^n) \prod_{t=1}^{n-1} p(x_t|x_{t+1}, y^n) \quad \text{(By principle (c))} \\
= p(x_n|y^n) \prod_{t=1}^{n-1} \frac{p(x_t|y^t)p(x_{t+1}|x_t, y^t)}{p(x_{t+1}|y^t)} \\
= \gamma_n(x_n) \prod_{t=1}^{n-1} \frac{\alpha_t(x_t)p(x_{t+1}|x_t)}{\beta_{t+1}(x_{t+1})} \quad \text{(By principle (a))}
\]

Write

\[
\ln p(x^n|y^n) = \sum_{t=1}^{n} g_t(x_t, x_{t+1})
\]

with

\[
g_t(x_t, x_{t+1}) \triangleq \begin{cases} 
\ln \frac{\alpha_t(x_t)p(x_{t+1}|x_t)}{\beta_{t+1}(x_{t+1})} & t = 1, \ldots, n-1 \\
\ln \gamma_n(x_n) & t = n
\end{cases}
\]

we can solve the MAP estimator \( \hat{x}_{\text{MAP}} \) with the help of the following definition:

**Definition 1.** For \( 1 \leq k \leq n \), Let

\[
M_k(x_k) := \max_{x_{k+1}} \sum_{t=k}^{n} g_t(x_t, x_{t+1}).
\]

It is straightforward from definition that \( M_1(\hat{x}_{\text{MAP}}) = \ln p(\hat{x}_{\text{MAP}}|y^n) = \max_{x^n} p(x^n|y^n) \) and

\[
M_k(x_k) = \max_{x_{k+1}} \max_{x_{k+2}} \left( g_k(x_k, x_{k+1}) + \sum_{t=k+1}^{n} g_t(x_t, x_{t+1}) \right)
\]

\[
= \max_{x_{k+1}} \left( g_k(x_k, x_{k+1}) + \max_{x_{k+2}} \sum_{t=k+1}^{n} g_t(x_t, x_{t+1}) \right)
\]

\[
= \max_{x_{k+1}} \left( g_k(x_k, x_{k+1}) + M_{k+1}(x_{k+1}) \right).
\]

Since \( M_n(x_n) = g(x_n, x_{n+1}) = \ln \gamma_n(x_n) \) only depends on one term \( x_n \), we may start from \( n \) and use the previous recursive formula to obtain \( M_1(x_1) \). This is called the Viterbi Algorithm.

The Bellman Equations, referred to in the communication setting as the Viterbi Algorithm, is an application of a technique called dynamic programming. Using the recursive equations derived in the previous section, we can express the Viterbi Algorithm as follows.
1: function VITERBI
2:   $M_n(x_n) \leftarrow \ln \gamma_n(x_n)$  \hspace{1cm} \triangleright \text{Initialization of log-likelihood}
3:   $\hat{x}_\text{MAP}(x_n) \leftarrow \emptyset$ \hspace{1cm} \triangleright \text{Initialization of the MAP estimator}
4:   \textbf{for} $k = n-1, \cdots, 1$ \textbf{do}
5:     $M_k(x_k) \leftarrow \max_{x_{k+1}} (g_k(x_k, x_{k+1}) + M_{k+1}(x_{k+1}))$  \hspace{1cm} \triangleright \text{Maximum of log-likelihood}
6:     $\hat{x}_{k+1}(x_k) \leftarrow \arg \max_{x_{k+1}} (g_k(x_k, x_{k+1}) + M_{k+1}(x_{k+1}))$
7:     $\hat{x}_\text{MAP}(x_k) \leftarrow [\hat{x}_{k+1}(x_k), \hat{x}_\text{MAP}(\hat{x}_{k+1}(x_k))]$  \hspace{1cm} \triangleright \text{Maximizing sequence with leading term } x_k
8: \textbf{end for}
9:   $M \leftarrow \max_{x_1} M_1(x_1)$  \hspace{1cm} \triangleright \text{Maximum of overall log-likelihood}
10:  $\hat{x}_1 \leftarrow \arg \max_{x_1} M_1(x_1)$
11:  $\hat{x}_\text{MAP} \leftarrow [\hat{x}_1, \hat{x}_\text{MAP}(\hat{x}_1)]$  \hspace{1cm} \triangleright \text{Overall maximizing sequence}
12: \textbf{end function}
In this lecture, we introduce new methods for solving Hidden Markov Processes in cases beyond discrete alphabets. This requires us to learn first about estimating pdfs based on samples from a different distribution. At the end, we introduce the Bayesian inference briefly.

1 Recap: Inference on Hidden Markov Processes (HMPs)

1.1 Setting
A quick recap on the setting of a HMP:

1. \( \{X_n\}_{n \geq 1} \) is a Markov process, called “the state process”.

2. \( \{Y_n\}_{n \geq 1} \) is “the observation process” where \( Y_t \) is the output of \( X_t \) sent through a “memoryless” channel characterized by the distribution \( P_{Y|X} \).

Alternatively, we showed in HW1 that the above is totally equivalent to the following:

1. \( \{X_n\}_{n \geq 1} \) is “the state process” defined by
   \[
   X_t = f_t(X_{t-1}, W_t)
   \]
   where \( \{W_n\}_{n \geq 1} \) is an independent process, i.e. each \( W_t \) is independent of all the others.

2. \( \{Y_n\}_{n \geq 1} \) is “the observation process” related to \( \{X_n\}_{n \geq 1} \) by
   \[
   Y_t = g_t(X_t, N_t)
   \]
   where \( \{N_n\}_{n \geq 1} \) is an independent process.

3. The processes \( \{N_n\}_{n \geq 1} \) and \( \{W_n\}_{n \geq 1} \) are independent of each other.

1.2 Goal
Since the state process, \( \{X_n\}_{n \geq 1} \) is “hidden” from us, we only get to observe \( \{Y_n\}_{n \geq 1} \) and wish to find a way of estimating \( \{X_n\}_{n \geq 1} \) based on \( \{Y_n\}_{n \geq 1} \). To do this, we define the forward recursions for causal estimation:

\[
\alpha_t(x_t) = F(\beta_t, P_{Y_t|X_t}, y_t) \quad (1)
\]
\[
\beta_{t+1}(x_{t+1}) = G(\alpha_t, P_{X_{t+1}|X_t}) \quad (2)
\]

where \( F \) and \( G \) are the operators defined in lecture 4.

1.3 Challenges
For general alphabets \( X, Y \), computing the forward recursion is a very difficult problem. However, we know efficient algorithms that can be applied in the following situations:

1. All random variables are discrete with a finite alphabet. We can then solve these equations by brute force in time proportional to the alphabet sizes.
2. For $t \geq 1$, $f_t$ and $g_t$ are linear and $N_t, W_t, X_t$ are Gaussian. The solution is the “Kalman Filter” algorithm that we derive in Homework 2.

Beyond these special situations, there are many heuristic algorithms for this computation:

1. We can compute the forward recursions approximately by quantizing the alphabets. This leads to a tradeoff between model accuracy and computation cost.
2. This lecture will focus on Particle Filtering, which is a way of estimating the forward recursions with an adaptive quantization.

## 2 Importance Sampling

Before we can learn about Particle Filtering, we first need to discuss importance sampling$^1$.

### 2.1 Simple Setting

Let $X_1, \cdots, X_N$ be i.i.d. random variables with density $f$. Now we wish to approximate $f$ by a probability mass function, $\hat{f}_N$, using our data.

The idea: we can approximate $f$ by taking

$$
\hat{f}_N = \frac{1}{N} \sum_{i=1}^{N} \delta(x - X_i)
$$

where $\delta(x)$ is a Dirac-delta distribution.

**Claim:** $\hat{f}_N$ is a good estimate of $f$.

Here we define the ”goodness” of our estimate by looking at

$$
\mathbb{E}_{X \sim \hat{f}_N}[g(X)]
$$

and seeing how close it is to

$$
\mathbb{E}_{X \sim f}[g(X)]
$$

for any function $g$ with $\mathbb{E}|g(X)| < \infty$. We show that by this definition, (3) is a good estimate for $f$.

**Proof:**

$$
\mathbb{E}_{X \sim \hat{f}_N}[g(X)] = \int_{-\infty}^{\infty} \hat{f}_N(x)g(x)dx
$$

$$
= \int_{-\infty}^{\infty} \frac{1}{N} \sum_{i=1}^{N} \delta(x - X_i)g(x)dx
$$

$$
= \frac{1}{N} \sum_{i=1}^{N} \int_{-\infty}^{\infty} \delta(x - X_i)g(x)dx
$$

$$
= \frac{1}{N} \sum_{i=1}^{N} g(X_i)
$$

\[\text{See section 2.12.4 of the Spring 2013 lecture notes for further reference.}\]
By the law of large numbers,
\[
\frac{1}{N} \sum_{i=1}^{N} g(X_i) \xrightarrow{a.s.} \mathbb{E}_{X \sim f}[g(X)]
\] (10)
as \( N \to \infty \). Hence,
\[
\mathbb{E}_{X \sim f_N}[g(X)] \to \mathbb{E}_{X \sim f}[g(X)]
\] (11)
so \( f_N \) is indeed a "good" estimate of \( f \).

### 2.2 General Setting

Suppose now that \( X_1, \cdots, X_N \) i.i.d. with common density \( q \), but we wish to estimate a different pdf \( f \).

**Theorem**: The following estimator
\[
\hat{f}_N(x) = \sum_{i=1}^{N} w_i \delta(x - X_i)
\] (12)
with weights
\[
w_i = \frac{f(X_i)}{q(X_i)} \sum_{j=1}^{N} \frac{f(X_j)}{q(X_j)}
\] (13)
is a good estimator of \( f \).

**Proof**: It’s easy to see
\[
\mathbb{E}_{X \sim \hat{f}_N}[g(X)] = \sum_{i=1}^{N} w_i g(X_i)
\] (14)
\[
= \sum_{i=1}^{N} \frac{f(X_i)}{q(X_i)} g(X_i)
\] (15)
\[
= \sum_{i=1}^{N} \frac{f(X_i)}{q(X_i)} \frac{g(X_i)}{q(X_i)}
\] (16)
\[
= \frac{1}{N} \sum_{i=1}^{N} \frac{f(X_i)}{q(X_i)} g(X_i)
\] (17)

As \( N \to \infty \), the above expression converges to
\[
\mathbb{E}\left[\frac{f(X)}{q(X)} g(X)\right] \to \int_{-\infty}^{\infty} \frac{f(x)}{q(x)} g(x) q(x) dx \]
\[
= \frac{\int_{-\infty}^{\infty} f(x) q(x) g(x) dx}{\int_{-\infty}^{\infty} f(x) q(x) dx}
\] (18)
\[
= \frac{\int_{-\infty}^{\infty} f(x) g(x) dx}{\int_{-\infty}^{\infty} f(x) dx}
\] (19)
\[
= \int_{-\infty}^{\infty} f(x) g(x) dx
\] (20)
\[
= \mathbb{E}_{X \sim f}[g(X)].
\] (21)
Hence,
\[ \mathbb{E}_{X \sim \hat{f}_N}[g(X)] \to \mathbb{E}_{X \sim f}[g(X)] \] (22)
as \( N \to \infty \).

### 2.3 Implications

The above theorem is useful in situations when we don’t know \( f \) and cannot draw samples relying on \( X_i \sim f \), but we know that \( f(x) = c \cdot h(x) \) for some constant \( c \) (in many practical cases it is computationally prohibitive to obtain the normalization constant \( c = (\int h(x) dx)^{-1} \)). Then

\[
\hat{f}_N(X_i) = \frac{f(X_i)}{\sum_{j=1}^{N} f(X_j)}
\]

(23)

\[
= \frac{c \cdot h(X_i)}{\sum_{j=1}^{N} c \cdot h(X_j)} \quad \text{for } i = 1, \ldots, N
\]

(24)

\[
= \frac{h(X_i)}{\sum_{j=1}^{N} h(X_j)}
\]

(25)

Using these \( w_i \)'s, we see that we can approximate \( f \) using samples drawn from a different distribution and \( h \).

### 2.4 Application

Let \( X \sim f_X \) and \( Y \) be the observation of \( X \) through a memoryless channel characterized by \( f_{Y|X} \). For MAP reconstruction of \( X \), we must compute

\[
f_{X|Y}(x|y) = \frac{f_X(x) f_{Y|X}(y|x)}{\int_{-\infty}^{\infty} f_X(\tilde{x}) f_{Y|X}(y|\tilde{x}) d\tilde{x}}
\]

(26)

In general, this computation may be hard due to the integral appearing on the bottom. But the denominator is just a constant. Hence, suppose we have \( X_i \) i.i.d. \( \sim f_X \) for \( 1 \leq i \leq N \). By the previous theorem, we can take

\[
\hat{f}_{X|y} = \sum_{i=1}^{N} w_i(y) \delta(x - X_i)
\]

(27)
as an approximation, where

\[
w_i(y) = \frac{f_{X|Y}(X_i)}{f_{X}(X_i)} \frac{f_{Y|X}(y|X_i)}{f_{Y}(y|X_i)}
\]

(28)

\[
= \frac{f_X(X_i) f_{Y|X}(y|X_i) f_{X}(X_i)}{f_X(X_i) f_{Y}(y|X_i) f_{X}(X_i)}
\]

(29)

\[
= \frac{f_{Y|X}(y|X_i)}{f_{X}(X_i)}
\]

(30)

Hence, using the samples \( X_1, X_2, \ldots, X_N \) we can build a good approximation of \( f_{X|Y}(\cdot|y) \) for fixed \( y \) using the above procedure.
Let us denote this approximation as
\[
\hat{f}_N(X_1, X_2, ..., X_N, f_X, f_{Y|X}, y) = \sum_{i=1}^{N} w_i \delta(x - X_i)
\] (31)

3 Particle Filtering

Particle Filtering is a way to approximate the forward recursions of a HMM using importance sampling to approximate the generated distributions rather than computing the true solutions.

**Algorithm 1:** Particle Filter

This algorithm requires some art in use. It is not well understood how errors propagate due to these successive approximations. In practice, the parameters need to be periodically reset. Choosing these hyperparameters is an application-specific task.

4 Inference under logarithmic loss

Recall the Bayesian decision theory and let us have the following:

- $X \in \mathcal{X}$ is something we want to infer;
- $X \sim p_x$ (assume that $X$ is discrete);
- $\hat{x} \in \hat{X}$ is something we wish to reconstruct;
- $\Lambda: \mathcal{X} \times \hat{X} \to \mathbb{R}$ is our loss function.

First we assume that we do not have any observation, and define the Bayesian response as
\[
U(p_x) = \min_{\hat{x}} \mathbb{E}_X \Lambda(X, \hat{x}) = \min_{\hat{x}} \sum_x p_x(s) \Lambda(x, \hat{x})
\] (32)

and
\[
\hat{X}_{\text{Bayes}}(p_X) = \arg \min_{\hat{x}} \mathbb{E}_X \Lambda(X, \hat{x}).
\] (33)

**Example:** Let $\mathcal{X} = \hat{X} = \mathbb{R}$, $\Lambda(x, \hat{x}) = (x - \hat{x})^2$. Then
\[
U(p_X) = \min_{\hat{x}} \mathbb{E}_{X \sim p_X} (X - \hat{x})^2 = \text{Var}(X)
\] (34)
\[
\hat{X}_{\text{Bayes}}(p_X) = \mathbb{E}[X].
\] (35)
Consider now the case we have some observation (side information) $Y$, where our reconstruction $\hat{X}$ can be a function of $Y$. Then

$$\mathbb{E}\Lambda(X, \hat{X}(Y)) = \mathbb{E}[\mathbb{E}[\Lambda(X, \hat{X}(y))|Y = y]]$$
$$= \sum_y \mathbb{E}[\Lambda(X, \hat{X}(y))|Y = y]p_Y(y)$$
$$= \sum_y p_Y(y) \sum_x p_{X|Y=y}(x)\Lambda(x, \hat{X}(y))$$

This implies

$$U(p_X) = \min_{\hat{X}(\cdot)} \mathbb{E}\Lambda(X, \hat{X}(Y))$$
$$= \min_{\hat{X}(\cdot)} \sum_y p_Y(y) \sum_x p_{X|Y=y}(x)\Lambda(x, \hat{X}(y))$$
$$= \sum_y p_Y(y) U(p_{X|Y=y})$$

and thus

$$\hat{X}_{opt}(y) = \hat{X}_{Bayes}(p_{X|Y=y})$$
In this lecture\(^1\), we introduce the topic of Bayesian Decision Theory and define the concept of loss function for assessing the accuracy of the estimation. The log loss function is presented as an important special case.

## 1 Notation

A quick summary of the notation:

1. **Random Variables (objects):** used more “loosely”, i.e., \(X, Y\)
2. **PMF (PDF):** used more “loosely”, i.e. \(P_X, P_Y\)
3. **Alphabets:** \(\mathcal{X}, \mathcal{X}^\ast\)
4. **Estimate of \(X\):** \(\hat{x}\)
5. **Specific Values:** \(x, y\)
6. **Loss Function:** \(\Lambda\)
7. **Markov Triplet:** \(X \rightarrow Y \rightarrow Z\) notation corresponds to a Markov triplet
8. **Logarithm:** log notation denotes \(\log_2\), unless otherwise stated

For discrete random variable (object), \(U\) has p.m.f: \(P_U(u) = P(U = u)\). Often, we’ll just write \(p(u)\). Similarly: \(p(x, y)\) for \(P_{X,Y}(x, y)\) and \(p(y|x)\) for \(P_{Y|X}(y|x)\), etc.

## 2 Bayesian Decision Theory

Let \(X \in \mathcal{X}, X \sim P_X, \hat{x} \in \mathcal{X}^\ast, \Lambda : \mathcal{X} \times \mathcal{X}^\ast \rightarrow \mathbb{R}\).

**Definition 1** (Bayes Envelope).

\[
U(P_X) \triangleq \min_{\hat{x}} \mathbb{E}[\Lambda(X, \hat{x})] = \min_{\hat{x}} \sum_x P_X(x) \Lambda(X, \hat{x})
\]

**Definition 2** (Bayes Response).

\[
\hat{X}_{Bayes}(P_X) \triangleq \arg \min_{\hat{x}} \mathbb{E}[\Lambda(X, \hat{x})]
\]

Essentially, given a prior distribution on \(X\), we are trying to achieve the best estimate through minimizing the expected value of some “loss function” that we have defined. If in addition to the prior distribution of \(X\), we are also given the observation \(Y\), the associated *Bayes Envelope* becomes:

\[
\min_{\hat{X}} \mathbb{E}[\Lambda(X, \hat{X}(Y))] = \sum_y U(P_{X|Y}(y))P_Y(y) = \mathbb{E}[U(P_{X|Y})]
\]

In this case, the *Bayes Response* takes the form:

\[
\hat{X}(y) = \hat{X}_{Bayes}(P_{X|Y=y})
\]

\(^1\)Reading: Lecture 3 of the EE378A Spring 2016 notes
3 Bayesian Inference Under Logarithmic Loss

In this section, we will be using the logarithm function as the loss function.

**Definition 3** (Logarithmic Loss Function).

\[
\Lambda(x, Q) \triangleq \log \frac{1}{Q(x)}
\]  

where \( x \) is the realization of \( X \), \( Q \) is the estimator of the pmf of \( X \).

Using Definition 3, the expected value of the log loss function could be expressed as:

\[
E[\Lambda(X, Q)] = \sum_x P_X(x) \Lambda(x, Q) = \sum_x P_X(x) \log \frac{1}{Q(x)} = \sum_x P_X(x) \log \left( \frac{P_X(x)}{Q(x)} \right)
\]  

**Definition 4** (Entropy).

\[
H(X) \triangleq \sum_x P_X(x) \log \frac{1}{P_X(x)}
\]

**Definition 5** (Relative Entropy, or the Kullback–Leibler Divergence).

\[
D(P_X \| Q) \triangleq \sum_x P_X(x) \log \frac{P_X(x)}{Q(x)}
\]

Using Equation (6,7) and Definitions (4,5):

\[
E[\Lambda(X, Q)] = H(X) + D(P_X \| Q)
\]

**Claim:** \( D(P_X \| Q) \geq 0 \), and \( D(P_X \| Q) = 0 \iff Q = P_X \).

Before attempting the proof, we provide Jensen’s inequality:

**Jensen’s Inequality:** Let \( Q \) denote a *concave* function, and \( X \) be any random variable. Then

\[
E[Q(X)] \leq Q(E[X])
\]

Further, if \( Q \) is strictly concave: \( E[Q(X)] = Q(E[X]) \iff X \) is a constant. In particular, \( Q(x) = \log(x) \) is a concave function, thus for a random variable \( X \geq 0 \):

\[
E[\log X] \leq \log E[X].
\]

**Proof of the Claim:**

\[
D(P_X \| Q) = E[\log \frac{P_X(x)}{Q(x)}] = -E[\log \frac{Q(x)}{P_X(x)}] \geq -\log E[\frac{Q(x)}{P_X(x)}] = -\log \sum_x P_X(x) \frac{Q(x)}{P_X(x)} = 0
\]

The last equality follows from \( \sum_x Q(x) = 1 \), since \( Q(x) \) is a pmf and thus sums to one.

Hence, it follows from the claim that

\[
\min_Q E[\Lambda(X, Q)] = H(X) = U(P_X)
\]
where the minimizer is \( Q = P_X \). Therefore:

\[
\hat{X}_{\text{Bayes}}(P_X) = P_X.
\] (15)

Introducing the notation \( H(X) \iff H(P_X) \) and making use of equation (3):

\[
\min_{\hat{X}(\cdot)} E[\Lambda(X, \hat{X}(Y))] = E[U(P_X|Y)] = \sum_y H(P_X|Y)P_Y(y)
\] (16)

**Definition 6** (Conditional Entropy).

\[
H(X|Y) \triangleq \sum_y H(P_X|Y)P_Y(y)
\] (17)

**Definition 7** (Mutual Information).

\[
I(X;Y) \triangleq H(X) - H(X|Y)
\] (18)

**Definition 8** (Conditional Relative Entropy).

\[
D(P_X|Y||Q_X|Y, P_Y) \triangleq \sum_y D(P_X|Y||Q_X|Y)P_Y(y)
\] (19)

**Definition 9** (Conditional Mutual Information).

\[
I(X, Y|Z) \triangleq H(X|Z) - H(X|Y, Z)
\] (20)

**Exercises:**

(a) \( H(X_1, X_2) = H(X_1) + H(X_2|X_1) \)

(b) \( I(X;Y) = H(Y) - H(Y|X) \)

(c) \( I(X_1, X_2; Y) = I(X_1; Y) + I(X_2; Y|X_1) \)

(d) \( D(P_{X_1, X_2}||Q_{X_1, X_2}, P_Y) = D(P_{X_1}||Q_{X_1}) + D(P_{X_2|X_1}||Q_{X_2|X_1}|P_{X_1}) \)

### 4 Justification of Log Loss as the Right Loss Function

For general loss function \( \Lambda \):

**Definition 10** (Coherent Dependence Function).

\[
C(\Lambda, P_X, Y) \triangleq \inf_{\hat{x}} E[\Lambda(X, \hat{x})] - \inf_{\hat{X}(\cdot)} E[\Lambda(X, \hat{X}(Y))]
\] (21)

The first term corresponds to the minimum expected error (depending on the choice of loss function) in estimating \( X \) based on only the prior distribution of \( X \). The second term corresponds to the minimum expected error in estimating \( X \) by also taking the observation \( Y \) into account. Therefore, the coherent dependence function gives a measure on how much the estimate improves given observation \( Y \), compared to estimating \( X \) based on only the prior distribution of \( X \).

Two examples are provided below to make this concept more clear:

1. \( \Lambda(x, \hat{x}) = (x - \hat{x})^2 \Rightarrow C(\Lambda, P_X, Y) = \text{Var}(X) - E[\text{Var}(X|Y)] \)
2. \( \Lambda(x, Q) = \log \frac{1}{Q(x)} \Rightarrow C(\Lambda, P_X, Y) = I(X, Y) \)

Proving the expressions below left as exercise:

(a) \( C(\Lambda, P_X, Y) \geq 0 \), and \( C(\Lambda, P_X, Y) = 0 \iff X \) and \( Y \) are independent
(b) $C(\Lambda, P_{X,Y}) \geq C(\Lambda, P_{X,Z})$ when $X - Y - Z$

**Natural Requirement:** If $T = T(X)$ is a function of $X$, then $C(\Lambda, P_{T,Y}) \leq C(\Lambda, P_{X,Y})$.

This requirement basically states that a function of $X$ cannot carry more information about $X$ compared to $X$ itself.

**Definition 11** (Sufficient Statistic). $T = T(X)$ is a sufficient statistic of $X$ for $Y$ if $X - T - Y$.

**Modest Requirement:** If $T$ is a sufficient statistic of $X$ for $Y$, then $C(\Lambda, P_{T,Y}) \leq C(\Lambda, P_{X,Y})$

**Theorem:** Assume $|\mathcal{X}| \geq 3$, the only $C(\Lambda, P_{X,Y})$ satisfying the modest requirement is $C(\Lambda, P_{X,Y}) = I(X; Y)$ (up to a multiplicative factor).

Since it was shown that $C(\Lambda, P_{X,Y}) = I(X; Y)$ if (and only if) the logarithmic loss function is used ($\Lambda(x, Q) = \log \frac{1}{Q(x)}$), it follows that the logarithmic loss function is the only loss function which satisfies the modest requirement.
In this lecture we are going to study prediction under Bayesian setting with general and log loss. We will also quickly start the subject of prediction of individual sequences.

1 Notation

A quick summary of the notation:

1. Random Variables (objects): used more “loosely”, i.e. $X, Y, U, V$;
2. Alphabets: $\mathcal{X}, \hat{\mathcal{X}}$;
3. Sequence of Random Variables: $X_1, X_2, \cdots, X_t$;
4. Loss function: $\Lambda(x, \hat{x})$;
5. Estimate of $X$: $\hat{x}$;
6. Set of distributions on Alphabet $\mathcal{X}$: $M(\mathcal{X})$.

For discrete random variable (object), $U$ has p.m.f: $P_U(\cdot) \triangleq P(U = \cdot)$. Often, we’ll just write $p(\cdot)$.

Similarly: $p(x, y)$ for $P_{X,Y}(x, y)$ and $p(y|x)$ for $P_{Y|X}(y|x)$, etc.

2 Prediction

Consider $X_1, X_2, \cdots, X_n \in \mathcal{X}$.

**Definition 1** (Predictor). A predictor $F$ is a sequence of functions $F = (F_t)_{t \geq 1}$ with $F_t : \mathcal{X}^{t-1} \to \hat{\mathcal{X}}$.

**Definition 2** (Prediction Loss). Given a loss function $\Lambda : \mathcal{X} \times \hat{\mathcal{X}} \to \mathbb{R}_+$, the prediction loss incurred by the predictor $F$ is defined as

$$L_F(X^n) = \frac{1}{n} \sum_{t=1}^{n} \Lambda(X_t, F_t(X_{t-1})).$$  \hspace{1cm} (1)

2.1 Prediction under Bayesian Setting

In Bayesian setting, we assume $X_1, X_2, \cdots$ are governed jointly by some underlying law $P$. Then we have the following prediction loss.

**Definition 3** (Prediction Loss Under Bayesian Setting).

$$\min_F \mathbb{E}_P[L_F(X^n)] = \frac{1}{n} \sum_{t=1}^{n} \min_{F_t} \mathbb{E}_P[\Lambda(X_t, F_t(X_{t-1}))] = \frac{1}{n} \sum_{t=1}^{n} \mathbb{E}_P[U(P_{X_t|X_{t-1}})]$$  \hspace{1cm} (2)

where $U(\cdot)$ is the Bayes envelope we have defined in the previous lecture.
As we can see from (2), our initial objective was to minimize expected prediction loss with respect to the whole sequence of functions. However, since we can interchange summation and expectation, we can treat the overall predictor $F$ as the composition of the sequential optimal predictors $F_t$. Finally, we note the overall loss is just the sum of expectations of Bayes envelopes.

**Theorem 4 (Miss-match Loss).** The additional prediction loss incurred by doing prediction based on law $Q$ instead of $P$ is upper bounded by some form of relative entropy:

$$ \mathbb{E}_P[L_{F_P}(X^n) - L_{F_Q}(X^n)] \leq \Lambda_{\text{max}} \cdot \sqrt{\frac{2}{n} D(P_X\|Q_X)}, \quad (3) $$

where $\Lambda_{\text{max}} = \sup_{x, \hat{x}} \Lambda(x, \hat{x})$.

Now we are going to introduce Lemma 5 which will be used to prove Theorem 4.

**Lemma 5.**

$$ \mathbb{E}_P[\Lambda(X, \hat{X}_{\text{Bayes}}(Q)) - \Lambda(X, \hat{X}_{\text{Bayes}}(P))] \leq \Lambda_{\text{max}} \|P - Q\|_1 \leq \Lambda_{\text{max}} \sqrt{2D(P\|Q)} \quad (4) $$

**Proof**

$$ \mathbb{E}_P[\Lambda(X, \hat{X}_{\text{Bayes}}(Q)) - \Lambda(X, \hat{X}_{\text{Bayes}}(P))] \leq \mathbb{E}_P[\Lambda(X, \hat{X}_{\text{Bayes}}(Q)) - \Lambda(X, \hat{X}_{\text{Bayes}}(P))] + \mathbb{E}_Q[-\Lambda(X, \hat{X}_{\text{Bayes}}(Q)) + \Lambda(X, \hat{X}_{\text{Bayes}}(P))] \quad (5) $$

$$ = \sum_x (P(x) - Q(x)) (\Lambda(x, \hat{X}_{\text{Bayes}}(Q)) - \Lambda(x, \hat{X}_{\text{Bayes}}(P))) \quad (6) $$

$$ \leq \sum_x |P(x) - Q(x)| \Lambda_{\text{max}} \quad (7) $$

$$ = \Lambda_{\text{max}} \|P - Q\|_1 \quad (8) $$

$$ \leq \Lambda_{\text{max}} \sqrt{2D(P\|Q)} \quad (9) $$

where in the last step we have used Pinsker’s inequality $D(P\|Q) \geq \frac{1}{2} \|P - Q\|_1^2$. \hfill \square

Now we prove Theorem 4 as follows:

$$ \mathbb{E}_P[L_{F_Q}(X^n) - L_{F_P}(X^n)] \quad (10) $$

$$ = \mathbb{E}_P[\sum_{t=1}^{n} (\Lambda(X_t, F_t(Q(X^{t-1}))) - \Lambda(X_t, F_t(P(X^{t-1}))))] \quad (11) $$

$$ = \frac{1}{n} \sum_{t=1}^{n} \mathbb{E}_P[\Lambda(X_t, F_t(Q(X^{t-1}))) - \Lambda(X_t, F_t(P(X^{t-1}))))|X^{t-1}] \quad \text{(Iterated Expectation)} \quad (12) $$

$$ \leq \frac{1}{n} \sum_{t=1}^{n} \mathbb{E}_P[\Lambda_{\text{max}} \sqrt{2D(P_{X_t|X^{t-1}}\|Q_{X_t|X^{t-1}}|P_{X^{t-1}}})] \quad \text{(Using Lemma 5)} \quad (13) $$

$$ \leq \Lambda_{\text{max}} \sqrt{\frac{2}{n} \mathbb{E}_P[D(P_{X_t|X^{t-1}}\|Q_{X_t|X^{t-1}}|P_{X^{t-1}})]} \quad \text{(Jensen’s inequality)} \quad (14) $$

$$ = \Lambda_{\text{max}} \sqrt{\frac{2}{n} D(P_X\|Q_X)} \quad \text{(Using chain rule for relative entropy)} \quad (15) $$

as desired.
3 Prediction Under Logarithmic Loss

In this section, we specialize our setup to the logarithmic loss while changing our predictor set into a set of distributions. So here we have $\hat{X} = M(\mathcal{X})$ and $\Lambda(x, P) = -\log P(x)$ as the logarithmic loss. Also note that $F_i(X^{t-1}) \in M(\mathcal{X})$.

So we can represent our predictor with 3 different setups as follows:

1. Obviously we can carry over the setup we worked previously and see our predictors as a set of functions $\{F_t\}_{t \geq 1}$.
2. Another approach would be to see each predictor as a sequence of conditional distributions $\{Q_{X_t|X_{t-1}}\}_{t \geq 1}$ where $F_i(X^{t-1}) = Q_{X_i|X^{t-1}}$.
3. And finally we can see our predictor as a model on the data. Then $Q$ would be the law we assigned to the sequence we observe.

Then we can immediately adapt previous results to log loss case. First one is $L_F(X^n)$:

\[
L_F(X^n) = \frac{1}{n} \sum_{t=1}^{n} \log \frac{1}{F_t(X^{t-1})[x_t]}
= \frac{1}{n} \sum_{t=1}^{n} \log \frac{1}{Q_{X_t|X^{t-1}}(x_t)}
= \frac{1}{n} \log \frac{1}{Q(X^n)} \quad \text{(Where now Q is our model about the data)}
\]

Now let’s look at the logarithmic loss under the Bayesian setting while assuming $X_1, X_2, \cdots$ sequence is governed by $P$:

\[
\mathbb{E}_P[L_F(X^n)] = \mathbb{E}_P[\frac{1}{n} \log \frac{1}{Q(X^n)}]
= \mathbb{E}_P[\frac{1}{n} \log \frac{P(X^n)}{Q(X^n)P(X^n)}]
= \frac{1}{n} [H(X^n) + D(P_{X^n}||Q_{X^n})]
\]

Then we can see that $\min_F \mathbb{E}_P[L_F(X^n)] = \frac{1}{n} H(X^n)$ and it is achieved when $F_i(X^{t-1}) = P_{X_i|X^{t-1}}$.

4 Prediction of Individual Sequences

As for the last part, we look at a different set of predictors. In this setup we are given a class of predictors $\mathcal{F}$, and we seek a predictor $F$ such that:

\[
\lim_{n \to \infty} \max_{X^n} \max_{G \in \mathcal{F}} [L_F(X^n) - \min_{G \in \mathcal{F}} L_G(X^n)] = 0
\]

4.1 Prediction of Individual Sequences under Log Loss

Now the question arises that given a class of law $\mathcal{F}$, can we find $P$ such that (23) is small for any $X^n$:

\[
\frac{1}{n} \log \frac{1}{P(X^n)} - \min_{Q \in \mathcal{F}} \frac{1}{n} \log \frac{1}{Q(X^n)} = \frac{1}{n} \log \frac{1}{P(X^n)} - \frac{1}{n} \log \frac{1}{\max_{Q \in \mathcal{F}} Q(X^n)}.
\]

This problem will be solved in the upcoming lecture.
In the last lecture we have studied the prediction problem under Bayesian setting, and in this lecture we are going to study the prediction problem under logarithmic loss in the individual sequence setting.

1 Problem setup

We first recall the problem setup: given a class of probability laws $\mathcal{F}$, we wish to find some law $P$ with a small worst-case regret

$$WCR_n(P, \mathcal{F}) \triangleq \max_{x^n} \left\{ \log \frac{1}{P(x^n)} - \min_{Q \in \mathcal{F}} \log \frac{1}{Q(x^n)} \right\}. \tag{1}$$

Then a first natural question is as follows: could we find some $P$ such that

$$WCR_n(P, \mathcal{F}) \leq n \epsilon_n \tag{2}$$

with $\epsilon_n \to 0$ as $n \to \infty$?

2 First answer when $|\mathcal{F}| < \infty$: uniform mixture

The answer to the previous question turns out to be affirmative when $|\mathcal{F}| < \infty$. Our construction of $P$ is as follows:

$$P_{\text{Unif}}(x^n) = \frac{1}{|\mathcal{F}|} \sum_{Q \in \mathcal{F}} Q(x^n). \tag{3}$$

The performance of $P_{\text{Unif}}$ is summarized in the following lemma:

**Lemma 1.** The worst-case regret of $P_{\text{Unif}}$ satisfies:

$$WCR_n(P_{\text{Unif}}, \mathcal{F}) \leq \log |\mathcal{F}|. \tag{4}$$

**Proof** Just apply $Q(x^n) \geq 0$ for any $Q \in \mathcal{F}$ and $x^n$ to conclude that

$$\log \frac{1}{P(x^n)} - \min_{Q \in \mathcal{F}} \log \frac{1}{Q(x^n)} = \log \frac{\max_{Q \in \mathcal{F}} Q(x^n)}{P(x^n)} \tag{5}$$

$$= \log |\mathcal{F}| + \log \frac{\max_{Q \in \mathcal{F}} Q(x^n)}{\sum_{Q \in \mathcal{F}} Q(x^n)} \tag{6}$$

$$\leq \log |\mathcal{F}|. \tag{7}$$

By Lemma 1, we know that we can choose $\epsilon_n = \frac{\log |\mathcal{F}|}{n} \to 0$ as $n \to \infty$, which provides an affirmative answer to our question.
3 Second answer: normalized maximum likelihood

In the previous section we know that \( \epsilon_n \to 0 \) is possible when \( |F| < \infty \), but we are still looking for some law \( P \) which gives the minimum \( \epsilon_n \), or equivalently, a minimum \( \text{WCR}_n(P,F) \). Complicated as the definition of \( \text{WCR}_n(P,F) \) suggests, surprisingly its minimum admits a closed-form expression:

**Theorem 2** (Normalized Maximum Likelihood). *The minimum worst-case regret is given by*

\[
\min_P \text{WCR}_n(P,F) = \log \sum_{x^n} \max_{Q \in F} Q(x^n).
\]  

*(8)*

This bound is attained by the normalized maximum likelihood predictor expressed as

\[
P_{\text{NML}}(x^n) = \frac{\max_{Q \in F} Q(x^n)}{\sum_{\tilde{x}^n} \max_{Q \in F} Q(\tilde{x}^n)}.
\]  

*(9)*

**Proof** Since \( \frac{P_{\text{NML}}(x^n)}{\max_{Q \in F} Q(x^n)} \) is a constant, it is clear that \( P_{\text{NML}} \) attains the desired equality. To prove the optimality, suppose \( P' \) is any other predictor. Since both \( P_{\text{NML}} \) and \( P' \) are probability distributions, there must exist some \( x^n \) such that

\[
P'(x^n) \leq P_{\text{NML}}(x^n).
\]  

*(10)*

Considering this sequence \( x^n \) gives

\[
\text{WCR}_n(P',F) \geq \log \frac{\max_{Q \in F} Q(x^n)}{P'(x^n)} \geq \log \frac{\max_{Q \in F} Q(x^n)}{P_{\text{NML}}(x^n)} = \log \sum_{x^n} \max_{Q \in F} Q(x^n)
\]  

*(11)*

as desired. \( \Box \)

4 Third answer: Dirichlet mixture

Although the normalized maximum likelihood predictor gives the optimal worst-case regret, it has a severe drawback that it is usually really hard to compute in practice. In contrast, the mixture idea used in the first answer can usually be efficiently computed (as will be shown below). This observation motivates us to come up with a mixture-based predictor \( P \), which enjoys a worst-case regret close to that of \( P_{\text{NML}} \).

The general mixture idea is as follows: suppose the model class \( F = (Q_\theta)_{\theta \in \Theta} \) can be parametrized by \( \theta \). Consider some prior \( w(\cdot) \) on \( \Theta \) and choose

\[
P_\omega(x^n) = \int_{\Theta} Q_\theta(x^n) w(d\theta)
\]  

*(14)*
as our predictor. Note that in this case,
\[ P_\omega(x_t|x_{t-1}) = \frac{P_\omega(x_t)}{P_\omega(x_{t-1})} \]
(15)
\[ = \frac{\int_\Theta Q_\theta(x_t)w(d\theta)}{\int_\Theta Q_\theta(x_{t-1})w(d\theta)} \]
(16)
\[ = \frac{\int_\Theta Q_\theta(x_t|x_{t-1})Q_\theta(x_{t-1})w(d\theta)}{\int_\Theta Q_\theta(x_{t-1})w(d\theta)} \]
(17)
\[ = \frac{\int_\Theta Q_\theta(x_t|x_{t-1})Q_\theta(x_{t-1})w(d\theta)}{\int_\Theta Q_\theta(x_{t-1})w(d\theta)} \]
(18)
\[ \triangleq \frac{\int_\Theta Q_\theta(x_t|x_{t-1})w(d\theta|x_{t-1})}{\int_\Theta Q_\theta(x_{t-1})w(d\theta)} \]
(19)

where
\[ w(d\theta|x_{t-1}) = \frac{Q_\theta(x_{t-1})w(d\theta)}{\int_\Theta Q_\theta(x_{t-1})w(d\theta)} \]
(20)

Then how to choose the prior \( w(\cdot) \)? On one hand, the prior \( w(\cdot) \) should be chosen such that it assigns roughly equal weights to the “representative” sources. On the other hand, it should be chosen in a way such that it is computationally tractable for the sequential probability assignments. We take a look at some concrete examples.

4.1 Bernoulli case

In the first example, we assume that \( Q_\theta = \text{Bern}(\theta)^n \) and \( \mathcal{F} = \{ Q_\theta \}_{\theta \in [0,1]} \). Note that in this case the model class consists of all i.i.d. distributions. We assign the Dirichlet prior
\[ w(d\theta) = \frac{1}{\pi} \frac{d\theta}{\sqrt{\theta(1-\theta)}} \sim \text{Beta}(\frac{1}{2}, \frac{1}{2}). \]
(21)

The properties of the resulting predictor are left as exercises:

**Exercise 3.** Denote by \( n_0 = n_0(x^n), n_1 = n_1(x^n) \) the number of 0’s and 1’s in the sequence \( x^n \), respectively. Prove the following:

(a) The conditional probability is given by
\[ P_\omega(x_t|x_{t-1}) = \frac{n_{x_t}(x_{t-1}) + \frac{1}{2}}{t} \]
(22)

This is called the Krichevsky–Trofimov sequential probability assignment.

(b) Show that \( \text{WCR}_n(P_\omega, \mathcal{F}) = \frac{1}{2} \log n + O(1) \).

(c) Considering the normalized maximum likelihood predictor, also show that \( \min_\rho \text{WCR}_n(P, \mathcal{F}) = \frac{1}{2} \log n + O(1) \). Hence, our Dirichlet mixture \( P_\omega \) attains the optimal leading term of the worst-case regret.

In fact, there is a general theorem on the performance of the mixture-based predictor:

**Theorem 4** (Rissanen, 1984). Let \( \mathcal{F} = \{ Q_\theta \}_{\theta \in \Theta} \) with \( \Theta \subset \mathbb{R}^d \). Under benign regularity and smoothness conditions, we have
\[ \min_\rho \text{WCR}_n(P, \mathcal{F}) = \frac{d}{2} \log n + o(\log n). \]
(23)


Moreover, there exists some Dirichlet prior \( w(\cdot) \) such that

\[
WCR_n(P_\omega, \mathcal{F}) = \frac{d}{2} \log n + o(\log n).
\]  

(24)

4.2 Markov sources

Given that we can compete with i.i.d. Bernoulli sequences, how can we compete under logarithmic loss with respect to Markov models? Assume \( \mathcal{X} = \{0, 1\} \) and denote by \( \mathcal{M}_k \) the set of all \( k \)-th order Markov models. Note that for \( Q \in \mathcal{M}_k \),

\[
Q(x^n | x_{-k+1}^0) = \prod_{i=1}^{n} Q(x_i | x_{i-k}^{i-1})
\]  

(25)

\[
= \prod_{\text{all possible } k \text{ tuples } u^k_i : x_{i-k}^{i-1} = u^k} \prod Q(x_i | u^k).
\]  

(26)

The product \( \prod_{i : x_{i-k}^{i-1} = u^k} Q(x_i | u^k) \) gives us the probability of the subsequence \( \{x_i\}_{x_{i-k}^{i-1} = u^k} \) under Bernstein \( (Q(1 | u^k)) \). Then it makes sense to employ the KT estimator on each of these subsequences.

Consider the following \( P \) such that

\[
P(x_t | x_{t-k+1}^{t-1}) = \frac{n_{x_{t-k+1}^{t-1}}(x_{t-k+1}^{t-1}) + 1/2}{n_{x_{t-k+1}^{t-1}}(x_{t-k+1}^{t-1}) + 1},
\]  

(27)

where \( n_{u^k}(x^n) = \sum_{i=1}^{n} 1(x_{i-k+1}^{i-1} = u^k) \).

Then, for this \( P \), we have

\[
WCR_n(P, \mathcal{M}_k) \leq \sum_{u^k} \frac{1}{2} \ln(n_{u^k}(x^{n-1})) + o(\log n)
\]  

(28)

\[
\leq \frac{2^k}{2} \ln \frac{n}{2^k} + o(\log n),
\]  

(29)

which matches the Rissanen lower bound on the leading term.
In the last lecture we have talked about the mixture idea for i.i.d. sources and Markov sources. Specifically, we introduced the Krichevsky–Trofimov probability assignment for a binary vector \( x^n \in \{0, 1\}^n \) given by

\[
P^{KT}(x^n) = \int \omega(\theta) \theta^{n_1(x^n)}(1-\theta)^{n_0(x^n)} d\theta = \frac{\Gamma(n_0 + \frac{1}{2})\Gamma(n_1 + \frac{1}{2})}{\left(\Gamma(\frac{1}{2})\right)^2\Gamma(n+1)},
\]

where \( \omega(\theta) \propto \frac{1}{\sqrt{\theta(1-\theta)}} \), \( n_1 \equiv n_1(x^n) \) is the number of 1’s in \( x^n \), \( n_0 \equiv n_0(x^n) \) is the number of 0’s (so that \( n = n_0 + n_1 \)) and \( \Gamma(\cdot) \) is the gamma function. Note that \( \omega(\theta) \) is a Dirichlet prior with parameters \( (\frac{1}{2}, \frac{1}{2}) \), and the integrand is just a Dirichlet distribution with parameters \( (n_0 + \frac{1}{2}, n_1 + \frac{1}{2}) \). The main result is that, for regular models with \( d \) degrees of freedom, the Dirichlet mixing idea achieves the optimal worst-case regret \( \frac{d}{2} \log n + o(\log n) \) for individual sequences of length \( n \).

In this lecture, we will extend this idea to general tree sources.

## 1 Tree Source

We first begin with an informal definition of a tree source.

**Definition 1 (Context of a symbol).** A context of a symbol \( x_t \) is a suffix of the sequence that precedes \( x_t \), that is, \( (\cdots, x_{t-2}, x_{t-1}) \).

**Definition 2 (Tree source).** A tree source is a source where a set of suffixes is sufficient to determine the probability distribution of the next item (regardless of the past) in the sequence.

A tree source is hence a generalization of an order-one Markov source, where only the previous element in the sequence determines the next: now, instead of just one element being sufficient, the suffix is sufficient. It is perhaps best elucidated by example.

**Example 3.** Let \( S = \{00, 10, 1\} \) and \( \Theta = \{\theta_s, s \in S\} \). Then

\[
\begin{align*}
p(X_t = 1 | X_{t-2} = 0, X_{t-1} = 0) &= \theta_{00} \\
p(X_t = 1 | X_{t-2} = 1, X_{t-1} = 0) &= \theta_{10} \\
p(X_t = 1 | X_{t-1} = 1) &= \theta_1
\end{align*}
\]

We can represent this structure in a binary tree:

```
          \( t-2 \)  \( t-1 \)  \( t \)   0   1
           \( \theta_0 \) \( \theta_1 \) 1
             \( 0 \) 0
               \( \theta_{00} \) \( \theta_{10} \)
```

**Remark** Note that the source in Example 3 is not first-order Markov, but can be cast as a special case of a second-order Markov source.

We denote as \( P_{S,\theta}(x^n) \) the associated probability mass function on \( x^n \in \mathcal{X}^n \).
2 Known tree, unknown parameters

Recall that with Markov sources, we could think of the pmf of any sequence as the product of conditional pmfs. For the tree sources, we can factorize similarly, but this pertains to suffixes.

Let $P_{KT}^{S}$ be the pmf induced by employing the Krichevsky–Trofimov sequence probability assignment (separately) on each subsequence corresponding to every $s \in S$. We will denote as $n_s$ the number of $x_i, 1 \leq i \leq n$ with context $s$, and we will show in Homework 4 that

$$\log \frac{1}{P_{KT}^{S}(x^n)} - \min_{\theta} \log \frac{1}{Q_{\text{Bernoulli}(\theta)}^{n}(x^n)} \leq \frac{1}{2} \log n + 1.$$

Now enumerating over different contexts, the regret on sequence $x^n$ is upper-bounded by

$$\log \frac{1}{P_{KT}^{S}(x^n)} - \min_{\theta} \log \frac{1}{P_{S,\theta}^{n}(x^n)} \leq \sum_{s \in S} \left[ \frac{1}{2} \log n_s + 1 \right]$$

$$= |S| \sum_{s \in S} \frac{1}{|S|} \left[ \frac{1}{2} \log n_s + 1 \right]$$

$$(a) \leq |S| \left[ \frac{1}{2} \log \sum_{s \in S} \frac{1}{|S|} n_s + 1 \right]$$

$$= \frac{|S|}{2} \log \frac{n}{|S|} + |S|$$

where (a) follows from Jensen’s inequality. Recall from Lecture 8 that Rissanen showed that the worst-case regret could at best be $\frac{d}{2} \log n + o(\log n)$. This shows that the above strategy—using the Krichevsky–Trofimov assignment—is optimal up to lower order terms.

3 Unknown tree

We call a context tree is of depth $D$ if it is a full binary tree of depth $D$, where node $s$ contains two integers $(n_{s0}, n_{s1})$, i.e., the number of 0’s and 1’s in the sequence $x^n$ which occurs right after the context $s$.

**Example 4.** Consider the sequence

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$x_5$</th>
<th>$x_6$</th>
<th>$x_7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

We can draw the tree, with each node $s$ labeled with $x(s)$:
For example, $x_3$ is preceded by ‘001’, so is included in $x(‘001’)$ (at $t − 3$), $x(‘01’)$ (at $t − 2$) and $x(‘1’)$ (at $t − 1$). Moreover, for $s = 001$, we have $n_{s_0} = n_{s_1} = 1$ (corresponding to $x_3$ and $x_6$, respectively).

Then the context-tree weighting at any leaf node $s$ is $P_{CTW}(x^n, s) = P^{KT}(n_{s_0}, n_{s_1})$ if depth($s$) = $D$. For any node $s$ other than a leaf node, we have

$$P_{CTW}(x^n, s) = \frac{1}{2}P^{KT}(n_{s_0}, n_{s_1}) + \frac{1}{2}P_{CTW}(x^n, 0s)P_{CTW}(x^n, 1s)$$

where $0s$ means the sequence formed by prepending ‘0’ to $s$, and $1s$ means the sequence formed by prepending ‘1’ to $s$.

Finally, the CTW probability assignment on $x^n$ can be found recursively at the root via $P_{CTW}(x^n) = P_{CTW}(x^n, \emptyset)$. 
In the last lecture we introduced how to use Krichevsky–Trofimov probability assignment to construct the context tree, and the corresponding weighting scheme for tree sources. In this lecture we will analyze the (worst-case, or the average-case) regret of the CTW algorithm under logarithmic loss.

1 Context Tree Weighting: Review

Notation: since the Krichevsky–Trofimov probability assignment only depends on the number of zeros and ones in the sequence, we define \( \vec{n} = (n_0(x^n), n_1(x^n)) \) and write \( P^{\text{KT}}(x^n) \) as \( P^{\text{KT}}(\vec{n}) \). The CTW algorithm imposes the Krichevsky–Trofimov probability assignment to every context \( s \in S \) separately, and for each context \( s \in S \), we need to associate with a vector \( \vec{n}_s = (n_{s0}(x^n), n_{s1}(x^n)) \), i.e., the number of occurrences of the context \( s0, s1 \) in the sequence \( x^n \), respectively.

Now revisit the last example with nodes labeled with their corresponding \( \vec{n} \):

```
1 1 0 0 1 1 0 ···
```

\( \vec{n}_{111} = (0,0) \)
\( \vec{n}_{011} = (1,0) \)
\( \vec{n}_{101} = (0,0) \)
\( \vec{n}_{001} = (1,1) \)
\( \vec{n}_{110} = (1,0) \)
\( \vec{n}_{010} = (1,0) \)
\( \vec{n}_{100} = (0,2) \)
\( \vec{n}_{000} = (0,0) \)

The probability assignment of the CTW algorithm on \( x^n \) and the context \( s \) is given as follows:

\[
P_{\text{CTW}}(x^n, s) = \begin{cases} 
P^{\text{KT}}(\vec{n}_s) & \text{if } \text{depth}(s) = D \\
\frac{1}{2} P^{\text{KT}}(\vec{n}_s) + \frac{1}{2} P_{\text{CTW}}(x^n, 0s) P_{\text{CTW}}(x^n, 1s) & \text{otherwise}
\end{cases}
\]

Recursively we find \( P_{\text{CTW}}(x^n) = P_{\text{CTW}}(x^n, \emptyset) \), and also the conditional probability

\[
P_{\text{CTW}}(x_{n+1}|x^n) = \frac{P_{\text{CTW}}(x^{n+1})}{P_{\text{CTW}}(x^n)}.
\]
Remark  The choice of the weight $\frac{1}{2}$'s in that second line is not important; it can be changed into any $(\alpha, 1 - \alpha)$ with $\alpha \in (0, 1)$ with almost no loss in theoretical analysis (as you may see below). In practice, the choice of $\alpha$ depends on the data.

2  Worst-Case Regret of CTW Algorithm

Consider the context tree in the previous example with depth 3, and suppose that the true tree model $S$ is as follows:

![Context Tree Diagram]

By definition of the CTW algorithm, we have

\[
\begin{align*}
P_{CTW}(x^n, 00) &\geq \frac{1}{2} P_{KT}(\vec{n}_{00}) \\
P_{CTW}(x^n, 10) &\geq \frac{1}{2} P_{KT}(\vec{n}_{10}) \\
P_{CTW}(x^n, 1) &\geq \frac{1}{2} P_{KT}(\vec{n}_1) \\
P_{CTW}(x^n, 0) &\geq \frac{1}{2} P_{CTW}(x^n, 00) P_{CTW}(x^n, 10) \\
P_{CTW}(x^n) &\geq P_{CTW}(x^n, \emptyset) \\
&\geq \frac{1}{2} P_{CTW}(x^n, 0) P_{CTW}(x^n, 1) \\
&\geq \frac{1}{2} \frac{1}{2} P_{CTW}(x^n, 00) P_{CTW}(x^n, 10) \frac{1}{2} P_{KT}(\vec{n}_1) \\
&\geq \frac{1}{25} P_{KT}(\vec{n}_{00}) P_{KT}(\vec{n}_{10}) P_{KT}(\vec{n}_1) \\
&= \frac{1}{25} P_{KT}(x^n).
\end{align*}
\]

As a result, for any sequence $x^n$, in this case we have

\[
\begin{align*}
\log \frac{1}{P_{CTW}(x^n)} - \min_{\theta} \log \frac{1}{P_{S,\theta}(x^n)} &\leq 5 + \log \frac{1}{P_{KT}(x^n)} - \min_{\theta} \log \frac{1}{P_{S,\theta}(x^n)} \\
&\leq 5 + \frac{3}{2} \log \frac{n}{3} + 3 \\
&= \frac{3}{2} \log \frac{n}{3} + 8.
\end{align*}
\]
In general, repeating the analysis above, we lose a factor of at most $\frac{1}{2}$ for each of the $|S|$ leaves and $|S| - 1$ internal nodes. Therefore, in general, for any $S$ corresponding to a tree source of depth $\leq D$,

$$\log \frac{1}{P_{CTW}(x^n)} \leq 2|S| - 1 + \log \frac{1}{P_{ST}(x^n)}$$

$$\leq 2|S| - 1 + \left[ \frac{|S|}{2} \log \frac{n}{|S|} + |S| + \min_{\theta} \log \frac{1}{P_{S,\theta}(x^n)} \right]$$

$$\leq \frac{|S|}{2} \log \frac{n}{|S|} + 3|S| - 1 + \min_{\theta} \log \frac{1}{P_{S,\theta}(x^n)},$$

which again shows the order-optimality of the context-tree weighting for any unknown tree model with depth at most $D$.

3 Average Regret of CTW Algorithm

Recall that the worst-case regret under logarithmic loss is given by

$$WCR_n(P, F) = \max_{x^n} \left[ \log \frac{1}{P(x^n)} - \min_{Q \in F} \log \frac{1}{Q(x^n)} \right].$$

Now, for all true probability distribution $Q \in F$, we have

$$D(Q_x^n \| P_x^n) = \mathbb{E}_Q \log \frac{Q(x^n)}{P(x^n)}$$

$$\leq \mathbb{E}_Q \left[ \log \frac{1}{P(x^n)} - \min_{Q' \in F} \log \frac{1}{Q'(x^n)} \right]$$

$$\leq WCR_n(P, F)$$

which shows that the average-case regret is upper bounded by the worst-case regret:

$$\sup_{Q \in F} D(Q_x^n \| P_x^n) \leq WCR_n(P, F).$$

Applying this result to the tree source, for any tree source $P_{S,\theta}$ with depth$(S) \leq D$, we have

$$D(P_{S,\theta}^n \| P_{CTW}^n) \leq \frac{|S|}{2} \log n + O(1).$$

As a result, we know from Lecture 7 that for prediction under a general loss function $\Lambda$,

$$\mathbb{E} \left[ L_{F_{CTW}}(X^n) - L_{F_{S,\theta}}(X^n) \right] \leq \Lambda_{\max} \sqrt{\frac{2}{n} \left( \frac{|S|}{2} \log n + O(1) \right)}.$$
In this lecture, we will introduce the quantity of directed information, provide motivation for estimating the directed information, and look at two estimators and their associated performance guarantees. First, though, we clarify a question from last time about the depth of context tree weighting.

1 Context Tree Weighting, Continued

Recall that for $Q_{CTW,D}$, denoting the probability assignment by context tree weighting with depth $D$, we showed in the previous lecture that

$$\log \frac{1}{Q_{CTW,D}(X^n)} - \min_{\theta} \log \frac{1}{P_{S,\theta}(X^n)} \leq \frac{|S|}{2} \log \frac{n}{|S|} + 3|S| - 1$$  \hspace{1cm} (1)

$$= \frac{|S|}{2} \log n + O(1)$$  \hspace{1cm} (2)

for any tree source $S$ with depth at most $D$. In other words, to obtain the optimal regret up to lower-order terms, we can choose the tree depth $D$ to be greater than our expected depth of the true tree source in practice.

To avoid the issue of estimating the true tree depth, we can further mix many context trees with different tree depths $D$. Let’s consider the following mixture: take any $\{w_D\}_{D\geq 1}$ such that all $w_D \geq 0$ and $\sum_{D=1}^{\infty} w_D = 1$, and let $Q = \sum_{D=1}^{\infty} w_D Q_{CTW,D}$. Now by the standard mixing argument, for any tree source $S$ with depth $D$, we have

$$\log \frac{1}{Q(X^n)} \leq \log \frac{1}{Q_{CTW,D}(X^n)} + \log \frac{1}{w_D}$$  \hspace{1cm} (3)

$$\leq \frac{|S|}{2} \log \frac{n}{|S|} + 3|S| - 1 + \log \frac{1}{w_D}.$$  \hspace{1cm} (4)

Thus, the mixture $Q$ as defined above is also guaranteed to perform within a constant term of optimal regret for any tree source $S$.

2 Universality

Note: Remember that if some data is governed by a law $P$, the excess log-loss of the probability assignment $Q$ (beyond what is optimal) is the relative entropy $D(P_X^n||Q_X^n)$, normalized to $\frac{1}{n} D(P_X^n||Q_X^n)$.

Definition 1 (Stationary). Probability measure $P$ on $X^\infty$ is stationary if for any $k$-tuple $(X_{i_1}, X_{i_2}, X_{i_3}, \cdots, X_{i_k})$ and any $m$,

$$(X_{i_1}, X_{i_2}, X_{i_3}, \cdots, X_{i_k}) \overset{d}{=} (X_{i_1+m}, X_{i_2+m}, X_{i_3+m}, \cdots, X_{i_k+m})$$  \hspace{1cm} (5)

The shifted $k$-tuple is equal in distribution to the original. This is also described as shift-invariance and is a common assumption for an unknown underlying distribution.

Definition 2 (Universal). Probability measure $Q$ on $X^\infty$ is universal if

$$\frac{1}{n} D(P_X^n||Q_X^n) \overset{n\to\infty}{\longrightarrow} 0$$  \hspace{1cm} (6)

for all stationary laws $P$. 


Exercise: Show that the mixture \( Q \) in (3) is universal.

Universality of \( Q \) is clearly a desirable property, as we are guaranteed to have a vanishing excess log-loss with any stationary law \( P \). Additionally, if some \( Q \) is universal, it can be used to estimate other quantities of interest, such as directed information.

3 Directed Information

**Definition 3** (Directed information). The **directed information** from \( X^n \) to \( Y^n \) is defined as

\[
I(X^n \to Y^n) \triangleq \sum_{i=1}^{n} I(X^i; Y_i|Y^{i-1})
\]

More generally,

\[
I(X^{n-d} \to Y^n) \triangleq I((\emptyset, X^{n-d}) \to Y^n) = \sum_{i=1}^{n} I(X^{i-d}; Y_i|Y^{i-1}).
\]

We remark that mutual information is the canonical measure of the relevance of one process to another, and the directed information is the canonical measure of the causal relevance of one process to another. Notice that directed information is a summation of mutual information between the first \( i \) elements of \( X \) and the \( i \)th element of \( Y \), conditioned on the previous \( i-1 \) values of \( Y \). Namely, we measure the causal relevance of \( X^i \) for our prediction of \( Y_i \), given \( Y^{i-1} \). The following two examples will illustrate why this is a very natural metric in prediction.

**Example 1:** Suppose that \((X_1, X_2, \cdots, X_n)\) and \((Y_1, Y_2, \cdots, Y_n)\) represent two stocks in the New York Stock Exchange. You are a young investor desperate to impress your boss at the hedge fund, and you would like to predict the price of stock \( Y \) tomorrow, i.e., at day \( i \). You can look solely at \( Y^{i-1} \), but, being more resourceful than that, you decide to determine the extent to which \( X^{i-1} \) also informs \( Y_i \)—that is, \( I(X^{i-1};Y_i|Y^{i-1}) \). Collected over time, we have \( \sum_{i=1}^{n} I(X^{i-1};Y_i|Y^{i-1}) \), which is equivalent to \( I(X^{n-1} \to Y^n) \).

**Example 2:** Now, on the basis of your excellent application of directed information, you have been promoted to hedge fund manager. Your new role requires you to be prescient of global trends, so this time let \((X_1, X_2, \cdots, X_n)\) be the Hang Seng Index, an overall performance measure for the Hong Kong Stock Exchange, and let \((Y_1, Y_2, \cdots, Y_n)\) be the Dow Jones Industrial Average. Because of the time difference between New York and Hong Kong, the trading has already closed in Hong Kong before the opening bell in New York. Therefore, when you predict \( Y_i \), you now have access to \( X_i \) in addition to \( X^{i-1} \) and \( Y^{i-1} \) as before. The conditional mutual information is \( I(X^i;Y_i|Y^{i-1}) \), and over a period of \( n \) days we get \( \sum_{i=1}^{n} I(X^i;Y_i|Y^{i-1}) = I(X^n \to Y^n) \).

We describe mutual information as a measure of relevance or correlation between two processes, without implying causation. Directed information allows us to be more explicit about the flow of information from one process to another. The relation is shown below, to be proved as an exercise.

**Exercise:** Show that \( I(X^n;Y^n) = I(X^n \to Y^n) + I(Y^{n-1} \to X^n) \)

Also note that this implies that \( I(X^n \to Y^n) \leq I(X^n;Y^n) \):

\[
I(X^n \to Y^n) = \sum_{i=1}^{n} I(X^i; Y_i|Y^{i-1}) \leq \sum_{i=1}^{n} I(X^n; Y_i|Y^{i-1}) = I(X^n; Y^n)
\]

where the second step is given by the data processing inequality.
4 Estimation of Directed Information

**Definition 4** (Directed information rate). Let $X$ and $Y$ be a pair of processes. The **directed information rate** from $X$ to $Y$ is

$$I(X \rightarrow Y) \triangleq \lim_{n \to \infty} \frac{1}{n} I(X^n \rightarrow Y^n),$$

provided that the limit exists.

*Exercise*: Show that the limit exists if $X$ and $Y$ are jointly stationary.

*Note*: Recall that for $X \sim P_X$, the entropy may be written $H(X)$ or equivalently $H(P_X)$, which is a function of the distribution.

Notice that

$$I(X^n \rightarrow Y^n) = \sum_{i=1}^{n} I(X^i; Y^i-1) = \sum_{i=1}^{n} [H(Y_i|Y^{i-1}) - H(Y_i|X^i, Y^{i-1})]$$  \hspace{1cm} (11)

This suggests an estimator for the directed information rate. Specifically,

$$\hat{I}(X^n \rightarrow Y^n) = \frac{1}{n} \sum_{i=1}^{n} [H(Y_i|Y^{i-1}) - H(Y_i|X^i, Y^{i-1})].$$  \hspace{1cm} (12)

Since the (conditional) entropy solely depends on the underlying unknown distribution, we can consider another probability assignment of our choice $Q$. The idea is that if $Q$ is universal, then the (conditional) entropy based on $Q$ should converge to that based on the true $P$.

**Theorem 5.** If $Q$ is universal and $(X, Y)$ are jointly stationary and ergodic (a concept we will not introduce here, which basically means that the law of large numbers holds in the process), then

$$E[I(X^n \rightarrow Y^n) - \frac{1}{n} \hat{I}(X^n \rightarrow Y^n)] \xrightarrow{n \to \infty} 0$$  \hspace{1cm} (13)

Therefore, this estimator of the directed information rate converges to the true rate.

With a few more constraints, we can further bound how quickly the estimate converges.

**Theorem 6.** If $Q$ is derived from context tree weighting (CTW) and the joint process of $(X, Y)$ is stationary, ergodic, aperiodic, and Markov (of arbitrary order), then

$$E[I(X^n \rightarrow Y^n) - \frac{1}{n} \hat{I}(X^n \rightarrow Y^n)] \leq C \frac{\log n}{\sqrt{n}}$$

where $C > 0$ is a universal constant not depending on $n$.

For a more in-depth discussion, please see the paper “Universal Estimation of Directed Information” (Jiao, Permuter, Zhao, Kim, and Weissman, 2012).

Another estimator is suggested by the observation that

$$I(X^n \rightarrow Y^n) = D(P_{Y|X^n,Y_{i-1}} P_{X^n,Y_{i-1}})$$  \hspace{1cm} (15)

with the proof left as an exercise. We can then construct

$$\hat{I}(X^n \rightarrow Y^n) \triangleq \frac{1}{n} \sum_{i=1}^{n} D(Q_{Y|X^n,Y_{i-1}} Q_{X^n,Y_{i-1}})$$  \hspace{1cm} (16)

for our suitable choice of $Q$. We omit the details here, but remark that this estimator has similar performance guarantees to the first, and often has a slightly smoother convergence in practice as relative entropy is always non-negative.
In this lecture\textsuperscript{1}, we set the state for the Discrete Universal Denoiser (DUDE), an algorithm for denoising sequences whose symbols take values in a discrete alphabet. The following table shows at a glance how this new problem setting compares to those of the algorithms we have studied in the past:

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Key idea</th>
<th>Input sequence alphabet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Viterbi algorithm/Kalman filter</td>
<td>Maximum a posteriori (MAP) decoding</td>
<td>Continuous</td>
</tr>
<tr>
<td>Context-tree weighting (CTW) algorithm</td>
<td>Decoding from context</td>
<td>Discrete, generated by a tree source with unknown parameters</td>
</tr>
<tr>
<td>Discrete Universal Denoiser (DUDE)</td>
<td>MAP decoding from context</td>
<td>Discrete, generating distribution unknown</td>
</tr>
</tbody>
</table>

1 Setting

Figure 1 shows a toy setting we consider for discrete denoising.

\[ x^n = (x_1, \ldots, x_n) \in \mathcal{X}^n \quad \text{Noisy Channel} \quad z^n \in \mathcal{Z}^n \quad \text{Denoiser} \quad \hat{x}^n(z^n) = (\hat{x}_1(z^n), \hat{x}_2(z^n), \ldots, \hat{x}_n(z^n)) \in \hat{\mathcal{X}}^n \]

**Figure 1:** Denoising setting. In this toy example, we assume that the parameters of the source \( X \) (i.e., \( p_X(x^n) \)), as well as the conditional probabilities of each output symbol produced by the channel on any given input, \( p_{X,Z}(x, z) \), are known.

We assume all noiseless sequences and observations are composed of symbols which take values in a finite alphabet, \( \mathcal{X} = \{a_1, \ldots, a_{|\mathcal{X}|}\} \) (thus the ‘discrete’ in the name of the denoiser). Denoising means that given a noisy observation sequence \( z^n \), we would like to reconstruct the original noiseless signal, \( x^n \), by outputting a best estimate \( \hat{x}^n(z^n) \). Assume we have a per-symbol loss function \( \Lambda : \mathcal{X} \times \hat{\mathcal{X}} \to \mathbb{R} \) and are allowed to look at an entire length-\( n \) observation \( z^n \) to guess \( \hat{x}^n \).

**Definition 1.** An n-block denoiser is a mapping

\[ \hat{X}^n(Z^n) : \mathcal{Z}^n \to \hat{\mathcal{X}}^n. \]

As in the past, we measure the performance of \( \hat{X}^n \) by the normalized cumulative loss on a block \( z^n \):

\[ L_{\hat{X}^n}(x^n, z^n) = \frac{1}{n} \sum_{i=1}^{n} \Lambda(x_i, \hat{X}^n(z^n)[i]). \quad (1) \]

\textsuperscript{1}Reading: Universal Discrete Denoising: Known Channel, T. Weissman, E. Ordentlich, G. Seroussi, S. Verdu, M.J. Weinberger, 2005.
(From now on we use the notation \( \hat{X}^n(z^n)[i] \leftarrow \hat{x}_i(z^n) \).)

Let us start our investigation by assuming that the symbols are generated probabilistically with known parameters, i.e., probabilities of all symbols, channel and noise are known to the denoiser. On the cumulative loss, an optimal denoiser is then described by

\[
\arg \min_{\hat{X}^n(\cdot) \in \mathcal{D}_n} E [L_{\hat{X}^n}(x^n, z^n)] = \arg \min_{\hat{X}^n(\cdot) \in \mathcal{D}_n} \left[ \frac{1}{n} \sum_{i=1}^{n} \Lambda(x_i, \hat{x}_i(z^n)) \right]
\]

where \( \mathcal{D}_n \) denotes the set of all block-\( n \) denoisers. We can observe from the right-hand side of (2) that the cumulative loss is separable; it is a sum of terms, each of which depends only on \( x_i \) and \( \hat{x}_i(z^n) \). This suggests a greedy strategy for producing such an estimate; under this strategy, \( \hat{x}_i(z^n) \) is nothing but the Bayes response to the conditional posterior probability of each symbol:

\[
\hat{x}_i(z^n) = \hat{X}_{\text{Bayes}}(P_{X_i|Z^n = z^n})
\]

and the corresponding value of expected loss is the Bayes envelope,

\[
\frac{1}{n} \sum_{i=1}^{n} E \left[ U(P_{X_i|Z^n}) \right].
\]

We remind the readers that the Bayes envelope is defined as \( U(P_X) := \min_{\hat{x} \in \hat{X}} \sum_x P_X(x) \Lambda(x, \hat{x}) \). In addition, the discreteness of the input alphabet permits us to write a loss matrix that will be useful in later sections:

\[
\{ \Lambda(\cdot, \hat{x}) \} := \left( \begin{array}{c|c|c} \lambda_{a_1} & \lambda_{a_2} & \ldots & \lambda_{a_{|X|}} \end{array} \right)
\]

where \( \lambda_{a_i} \) is the column vector \( \Lambda(\cdot, \hat{x} = a_i) \). With this definition in hand, we may rewrite the Bayes envelope in terms of the \( \lambda_{a_i} \)s:

\[
U(P_X) = \min_{\hat{x}} P_X^T \cdot \lambda_{\hat{x}}
\]

The conclusion of this analysis reveals that the Bayes response denoiser (3) achieves the minimum loss on a sequence with known generating distribution. But we have glossed over some things in the above analysis: Firstly, the Kalman filter showed us that deriving the Bayes envelope is already hard in the continuous case; what more when the symbols come from a discrete alphabet. Secondly, we should not expect realistically to be handed on a silver platter with the probabilistic parameters of the problem; for one, we often do not know the parameters of the generating distribution of the given sequence. We have in fact treaded in these waters before; see the CTW algorithm.

2 Towards construction of a universal scheme

From now on we assume that we do not know the generating distribution of the sequence symbols, but we have some characterization of the channel. We would like to construct a universal scheme, i.e., a scheme that attains optimal performance asymptotically no matter the true law that governs the data. Here, the “optimal performance denoiser” is \( \hat{X}_{\text{Bayes}}(P_{X|Z^n}) \) as established previously.

In this section we consider the “single-letter” problem as shown in Figure 2, where our input sequence is now only one symbol, \( X \), and we are to guess \( \hat{X} \) based on \( Z \). Again, \( X \) and \( \hat{X} \) take values in finite alphabet, \( X = \{ a_1, \ldots, a_{|X|} \} \). We can also define the channel transition matrix showing probabilities of mapping inputs to outputs:

\[
\{ \Pi(x, z) \} := \left( \begin{array}{c|c|c|c} \pi_{a_1} & \pi_{a_2} & \ldots & \pi_{a_{|X|}} \end{array} \right)
\]
where $\pi_{a_i}$ is the column vector $P_{X,Z}(\cdot, Z = a_i)$.

Before we proceed, let us define the Schur/Hadamard product represented by the symbol $\odot$:

**Definition 2.** For vectors $v_1, v_2$, $v_1 \odot v_2$ is the vector obtained by element-wise multiplication.

$$[v_1 \odot v_2]_i = v_{1,i} v_{2,i}$$

Secondly, we generalize the Bayes response (implicitly defined on our loss function $\Lambda(x, \hat{x})$) to take an arbitrary argument $v$:

$$\hat{X}_{\text{Bayes}}(v) = \arg \min_{\hat{x}} v^T \cdot \lambda_{\hat{x}}$$

With these definitions in hand, we may factorize $P_Z$ with a simple calculation:

$$P_Z(z) := \sum_x P_{X,Z}(x,z) = \sum_x P_X(x)\Pi(x,z) = P_X^T \cdot \pi_z = [P_X^T \cdot \Pi](z)$$

i.e. $P_Z^T = P_X^T \cdot \Pi$. Assume now that $|X| = |Z|$ and $\Pi$ is invertible\(^2\). Then we may re-write this as

$$P_X = \Pi^{-T} P_Z$$

and we may readily write the conditional probability

$$P_{X|Z=z}(x) := \frac{P_X(x)\Pi(x,z)}{P_Z(z)} = \frac{[\Pi^{-T} P_Z](x) \cdot \pi_z(x)}{P_Z(z)}$$

$$P_{X|Z=x} = \frac{[\Pi^{-T} P_Z] \odot \pi_Z}{P_Z(z)}$$

Let $\phi(z)$ represent the operation of denoising. The previous section established that

$$\min_{\phi(\cdot)} \mathbb{E}[\Lambda(x, \phi(z))] = \mathbb{E}[U(P_{X|Z})]$$

and this is achieved by

$$\phi_{\text{opt}}(z) = \hat{X}_{\text{Bayes}}(P_{X|Z=z}) = \hat{X}_{\text{Bayes}}([\Pi^{-T} P_Z] \odot \pi_Z)$$

where we have elided the denominator of (9) in the substitution because the overall normalization is not a function of $\hat{x}$ and does not affect the minimization.

We can generalize the above analysis to situations where $|X| \leq |Z|$, i.e., $\Pi$ is non-square but of full row-rank: in those situations (to be proven in the homework) we have

$$\phi_{\text{opt}}(z) = \hat{X}_{\text{Bayes}}(P_{X|Z=z}) = \hat{X}_{\text{Bayes}}([\Pi \Pi^T]^{-1} \Pi \cdot P_Z \odot \pi_Z).$$

By the way, $(\Pi \Pi^T)^{-1} \Pi$ can be defined for any arbitrary non-square matrix $\Pi$ and is known as the Moore-Penrose pseudoinverse.

\(^2\)Usually a safe assumption but notable exceptions include the binary symmetric channel with flipping probability $\frac{1}{2}$ – which is not a useful channel anyway.
3 Universal Discrete Denoising and introduction to DUDE

Now we extend the above analysis to denoising of a sequence, not just a single letter – and in doing so we introduce the DUDE. Figure 1 is applicable to our new setting too, but now we have far less knowledge of the system. More concretely, the assumptions are now:

1. Finite alphabets $\mathcal{X}, \mathcal{Z}, \hat{\mathcal{X}}$.

2. Source $X$ is governed by a probabilistic and stationary but otherwise unknown generating distribution.

3. Channel is known to be a discrete memoryless channel; and $\Pi$ is known and of full row rank.

4. Loss function $\Lambda$ is given.

The idea of DUDE is that we “correct by the context”; whereas in Section 1 we had the luxury of knowing $P_{Z_i|Z^n=z^n}$ which we could input into the Bayes response function, now we must estimate that. We do so by looking at left and right length-$k$ contexts around the symbol to be denoised:

$$z_1 \ldots z_{i-k} \ldots z_{i-1} z_i z_{i+1} \ldots z_{i+k} \ldots z_n$$

For every pair of left and right contexts, we can define a count vector $m \in \mathbb{R}^{\lvert \mathcal{Z} \rvert}$ to capture the number of times a symbol $z \in \mathcal{Z}$ appears in these contexts over the entire sequence

$$m(z^n, \ell^k, r^k)[z] := \{k + 1 \leq i \leq n - k : z_{i-1} = \ell^k, z_{i+k} = r^k, z_i = z\},$$

the DUDE denoiser effectively calculates the Bayes response (3) with this empirical probability distribution. That is, DUDE($k$) performs the function $F$ where:

$$\hat{x}_i(z^n) = F(\Lambda, \Pi, m(z^n, z_{i-k}^{i-1}, z_{i+1}^{i+k}), z_i) \quad k + 1 \leq i \leq n - k$$

$$= \arg\min_{\hat{x}^T} \lambda^T \left[ [\Pi^{-T} \cdot m(z^n, z_{i-k}^{i-1}, z_{i+1}^{i+k}) \odot \pi] \right].$$

Readers will note the similarity with (10). Surprisingly, it will turn out that the operation performed by the DUDE is linear in $n$. 

4
1 Recap

The system model of the denoising problem is presented in Fig. 1. Let $X^n$ be the unknown input sequence that passes through a discrete memoryless channel. The channel is characterized by its transition matrix $\Pi$ which is known and is of full row rank. The output of the noisy channel is denoted by $Z^n$ which is observed by the denoiser. The denoiser then outputs the reconstruction sequence $\hat{X}^n$. Here, $X_i \in \mathcal{X}$, $Z_i \in \mathcal{Z}$, $\hat{X}_i \in \hat{\mathcal{X}}$, where $\mathcal{X}$, $\mathcal{Z}$, and $\hat{\mathcal{X}}$ are finite alphabets. The distortion of the reconstruction sequence is characterized by a given loss function $\Lambda : \mathcal{X} \times \hat{\mathcal{X}} \to [0, +\infty)$.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{denoising_diagram}
\caption{Denoising setting.}
\end{figure}

The DUDE with context length $k$ works in the following way: in the first pass, the denoiser derives the counting vector of dimension $|Z|$ as follows:

$$m(z^n, l^k, r^k)[z] = |\{ k + 1 \leq i \leq n - k : z^{i+k}_{i-k} = (l^k, z, r^k)\} |.$$  

Note that $m[z]$ counts the number of items in the sequence $z^n$ that equals $z$ with left context $l^k$ and right context $r^k$. In the second pass, DUDE outputs the reconstruction sequence $\hat{X}^n$ with

$$\hat{X}_i(z^n) = \phi(\Lambda, \Pi, m(z^n, z^{i-1}_{i-k}, z^{i+k}_{i+1}), z_i) \text{, for } k + 1 \leq i \leq n - k,$$

where

$$\phi(\Lambda, \Pi, v, z) = \hat{X}_{\text{Bayes}} \left( (\Pi \Pi^T)^{-1} \Pi v \odot \pi_z \right)$$

$$= \arg \min_{\hat{x}} \lambda^T_{\hat{x}} \left( (\Pi \Pi^T)^{-1} \Pi v \odot \pi_z \right).$$

**Exercise 1.** Consider the following specific examples:

- For BSC($\delta$) and Hamming loss, i.e., $\Pi = \begin{pmatrix} 1 - \delta & \delta \\ \delta & 1 - \delta \end{pmatrix}$, $\Lambda = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$. Show that

$$\phi(\Lambda, \Pi, v, z) = \begin{cases} z \\ \frac{v(z)}{\pi(z)} \geq \frac{2\delta(1-\delta)}{\delta^2 + (1-\delta)^2} \end{cases}.$$  

- For BEC($\epsilon$) and Hamming loss, the alphabets are $\mathcal{Z} = \mathcal{X} \cup \{e\}$, $\hat{\mathcal{X}} = \mathcal{X}$, and

$$\Pi(x, z) = \begin{cases} 1 - \epsilon & z = x \\ \epsilon & z = e \\ 0 & \text{otherwise} \end{cases}.$$  

Show that

$$\phi(\Lambda, \Pi, v, z) = \begin{cases} z \\ \arg \max_{\hat{x} \in \hat{\mathcal{X}}} v(\hat{x}) \text{, } z \neq e \\ z = e \end{cases}.$$
2 Choice of the context length $k$

The choice of $k$ is important for the performance of DUDE. On one hand, a large $k$ provides more information in each context sequence; on the other hand, when $k$ is large, we will have a small number of counts and might suffer from unreliable context statistics. In practice, we choose

$$k = k_n = \left\lceil \frac{1}{5} \log n \frac{\log |Z|}{\log n} \right\rceil,$$

which can be justified through performance analysis. In the following sections, we denote by $\hat{X}_n^{DUDE}$ the reconstructed sequence given by the DUDE with context length $k_n$.

3 Performance of DUDE

3.1 Stochastic setting

In this section we assume that the source $X$ is a stationary stochastic process. We introduce the definition of denoisibility as follows:

$$D(X, Z) = \lim_{n \to \infty} \min_{\hat{X}^n \in \mathcal{D}_n} \mathbb{E} L_{\hat{X}_n}(X^n, Z^n),$$

where $\mathcal{D}_n$ is the set of all denoisers of block length $n$. Note that

$$\min_{\hat{X}^n \in \mathcal{D}_n} \mathbb{E} L_{\hat{X}_n}(X^n, Z^n) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E} U(P_{X_i|Z^n}).$$

We introduce the properties of $D(X, Z)$ in the following exercises.

Exercise 2.

- If $(X, Z)$ is jointly stationary, then $D(X, Z)$ exists.
- $D(X, Z) = \lim_{l \to \infty} \mathbb{E} U(P_{X_0|Z_{-1}^l})$. Note that when $X, Z$ are jointly stationary $\mathbb{E} U(P_{X_i|Z^n}) = \mathbb{E} U(P_{X_0|Z_{-1}^{i-1}})$.

In the following theorem, we show the optimality of DUDE for the stochastic setting.

**Theorem 3.** For any stationary process $X$,

$$\lim_{n \to \infty} \mathbb{E} L_{\hat{X}^{DUDE}_n}(X^n, Z^n) = D(X, Z).$$

3.2 Semi-stochastic setting

In this section, we assume that the source $x = (x_1, x_2, \ldots)$ is some deterministic sequence. Note that $Z$ is still a stochastic process with distribution $\mathbb{P}(Z^n = z^n) = \prod_{i=1}^{n} \pi(x_i, z_i)$. This is actually more general than the stochastic setting since we assume nothing about the input sequence.

Define

$$D_k(x^n, z^n) = \min_{f:Z^{2k+1} \to X} \frac{1}{n} \sum_{i=k+1}^{n-k} \Lambda(x_i, f(z^{i+k}_{i-k}))$$

(12)

to be the error induced by the optimal denoiser with double-sided context length $k$. Note that here the optimal denoiser has access to both the original sequence and the noisy sequence, but is restricted to have the same output for symbols with the same length-$k$ contexts.

**Theorem 4.** For any source sequence $x$,

$$\lim_{n \to \infty} \left( L_{\hat{X}^{DUDE}_n}(x^n, Z^n) - D_k(x^n, Z^n) \right) = 0 \quad w.p.1.$$
Note that Theorem 4 is stronger than Theorem 3. We can show that Theorem 3 can be proved directly by Theorem 4. We first introduce some mathematical background.

For a sequence \( \{a_n\} \), we define

\[
\limsup_{n \to \infty} a_n = \lim_{n \to \infty} \sup_{m \geq n} a_m, \tag{14}
\]

\[
\liminf_{n \to \infty} a_n = \lim_{n \to \infty} \inf_{m \geq n} a_m. \tag{15}
\]

Note that \( \limsup \) and \( \liminf \) exist for any sequence. Note also the following fact:

**Fact 5.** \( \lim_{n \to \infty} a_n \) exists iff. \( \limsup_{n \to \infty} a_n = \liminf_{n \to \infty} a_n \).

We will also need the following lemma:

**Lemma 6** (Fatou’s lemma). If \( \{R_n\} \) is a sequence of non-negative random variables, then

\[
\mathbb{E} [\liminf_{n \to \infty} R_n] \leq \liminf_{n \to \infty} \mathbb{E} [R_n]. \tag{16}
\]

With the preparation above, we prove Theorem 3 with Theorem 4.

**Proof** For any fixed \( l \), we have

\[
0 = \mathbb{E} \left[ \limsup_{n \to \infty} \left( L_{X^n_{DUDE}} (x^n, Z^n) - D_{k_n} (x^n, Z^n) \right) \right] \tag{17}
\]

\[
\geq \limsup_{n \to \infty} \left[ \mathbb{E} L_{X^n_{DUDE}} (x^n, Z^n) - \mathbb{E} D_{k_n} (x^n, Z^n) \right] \tag{18}
\]

\[
\geq \limsup_{n \to \infty} \left[ \mathbb{E} L_{X^n_{DUDE}} (x^n, Z^n) - \mathbb{E} D_l (x^n, Z^n) \right], \tag{19}
\]

where the first inequality follows from Fatou’s lemma. Note that we can prove

\[
\mathbb{E} D_l (x^n, Z^n) \leq \mathbb{E} U \left( P_{X_0|Z_{l-1}} \right). \tag{20}
\]

Hence, \( \limsup_{n \to \infty} \mathbb{E} L_{X^n_{DUDE}} (x^n, Z^n) \leq \mathbb{E} U \left( P_{X_0|Z_{l-1}} \right) \). Since \( l \) is arbitrary, we have

\[
\limsup_{n \to \infty} \mathbb{E} L_{X^n_{DUDE}} (x^n, Z^n) \leq \lim_{l \to \infty} \mathbb{E} U \left( P_{X_0|Z_{l-1}} \right) = \mathbb{D}(X, Z). \tag{21}
\]

On the other hand, \( \mathbb{D}(X, Z) \) measures the fundamental denoisability, and we must have

\[
\liminf_{n \to \infty} \mathbb{E} L_{X^n_{DUDE}} (x^n, Z^n) \geq \mathbb{D}(X, Z) \tag{22}
\]
1 Introduction

DUDE considers the denoising setting with a discrete source alphabet and reconstruction alphabet, and performs well under both the stochastic and semi-stochastic settings. However, the context-based counting idea used by DUDE cannot be directly applied to the continuous setting. In this lecture we start to look at a new problem: the nonparametric function estimation problem with a continuous alphabet and new assumptions on the underlying function.

We look at the following regression setting: we have \( n \) observations \( y_i = f(x_i) + \sigma \xi_i, \quad i = 1, \cdots, n \) (1)
where \( f(\cdot) \) is our target function on \([0, 1]\) which we assume to lie in some prescribed function class \( F \), \( x_i \) are equally spaced points on \([0, 1]\), \( \sigma \) is the known noise level, and \( \xi_i \sim \mathcal{N}(0, 1) \) are i.i.d Gaussian noise. Our target is to find some estimator \( \hat{f} \) which comes close to minimizing the worst-case \( L_q \) risk:
\[
\sup_{f \in F} \mathbb{E} f \| \hat{f} - f \|_q \lesssim \inf_{\hat{f}^* \in F} \sup_{f \in F} E f \| \hat{f}^* - f \|_q. \tag{2}
\]

Notations: for non-negative sequences \( \{a_n\}, \{b_n\} \), we use \( a_n \lesssim b_n \) to denote \( \limsup_{n \to \infty} \frac{a_n}{b_n} < \infty \), i.e., there exists universal constant \( C \) not depending on \( n \) such that \( a_n \leq C b_n \) for any \( n \in \mathbb{N} \). Similarly, we write \( a_n \gtrsim b_n \) to denote \( b_n \lesssim a_n \), and \( a_n \asymp b_n \) to denote \( a_n \lesssim b_n \) and \( b_n \lesssim a_n \).

References:

2 The Bias–Variance Decomposition

Before looking at specific function class \( F \) and estimators \( \hat{f} \), we take a look at how to evaluate the performance of any given estimator \( \hat{f} \). We define the following two quantities:

1. Deterministic error: \( b(x) \triangleq \mathbb{E}_f \hat{f}(x) - f(x) \); this term corresponds to the bias, and is a deterministic scalar.

2. Stochastic error: \( s(x) \triangleq \hat{f}(x) - \mathbb{E}_f \hat{f}(x) \); this term corresponds to the variance, and is a random variable.

With the previous definition, for the \( L_q \) loss with \( 1 \leq q < \infty \), we can write
\[
\mathbb{E}_f \| \hat{f} - f \|_q = \mathbb{E}_f \| \hat{f} - \mathbb{E}_f \hat{f} + \mathbb{E}_f \hat{f} - f \|_q \tag{3}
\leq \mathbb{E}_f \| \hat{f} - \mathbb{E}_f \hat{f} \|_q + \mathbb{E}_f \| \mathbb{E}_f \hat{f} - f \|_q \tag{4}
= \mathbb{E}_f \| s \|_q + \| b \|_q \tag{5}
\leq (\mathbb{E}_f \| s \|^2_q)^{\frac{1}{2}} + \| b \|_q \tag{6}
\]
where we have used the triangle inequality and Jensen’s inequality. As a result, we can decompose the \( L_q \) risk into the bias term \( \| b \|_q \) and the variance term \( (\mathbb{E}_f \| s \|^2_q)^{\frac{1}{2}} \) separately.
Function Estimation in Hölder Balls

First we assume that the function class $\mathcal{F}$ is the Hölder ball with smoothness parameter $s$, and construct a sound estimator $\hat{f}$ for $f$ in this case.

3.1 Introduction to Hölder Balls

The Hölder ball $H^s(L)$ consists of all functions which are Hölder smooth of order $s > 0$, i.e., for $s = m + \alpha$ with $m \in \mathbb{N}, \alpha \in (0, 1]$, $\mathcal{H}^s(L) = \left\{ f \in C[0, 1] : \sup_{x \neq y \in [0, 1]} \frac{|f^{(m)}(x) - f^{(m)}(y)|}{|x - y|^\alpha} \leq L \right\}$. (7)

In other words, the Hölder ball consists of functions which are “smooth” of a certain smoothness order.

Exercise 1. Show that for $s > 1$, if we use $m = 0, \alpha = s$ in the definition (7), then the only function which satisfies the new definition is the constant function.

3.2 Kernel-based Estimator

Now we consider how to find an estimator $\hat{f}$ of $f$. First suppose that there is no noise, i.e., $\sigma = 0$. In this case, we know perfectly the value of $f$ on points $x_i$, but do not know the value of $f(x) \in (x_i, x_{i+1})$ in between. However, since we know that the function $f$ is smooth, it may be tempted to use interpolation to obtain $f(x)$, e.g.,

$$\hat{f}(x) = \frac{x_{i+1} - x}{x_{i+1} - x_i} f(x_i) + \frac{x - x_i}{x_{i+1} - x_i} f(x_{i+1}).$$

(8)

In general, we may write

$$\hat{f}(x) = \sum_{i : x_i \in I_h(x)} w(x, x_i) f(x_i)$$

(9)

for some weight function $w(x, x_i)$, where $I_h(x) \triangleq [x - h, x + h]$ is a small “window” around $x$ with bandwidth $h$ (to be specified later). Also in the noisy case where only $y_i$ is available, we may construct

$$\hat{f}(x) = \sum_{i : x_i \in I_h(x)} w(x, x_i) y_i.$$ 

(10)

Note that for simplicity we have ignored the boundary effect when $x$ is close to zero or one.

Now we specify our choice of the weight function. A natural requirement is that $\sum_{i : x_i \in I_h(x)} w(x, x_i) = 1$ for any $x$, i.e., the weights should sum into one. This requirement can be generalized in the following way: when $f(x) = x^k$ for $k = 0, \cdots, m$, the estimator in the LHS of (9) should be equal to the true value $f(x) = x^k$. In other words, we require that

$$\sum_{i : x_i \in I_h(x)} w(x, x_i)x_i^k = x^k, \quad \forall k = 0, 1, \cdots, m.$$ 

(11)

Note that as long as $nh \to \infty$, (11) can be satisfied since we have $2nh$ degrees of freedom while there are only $m + 1$ linear constraints. The following lemma shows some properties of the weights:

Lemma 2. For any $x \in [0, 1]$, there exists some weight function $w(x, \cdot)$ which satisfies (11) and

$$\|w(x, \cdot)\|_1 \lesssim 1, \quad \|w(x, \cdot)\|_2 \lesssim \frac{1}{\sqrt{nh}}.$$ 

(12)

As a sanity check, just consider the case where $m = 0$ and $w(x, x_i) = \frac{1}{2nh}$ for any $x_i \in I_h(x)$. 


3.3 Performance Analysis

Now we analyze the performance of the estimator in (10). By the bias–variance decomposition, it suffices to deal with the variance and the bias separately.

For the variance (stochastic error), we have

\[ s(x) = \hat{f}(x) - E_{\hat{f}}f(x) \]
\[ = \sum_{i : x_i \in I_h(x)} w(x, x_i)y_i - \sum_{i : x_i \in I_h(x)} w(x, x_i)E_{\hat{f}}y_i \]
\[ = \sum_{i : x_i \in I_h(x)} w(x, x_i)(f(x_i) + \sigma \xi_i) - \sum_{i : x_i \in I_h(x)} w(x, x_i)f(x_i) \]
\[ = \sigma \sum_{i : x_i \in I_h(x)} w(x, x_i)\xi_i \]
\[ \sim \mathcal{N}(0, \sigma^2 \|w(x, \cdot)\|_2^2). \]  

Since for \( X \sim \mathcal{N}(0, \sigma^2) \), we have \( \mathbb{E}|X|^q \lesssim \sigma^q \) for any \( q \in [1, \infty) \), for the stochastic error we have \( \mathbb{E}|s(x)|^q \lesssim \|w(x, \cdot)\|_2^q \) (suppressing the dependence on \( \sigma \)), and thus

\[ (\mathbb{E}\|s\|_q^q) \frac{1}{q} \lesssim \|w(x, \cdot)\|_2 \lesssim \frac{1}{\sqrt{nh}} \]  

where in the last step we have used Lemma 2.

Now we take a look at the bias. By definition,

\[ b(x) = E_{\hat{f}}f(x) - f(x) = \sum_{i : x_i \in I_h(x)} w(x, x_i)f(x_i) - f(x). \]

By our choice of the weights, we know that for any polynomial \( P(x) \) of degree at most \( m \), we have

\[ |b(x)| = \left| \sum_{i : x_i \in I_h(x)} w(x, x_i)(f(x_i) - P(x_i)) + (f(x) - P(x)) \right| \]
\[ \leq (\|w(x, \cdot)\|_1 + 1) \cdot \|f - P\|_{\infty, I_h(x)} \]
\[ \lesssim \|f\|_{\infty, I} \left(\inf_{P \in \text{Poly}_m} \|f - P\|_{\infty, I_h(x)} \right) \]

where \( \|f\|_{\infty, I} \triangleq \text{ess sup}\{|f(x)| : x \in I\} \). Since this inequality holds for any \( P \), we further have

\[ |b(x)| \lesssim \inf_{P \in \text{Poly}_m} \|f - P\|_{\infty, I_h(x)} \]

where \( \text{Poly}_m \) denotes the class of all polynomials of degree at most \( m \). Hence, to control the bias, it suffices to find a good polynomial approximation of \( f \) in the interval \( I_h(x) \).

Due to the definition of the Hölder ball, a natural choice of the polynomial is the Taylor expansion polynomial, i.e.,

\[ P(y) = \sum_{k=0}^{m} \frac{f^{(k)}(x)}{k!}(y - x)^k. \]
By Taylor’s theorem and the Lagrange remainder term, for any \( y \in I_h(x) \) we have

\[
|f(y) - P(y)| = \left| f(y) - \sum_{k=0}^{m} \frac{f^{(k)}(x)}{k!} (y - x)^k \right|
\]

(25)

\[
= \left| \frac{f^{(m)}(\xi) - f^{(m)}(x)}{m!} \cdot |y - x|^m \right|
\]

(26)

\[
\leq \frac{L|\xi - x|^m}{m!} \cdot |y - x|^m
\]

(27)

\[
\leq \frac{Lh^m}{m!} \cdot h^m
\]

(28)

\[
\lesssim nh^s
\]

(29)

where again we suppress the dependence on \( L \). As a result, we have \( |b(x)| \lesssim h^s \) for any \( x \in [0, 1] \), and thus

\[
\|b\|_q \lesssim h^s.
\]

(30)

Combining the bias and variance together, we have

\[
\sup_{f \in \mathcal{H}^s(L)} \mathbb{E}_f \| \hat{f} - f \|_q \lesssim \frac{1}{\sqrt{n}} + h^s.
\]

(31)

Setting these two terms to be equal yields \( h \asymp n^{-\frac{1}{q+s}} \), and thus

\[
\sup_{f \in \mathcal{H}^s(L)} \mathbb{E}_f \| \hat{f} - f \|_q \lesssim n^{-\frac{1}{q+s}}.
\]

(32)

In fact, this bound turns out to be optimal:

**Theorem 3.** For any \( s > 0 \) and \( q \in [1, \infty) \), the minimax \( L_q \) risk in estimating \( f \) is given by

\[
\inf_{\hat{f}} \sup_{f \in \mathcal{H}^s(L)} \mathbb{E}_f \| \hat{f} - f \|_q \asymp n^{-\frac{1}{q+s}}.
\]

(33)

Moreover, the kernel-based estimator in (10) with bandwidth \( h \asymp n^{-\frac{1}{q+s}} \) attains the minimax risk within multiplicative constants.

### 3.4 Discussions

We briefly discuss the previous result. The most important point is on the bias–variance tradeoff: when the bandwidth \( h \) increases, the deterministic error \( b(x) \) gets larger and the stochastic error \( s(x) \) gets smaller. Specifically, we have

\[
\mathbb{E}|s(x)|^q \lesssim \frac{1}{\sqrt{n}} + h^s,
\]

(34)

\[
|b(x)| \lesssim \inf_{P \in \text{Poly}_m} \| f - P \|_{\infty, I_h(x)}.
\]

(35)

The reason is also straightforward intuitively: as the bandwidth \( h \) increases, the error incurred by using \( f(x \pm h) \) to approximate \( f(x) \) gets larger, i.e., the bias gets larger. In contrast, the number of the points to be averaged over (i.e., \( 2nh \)) gets larger, which yields to a smaller variance. Finally, we need to choose a suitable bandwidth \( h^* \) to balance the bias and variance, as is shown in the following figure 1.

The second point to note is that the overall estimator \( \hat{f} \) is linear in the observations \( y_1, \cdots, y_n \). This is called a *linear* estimator. Theorem 3 shows that, for nonparametric function estimation over Hölder balls, linear estimator can attain the minimax risk within constants. Then a natural question arises: can linear
approaches always attain the minimax risk in nonparametric function estimation for other natural function classes \( \mathcal{F} \)? We will show in the next section that the answer is no in general.

The third point is more technical and high-level: in choosing the weight function, we require that our weight should keep all polynomials of degree at most \( m \). Some more thoughts lead to the following question: why are polynomials so special here? Can we choose other basis functions? To answer this question, we need to introduce the Kolmogorov \( n \)-width:

**Definition 4 (Kolmogorov \( n \)-width).** Let \((X, \| \cdot \|)\) be a normed vector space, and \( K \subset X \) be a compact subset. The Kolmogorov \( n \)-width of \( K \) is defined by

\[
d_n(K) \triangleq d_n(K, \| \cdot \|) \triangleq \inf_{V \subset X} \sup_{x \in K} \inf_{y \in V} \| x - y \|
\]

where the first infimum is taken over all linear subspaces \( V \subset X \) of dimension at most \( n \).

The following lemma shows the Kolmogorov \( n \)-width of the Hölder ball:

**Lemma 5.** For the Hölder ball \( \mathcal{H}^s(L) \) with smoothness parameter \( s > 0 \), its Kolmogorov \( n \)-width is

\[
d_n(\mathcal{H}^s(L)) \asymp n^{-s}
\]

and is attained by the polynomial basis \( V = \text{span}\{1, x, \cdots, x^{n-1}\} \).

Now we take a look at the bias (35), which shows that the bias is determined by the approximation error of the basis functions we choose. As a result, we should choose the basis functions which attain the Kolmogorov \( n \)-width, and by the previous lemma we know that polynomial basis is the right basis to use here. In general, we would like to remark another important point here: the choice of basis is very important!! We will see more examples in the next lecture.
4 Function Estimation in Sobolev Balls

Next we consider the same regression problem in Sobolev balls, which is a generalization of the Hölder ball in the sense that some spatial inhomogeneity is allowed here. We will show that in certain scenarios, any linear approach fails to give the optimal minimax risk.

4.1 Introduction to Sobolev Balls

Sobolev space is another space which naturally measures the smoothness of a function, and plays an important role in approximation theory and partial differential equations. Specifically, the (1D) Sobolev ball $W^{k,p}_1(L)$ consists of all functions $f$ on $[0,1]$ such that

$$
\|f^{(k)}\|_p \leq L
$$

where $k > 0$ is an integer, and $p \in [1, \infty]$. Technically, we remark that the derivative is defined in terms of distributions (not the classical derivative) and exists for any function. Moreover, to ensure a continuous embedding $W^{k,p}_1(L) \subset C[0,1]$, we need an additional assumption $k > \frac{1}{p}$. To see a counterexample, consider $f(x) = 1(x \geq \frac{1}{2})$, we have $f'(x) = \delta(x - \frac{1}{2})$ with $\|f'\|_1 = 1$, so $f \in W^{1,1}_1(1)$ but $f$ is not continuous. We also note that the “effective” smoothness $s$ of $W^{k,p}_1(L)$ will become

$$
s = k - \frac{1}{p}.
$$

4.2 Performance Analysis of Linear Approaches

Let’s consider the performance of the linear approach in (10), where the weight function keeps all polynomial of degree at most $k - 1$. By the bias-variance tradeoff, it suffices to deal with these two quantities separately.

For the variance, (34) does not depend on the specific function class $F$, so we still have

$$
(\mathbb{E}\|s\|_q^q) \lesssim \frac{1}{\sqrt{nh}}.
$$

For the bias, the polynomial approximation error incurred by the Taylor expansion polynomial will change for the Sobolev ball. In fact, by the same argument, for any $y \in I_h(x)$ we have

$$
|f(y) - P(y)| \lesssim h^{k-1} \int_x^{\xi} |f^{(k)}(z)| dz \leq h^{k-1} \left( \int_x^{\xi} |f^{(k)}(z)|^p dz \right)^{\frac{1}{p}} \left( \int_x^{\xi} 1^q dz \right)^{1 - \frac{1}{q}}
$$

As a result, we have

$$
\|b\|_q^q \lesssim \int_0^1 \left( h^{k-\frac{1}{p}} \cdot \left( \int_{I_h(x)} |f^{(k)}(z)|^p dz \right)^{\frac{1}{p}} \right)^q dx\]
$$
To upper bound this quantity, we need to employ the Sobolev ball condition. Note that without the exponent \( q/p \), we have
\[
\int_0^1 \left( \int_{h(x)} |f^{(k)}(z)|^p dz \right) dx = \int \int_{|x-z| \leq h} |f^{(k)}(z)|^p dx dz \leq 2h \int_0^1 |f^{(k)}(z)|^p dz \leq 2h L^p \lesssim h. \tag{47}
\]
In other words, we would like to find the maximum of some integral with exponent \( q/p \) given that this integral without the exponent is a fixed constant. Recall the following fact:

**Exercise 6.** For non-negative reals \( a_1, \cdots, a_n \), show that
\[
a_1 r + a_2 r + \cdots + a_n r \leq \begin{cases} n^{1-r}(a_1 + a_2 + \cdots + a_n)^r & \text{if } r \in [0, 1], \\ (a_1 + a_2 + \cdots + a_n)^r & \text{if } r > 1. \end{cases} \tag{48}
\]

Using this fact (and after some analysis), we can obtain that
\[
\|b\|_{q} \lesssim \begin{cases} h^{kq} & \text{if } q \leq p, \\ h^{(k-\frac{1}{p}+\frac{1}{q})q} & \text{if } p < q < \infty. \end{cases} \tag{49}
\]

Now upon choosing the optimal bandwidth \( h \), we arrive at
\[
\sup_{f \in W^{k,p}_1(L)} \mathbb{E}_f \| \hat{f} - f \|_q \lesssim \begin{cases} n^{-\frac{k}{p+k}} & \text{if } q \leq p, \\ n^{-\frac{k-\frac{1}{p}+\frac{1}{q}}{2(k-\frac{1}{p}+\frac{1}{q})+1}} & \text{if } p < q < \infty. \end{cases} \tag{50}
\]

It turns out that this is the best performance we can hope for using linear approaches:

**Theorem 7.** For any \( p \in [1, \infty], k > \frac{1}{p} \) and \( q \in [1, \infty) \), the minimax linear risk in estimating \( f \) over Sobolev ball \( W^{k,p}_1(L) \) is given by
\[
\inf_{\hat{f} \in W^{k,p}_1(L)} \sup_{f \in W^{k,p}_1(L)} \mathbb{E}_f \| \hat{f} - f \|_q \lesssim \begin{cases} n^{-\frac{k}{p}} & \text{if } q \leq p, \\ n^{-\frac{k-\frac{1}{p}+\frac{1}{q}}{2(k-\frac{1}{p}+\frac{1}{q})+1}} & \text{if } p < q < \infty. \end{cases} \tag{51}
\]

where the infimum is taken over all linear estimators.

### 4.3 Optimal Performance

One may wonder whether the performance of the best linear estimator in Theorem 7 is optimal even if we allow for non-linear estimators. The answer is yes when \( q \leq p \), and is no otherwise. This can be seen intuitively with the help of the previous exercise:

1. When \( q \leq p \), the bias-maximizing function \( f \) is non-zero everywhere in \([0, 1]\). In this case, the function \( f \) is almost homogeneous, and it is fine to use the same bandwidth \( h \) everywhere. This is called the “regular” regime.

2. When \( p < q < \infty \), the bias-maximizing function \( f \) is only supported on some interval of length \( h \), and vanishes outside. As a result, if we knew this supporting interval, we can simply set the bandwidth outside this interval to be \( \Theta(1) \) (otherwise we are accumulating noise but no signal!), and the variance becomes
\[
(\mathbb{E} \| s \|_q^2)^{\frac{1}{2}} \lesssim \frac{1}{\sqrt{n h}} \cdot h^{\frac{1}{2}}, \tag{52}
\]
which is smaller than the case where we use the same bandwidth everywhere. Combining the bias bound \( \|b\|_q \lesssim h^{k - \frac{1}{p} + \frac{1}{q}} \), the optimal bandwidth will result in the error

\[
\sup_{f \in W^{k,p}_1(L)} \mathbb{E}_f \|\hat{f} - f\|_q \lesssim n^{-\frac{k - \frac{1}{p} + \frac{1}{q}}{2(k - \frac{1}{p} + \frac{1}{q}) + 1}}. \tag{53}
\]

This is called the “sparse” regime.

The previous insights explain the reason why linear approaches cannot always obtain the minimax risk, and remark that different bandwidths should be used at different areas. The exact minimax risk is given in the following theorem:

**Theorem 8.** For any \( p \in [1, \infty] \), \( k > \frac{1}{p} \) and \( q \in [1, \infty] \), the minimax risk in estimating \( f \) over Sobolev ball \( W^{k,p}_1(L) \) is given by

\[
\inf_{\hat{f}} \sup_{f \in W^{k,p}_1(L)} \mathbb{E}_f \|\hat{f} - f\|_q \approx \begin{cases} n^{-\frac{k - \frac{1}{p} + \frac{1}{q}}{2(k - \frac{1}{p} + \frac{1}{q}) + 1}} & \text{if } q \leq (2k + 1)p, \\ \left(\frac{n}{\log n}\right) - \frac{k - \frac{1}{p} + \frac{1}{q}}{2(k - \frac{1}{p} + \frac{1}{q}) + 1} & \text{if } (2k + 1)p < q \leq \infty. \end{cases} \tag{54}
\]

where the infimum is taken over all possible estimators.

5 **Adaptive Bandwidth: Lepski’s Trick**

As is shown in the previous section, we need an adaptive bandwidth when:

1. there is spatial homogeneity: the function \( f \) may behave differently at different points, and we should choose a suitable bandwidth to achieve the bias-variance tradeoff “locally”;

2. some parameters, e.g., \( s, k, p \), are unknown: previously our choice of the optimal bandwidth relies on the parameters \( s, k, p \). If these parameters are unknown, we need to select a good bandwidth based on empirical data we have seen.

Let’s first take a look at what we have already known. For any \( x \in [0, 1] \), consider the linear estimator \( \hat{f}_h \) in (10) with bandwidth \( h = h(x) \) to estimate \( f(x) \). Denote by \( b_h(x) \), \( s_h(x) \) the corresponding bias and variance of this estimator with bandwidth \( h \), then we would like to find some optimal bandwidth \( h^* = h^*(x) \) to balance the bias and variance\(^1\):

\[
b_h^*(x) = s_h^*(x). \tag{55}
\]

The problem is that we do not know anything about the LHS other than the monotonicity of \( b_h(x) \) in \( h \), and for the RHS we also only know that \( |s_h(x)| \lesssim \sqrt{\frac{\log n}{nh}} \) with overwhelming probability. Let’s just define

\[
s_h(x) \triangleq \Theta\left(\sqrt{\frac{\log n}{nh}}\right). \tag{56}
\]

\(^1\)strictly speaking, \( b_h(x) \), \( s_h(x) \) are not the exact deterministic and stochastic errors, but suitable upper bounds (e.g., (35), (34)) of these quantities.
5.1 Lepski’s Selection Rule

Surprisingly, the monotonicity of the bias suffices to provide the knowledge to choose the bandwidth adaptively. Here is Lepski’s selection rule: for any \( x \in [0, 1] \), define the set

\[
A(x) \triangleq \{ h \geq 0 : |\hat{f}_h(x) - \hat{f}_{h'}(x)| \leq 4 s_{h'}(x), \quad \forall h' \in (0, h) \}
\]

consisting of “admissible” bandwidths, and then choose the bandwidth \( \hat{h}(x) \) to be

\[
\hat{h}(x) \triangleq \max A(x).
\]

Finally, the overall estimator \( \hat{f}_{\text{Lep}} \) is given by

\[
\hat{f}_{\text{Lep}}(x) = \hat{f}_{\hat{h}(x)}(x).
\]

Note that we only add a little bit “non-linearity” here: we still use a kernel-based estimator at any point, but the bandwidth differs spatially. Moreover, the overall estimator does not require the knowledge of any parameters (except for an upper bound on the smoothness parameter, for we need to fix a weight function (kernel) to keep all polynomials of degree up to the smoothness parameter).

5.2 Intuitions of Lepski’s Trick

Let’s explain intuitively why Lepski’s trick will work, i.e., we will have

\[
|\hat{f}_h(x) - f(x)| \lesssim |\hat{f}_{h^*}(x) - f(x)|.
\]

The analysis decomposes into two steps:

1. We first show that \( h^* \in A(x) \). It suffices to check the definition: when \( h < h^* \), we have

\[
|\hat{f}_{h^*}(x) - \hat{f}_h(x)| \leq |\hat{f}_{h^*}(x) - f(x)| + |\hat{f}_h(x) - f(x)|
\]

\[
\leq b_{h^*}(x) + s_{h^*}(x) + b_h(x) + s_h(x)
\]

\[
\leq s_{h^*}(x) + s_h(x) + s_h(x) + s_h(x)
\]

\[
= 4 s_h(x).
\]

Hence, \( h^* \) satisfies the condition, and \( h^* \in A(x) \).

2. Next we show that \( \hat{h} \) indeed works. Since \( \hat{h} = \max A(x) \) and \( h^* \in A(x) \), we have \( \hat{h} \geq h^* \), and by the definition of \( A(x) \) we know that

\[
|\hat{f}_h(x) - f(x)| \leq |\hat{f}_{h^*}(x) - \hat{f}_{h^*}(x)| + |\hat{f}_{h^*}(x) - f(x)|
\]

\[
\leq 4 s_{h^*}(x) + b_{h^*}(x) + s_{h^*}(x)
\]

\[
\leq 6 s_{h^*}(x)
\]

as desired.

\(^2\)strictly speaking, we should require that \( h' \) take value in a finite fine grid of \([0, 1]\) instead of the continuum.
5.3 Performance Analysis of Lepski’s Trick

The performance of Lepski’s estimator $\hat{f}^{\text{Lep}}$ is summarized in the following theorem:

**Theorem 9.** For any $p \in [1, \infty], k > \frac{1}{p}$ and $q \in [1, \infty]$, the Lepski’s estimator $\hat{f}^{\text{Lep}}$ satisfies

$$\sup_{f \in W_{k,p}^+} \mathbb{E}_f \|f^{\text{Lep}} - f\|_q \asymp \begin{cases} \left(\frac{n}{\log n}\right)^{-\frac{k}{2k + 1}} & \text{if } q \leq (2k + 1)p, \\ \left(\frac{n}{\log n}\right)^{-\frac{k - \frac{1}{2} + \frac{1}{2}}{2(k + \frac{1}{2}) + 1}} & \text{if } (2k + 1)p < q \leq \infty. \end{cases}$$

Compared with Theorem 8, we see that this estimator is optimal in estimating function over Sobolev balls within logarithmic factors. In fact, this logarithmic factor is unavoidable if we want our estimator to be adaptive for any smoothness parameters and any not-too-small subset of $[0, 1]$, and is the price we need to pay for adaptivity.
1 Recap

In the last lecture, we considered the nonparametric function estimation problem in the regression setting. In particular, we showed that the kernel-based estimator with a suitable-chosen adaptive bandwidth essentially achieves the minimax risk over Hölder and Sobolev balls. Specifically, the estimator is constructed as follows:

1. Fix some kernel (or weight function) with suitable regularity conditions (e.g., keep polynomials up to a prescribed degree), and use this kernel to construct a linear estimator \( \hat{f}_h(x) \) for any point \( x \in [0, 1] \) and bandwidth \( h > 0 \);
2. For any point \( x \in [0, 1] \), use some rule (e.g., Lepski’s trick) to choose an adaptive bandwidth \( \hat{h}(x) \);
3. Finally, for any point \( x \in [0, 1] \), use \( \hat{f}_{\hat{h}(x)}(x) \) as the estimator for \( f(x) \).

We also recall the Lepski’s trick: for any \( x \in [0, 1] \), we pick up a set consisting of “admissible” bandwidths:

\[
A = \{ h \in [0, 1] : |\hat{f}_h(x) - \hat{f}_{h'}(x)| \leq 4s_{h'}(x), \quad \forall h' \in (0, h) \} \tag{1}
\]

and then choose \( \hat{h}(x) = \max A \). We validate this choice via two steps: we show that the optimal bandwidth \( h^* \) which balances the bias and variance locally belongs to this set, and then we show that \( \hat{h} \) also works by relating \( \hat{h} \) to the unknown \( h^* \). This idea can also be generalized to high-dimensional cases with caution, and you may look at Homework 6 for details.

The main insights of the previous approach are two-fold:

1. The bias-variance tradeoff should be understood and analyzed carefully: this step determines the choice of the optimal bandwidth;
2. “A little bit” non-linearity needs to be added to the estimator: we have shown in the previous lecture that all linear approaches may fail to give the order-optimal risk in certain models, while an “almost” linear estimator with “a little bit” non-linearity can succeed. In the previous example, we almost use a linear kernel-based estimator, and the only additional non-linearity is to choose the bandwidth differently at different points.

In this lecture, we will attack the same problem using a different approach called wavelet shrinkage, where we can see the same phenomena from a different viewpoint.

2 Gaussian White Noise Model and Change of Basis

In the last lecture we have looked at the regression problem in Gaussian noise, and also remarked that the same idea can also be used in the density estimation setting, where the kernel-based estimator is called the KDE (Kernel Density Estimator). In this lecture we will look at the Gaussian White Noise Model, and remark that this is essentially the same as the previous two models.
2.1 Equivalences between Models

We call two statistical models are equivalent if they can almost simulate each other. An equivalent formulation is that, for any bounded loss function and any objective to be estimated, if there is an estimator $\hat{f}_1(x)$ for one model, we can construct another estimator $\hat{f}_2(y)$ for the other model whose estimation performance (i.e., the risk) is almost no worse than that of $\hat{f}_1(x)$ under any true parameter $\theta \in \Theta$, and vice versa. Intuitively, if two models are equivalent, for estimation purposes it suffices to look at any one of them. For a rigorous treatment, we refer interested readers to the Le Cam distance.

We will consider four different models in the context of nonparametric estimation:

1. **Regression Model**: we observe $n$ samples $\{y_i\}_{i=1}^n$ with $y_i = f(x_i) + \sigma \xi_i$ for $1 \leq i \leq n$, where $x_i = \frac{i}{n}$ and $\xi_i \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$;

2. **Gaussian White Noise Model**: we observe a process $(Y_t)_{t \in [0,1]}$ with
   \[
   Y_t = \int_0^t f(s) ds + \frac{\sigma}{\sqrt{n}} B_t, \quad t \in [0,1]
   \] (2)
   where $(B_t)_{t \in [0,1]}$ is the standard Brownian Motion on $[0,1]$. This model can also be written in the stochastic differential equation (SDE) form as $dY_t = f(t)dt + \frac{\sigma}{\sqrt{n}} dB_t$;

3. **Density Estimation Model**: we observe $n$ i.i.d samples $y_1, \cdots, y_n \overset{i.i.d}{\sim} g(\cdot)$, where the density $g$ is supported on $[0,1]$;

4. **Poisson Process Model**: we observe a Poisson process $(Y_t)_{t \in [0,1]}$ with a time-varying intensity $ng(\cdot)$, where the density function $g$ is supported on $[0,1]$.

In each model, there is some unknown function/density treated as the unknown parameter, and we observe discrete samples or a stochastic process. Suppose that the parameter space is $f \in \mathcal{F}$, where $\mathcal{F}$ is some function class possessing certain order of smoothness. The main result is that these four models are equivalent:

**Theorem 1.** Under certain technical conditions, these four models are asymptotically equivalent as $n \to \infty$, with $g = f^2, \sigma = \frac{1}{2}$ when talking about the last two models.

2.2 Change of Basis

Due to the model equivalence, in this lecture we consider the Gaussian white noise model

\[
 dY_t = f(t) dt + \frac{\sigma}{\sqrt{n}} dB_t, \quad t \in [0,1]
 \] (3)

and our target is to find an estimator $\hat{f}$ which comes close to minimizing the worst-case squared-error risk

\[
 \sup_{f \in \mathcal{F}} \mathbb{E}_f \| \hat{f} - f \|^2.
 \] (4)

Now we take a look at how the model and the loss function behave after we transform the problem into a different domain.

---

Let \((\phi_j)_{j=1}^\infty\) be an orthonormal basis of \(L^2[0,1]\), then we can represent the function \(f\) via its coefficients \(\theta = (\theta_j)_{j=1}^\infty\), where
\[
\theta_j \triangleq \int_0^1 \phi_j(t)f(t)dt.
\] (5)

Note that the restriction \(f \in \mathcal{F}\) will be transformed into the condition \(\theta \in \Theta\) for some proper parameter set \(\Theta\). Also, for any estimator \(\hat{f}\), we can also represent it by
\[
\hat{\theta}_j \triangleq \int_0^1 \phi_j(t)\hat{f}(t)dt.
\] (6)

As for the observation, we may define
\[
y_j \triangleq \int_0^1 \phi_j(t)dY_t
\] (7)

\[
= \int_0^1 \phi_j(t) \left( f(t)dt + \frac{\sigma}{\sqrt{n}}dB_t \right)
\] (8)

\[
= \int_0^1 \phi_j(t)f(t)dt + \frac{\sigma}{\sqrt{n}} \int_0^1 \phi_j(t)dB_t
\] (9)

\[
\equiv \theta_j + \epsilon \cdot \xi_j
\] (10)

where \(\epsilon \triangleq \frac{\sigma}{\sqrt{n}}\) denotes the noise level, and
\[
\xi_j \triangleq \int_0^1 \phi_j(t)dB_t.
\] (11)

**Exercise 2.** Use the orthonormality of \((\phi_j)_{j=1}^\infty\) to conclude that \(\xi_j \sim \mathcal{N}(0,1)\).

By the previous exercise, we know that the Gaussian white noise model reduces to the following **Gaussian sequence model** under the basis \((\phi_j)_{j=1}^\infty\):
\[
y_j = \theta_j + \epsilon \xi_j, \quad \theta = (\theta_1, \theta_2, \cdots) \in \Theta, \quad \epsilon = \frac{\sigma}{\sqrt{n}}, \quad \xi_j \sim \mathcal{N}(0,1).
\] (12)

Also, \(\|\hat{\theta} - \theta\|_2 = \|\hat{f} - f\|_2\) by Parseval’s identity, our target becomes to find some \(\hat{\theta} = (\hat{\theta}_1, \hat{\theta}_2, \cdots)\) which comes close to minimize
\[
\sup_{\theta \in \Theta} \mathbb{E}_\theta \|\hat{\theta} - \theta\|_2^2 = \sup_{\theta \in \Theta} \sum_{j=1}^\infty \mathbb{E}_\theta (\hat{\theta}_j - \theta_j)^2.
\] (13)

As a result, under the basis change, we operate in a Gaussian sequence model and would like to estimate the mean vector simultaneously.

### 2.3 Projection Estimator and Bias-Variance Tradeoff

First let’s see how the bias-variance tradeoff will look at after the change of basis. Since the observation itself is a natural estimator for the mean in the Gaussian model, a natural candidate to solve the Gaussian sequence model (12) is the **projection estimator**:
\[
\hat{\theta}_j = \begin{cases} 
y_j, & \text{if } 1 \leq j \leq m, \\
0, & \text{if } j > m.
\end{cases}
\] (14)
It’s easy to see that
\[ E_\theta (\hat{\theta}_j - \theta_j)^2 = \begin{cases} \epsilon^2, & \text{if } 1 \leq j \leq m, \\ \theta^2_j, & \text{if } j > m. \end{cases} \] (15)

As a result, we have
\[ E_\theta \| \hat{\theta} - \theta \|^2 = m\epsilon^2 + \sum_{j>m} \theta^2_j. \] (16)

Now we have the bias-variance tradeoff in the Gaussian sequence model:

1. Bias: the second term \( \sum_{j>m} \theta^2_j \), which originates from throwing away the data in the tail. Note that when \( m \) increases, the bias will decrease;

2. Variance: the first term \( m\epsilon^2 \), which originates from the noise in the observation sequence \( (y_j)_{j=1}^\infty \). Note that when \( m \) increases, the variance will increase.

We can think of the threshold \( m \) as the reciprocal of the bandwidth \( h \) in the kernel-based method: as we know from Fourier analysis, the bandwidth in the time domain and that in the frequency domain satisfies the uncertainty principle. Also, in this case, the relationship of the bias/variance in \( m \) corresponds to that of the bias/variance in \( h \) in the time domain. As a result, in the transformed domain, the bias-variance tradeoff depends on the cutting position of the observed sequence.

3 Besov Ball and Wavelet Basis

Before introducing how to add the non-linearity in the transformed domain, first we specify the choice of the function class \( \mathcal{F} \) and the orthonormal basis \( (\phi_j)_{j=1}^\infty \). Specifically, we will choose \( \mathcal{F} \) to be the Besov ball \( \mathcal{B}_{p,q}^s (L) \), and \( (\phi_j)_{j=1}^\infty \) to be the wavelet basis. To avoid technicality, we will treat them informally and only talk about the insights behind these concepts, and refer interested readers to the following reference:


3.1 Introduction to Besov Ball

Like the Hölder and Sobolev balls, the Besov ball is another ball which characterizes the smoothness of a function in a more delicate and complicated way. Specifically, for any \( s > 0, p, q \in [1, \infty] \), we may define a norm \( \| \cdot \|_{\mathcal{B}_{p,q}^s} \) which is somehow (informally) close to
\[ \| f \|_{\mathcal{B}_{p,q}^s} \approx \| f^{(s)} \|_p \] (17)

where \( f^{(s)} \) is the order-\( s \) derivative of \( f \). Note that \( s \) may not be an integer, but in the above definition we have a proper definition of a fractional derivative, and the parameter \( q \) also affects the definition of the derivative slightly. Intuitively, \( \| \cdot \|_{\mathcal{B}_{p,q}^s} \) is another norm which characterizes the order-\( s \) smoothness.

Naturally, the Besov ball \( \mathcal{B}_{p,q}^s (L) \) is defined by
\[ \mathcal{B}_{p,q}^s (L) \equiv \{ f : \| f \|_{\mathcal{B}_{p,q}^s} \leq L \}. \] (18)
3.2 Introduction to Wavelet Basis

The wavelet basis is an orthonormal basis which exploits the idea of multi-resolution analysis: any function is viewed from multiple resolutions. Specifically:

1. There is a father wavelet \( \phi(x) \) and a mother wavelet \( \psi(x) \) on \([0, 1]\):

   \[
   \phi_{jk}(x) \triangleq 2^\frac{j}{2} \phi(2^j x - k), \quad \psi_{jk}(x) \triangleq 2^\frac{j}{2} \psi(2^j x - k),
   \]

   with \( j \in \mathbb{N}, 0 \leq k \leq 2^j - 1 \).

   Note that \( \text{supp}(\phi_{jk}) = \text{supp}(\psi_{jk}) = [\frac{k}{2^j}, \frac{k+1}{2^j}] \), the level \( j \) characterizes the resolution \( 2^{-j} \), and the parameter \( k \) characterizes the spatial location to look at.

Example 3. The Haar wavelet is defined by \( \phi(x) = 1(x \in [0, 1]), \psi(x) = 1(x \in [0, \frac{1}{2}]) - 1(x \in [\frac{1}{2}, 1]) \). This is the first wavelet basis proposed in 1909.

Not all functions can be the father and wavelet wavelets. A crucial property (besides the orthonormality) for wavelets is that, defining

\[
V_j \triangleq \text{span}\{\phi_{jk}(x), 0 \leq k \leq 2^j - 1\}
\]

\[
W_j \triangleq \text{span}\{\psi_{jk}(x), 0 \leq k \leq 2^j - 1\}
\]

then for any \( j_0 \in \mathbb{N} \) we have

\[
L^2[0, 1] = V_{j_0} \oplus W_{j_0} \oplus W_{j_0+1} \oplus \cdots.
\]

As a result, any \( f \in L^2[0, 1] \) can be written as

\[
f(x) = \sum_{k=0}^{2^{j_0} - 1} \alpha_{j_0,k} \phi_{j_0,k}(x) + \sum_{j=j_0}^{\infty} \sum_{k=0}^{2^j - 1} \beta_{j,k} \psi_{j,k}(x)
\]

for some coefficients \( (\alpha_{j_0,k}), (\beta_{j,k}) \). The first term corresponds to the “gross information”, i.e., some information in the average sense, e.g., the average magnitude in a small interval. The second term corresponds to the “detail information”, i.e., some information related to the local change, e.g., how function oscillates in a small interval. We can view the detail in different scale/resolution, which is characterized by different levels \( j = j_0, j_0 + 1, \ldots \).

The reason why we introduce the wavelet basis is that it is the right basis for the Besov ball, as is shown in the following theorem.

Theorem 4. Under certain regularity conditions on the wavelet basis, the Besov norm \( \| \cdot \|_{B_{p,q}} \) for function is equivalent to \( \| \cdot \|_{B_{p,q}} \) for its wavelet coefficients, where

\[
\|f\|_{B_{p,q}} \triangleq \left( \sum_{k=0}^{2^{j_0} - 1} |\alpha_{j_0,k}|^p \right)^{\frac{1}{p}} + \left[ \sum_{j=j_0}^{\infty} \left( \sum_{k=0}^{2^j - 1} |\beta_{j,k}|^p \right)^{\frac{1}{p}} \right]^{\frac{q}{p}}.
\]

We can also write it in a more compact form as \( \|f\|_{B_{p,q}} = \|\alpha_{j_0}\|_p + \|2^{(s+\frac{1}{2}-\frac{1}{p})}\|\beta_j\|_p\|q, \) where the \( \ell_p \) norm is taken with respect to \( k \), and the \( \ell_q \) norm is taken with respect to \( j \).
Recall that we call two norms \((X, \| \cdot \|_1), (X, \| \cdot \|_2)\) are equivalent if there exists universal constants 
\(c_1, c_2 > 0\) such that \(c_1 \| f \|_2 \leq \| f \|_1 \leq c_2 \| f \|_2\) for any \(f \in X\). Then the following corollary is immediate:

**Corollary 5.** Defining

\[
\Theta_{p,q}^s \triangleq \{ \theta = (\alpha_{j_0,k}), (\beta_{j,k}) : \|\theta\|_{b_{p,q}} \leq 1 \},
\]

there exists some constants \(c_1, c_2 > 0\) such that

\[
c_1 \Theta_{p,q}^s \subset \mathcal{B}_{p,q}^s (L) \subset c_2 \Theta_{p,q}^s.
\]

We remark that although the form of \(\Theta_{p,q}^s\) is still quite complicated, we will make use of some crucial properties of \(\Theta_{p,q}^s\) to propose sound estimators and validate the fact that the wavelet basis is the right basis for the Besov space.

## 4 Thresholding and VisuShrink Estimator

In this section we present the thresholding idea to add the correct non-linearity, and thereby motivates the VisuShrink estimator. The reference for this section and the following ones is:


### 4.1 Ideal Truncated Estimator

Before we look into the Gaussian sequence estimation problem, first we gain some insights from the Gaussian mean estimation in the scalar case. Consider estimating the mean \(\theta \in \mathbb{R}\) in the following scalar model:

\[
y = \theta + \epsilon \xi, \quad \xi \sim \mathcal{N}(0, 1)
\]

where \(\epsilon > 0\) is known, and the only assumption we impose on \(\theta\) is that \(|\theta| \leq \tau\) for some known \(\tau\). The target is to find some estimator \(\hat{\theta}\) such that the mean squared error \(\mathbb{E}_\theta (\hat{\theta} - \theta)^2\) is small. The following insights are straightforward:

1. When \(\tau\) is large, there is essentially no restriction on the parameter \(\theta\), and it is expected that the observation itself \(\hat{\theta} = y\) should be a near-optimal estimator. In fact, if \(\tau = \infty\), the natural estimator \(\hat{\theta} = y\) is the Uniformly Minimum Variance Unbiased Estimator (UMVUE) and also minimax.

2. When \(\tau\) is small (e.g., much smaller than \(\epsilon\)), the signal \(\theta\) is almost completely obscured by the noise. In this case, \(\hat{\theta} = 0\) is expected to be a good estimate, since \(\mathbb{E}(\hat{\theta} - \theta)^2 = \theta^2 \leq \tau^2\) is really small.

Based on these insights, we expect that either \(\hat{\theta} = y\) or \(\hat{\theta} = 0\) can do a good job. As a result, we introduce the following concept of the *ideal truncated estimator*: suppose that there is a genie who knows the true parameter \(\theta\) but is restricted to use \(\hat{\theta} = y\) or \(\hat{\theta} = 0\), it is easy to see that the optimal estimator is

\[
\hat{\theta}_{ITE} = y \mathbb{1}(|\theta| \geq \epsilon)
\]

whose mean squared error is given by

\[
\mathbb{E}_\theta (\hat{\theta}_{ITE} - \theta)^2 = \min \{ \theta^2, \epsilon^2 \}.
\]

The following theorem shows that the ideal truncated estimator essentially attains the minimax risk for any \(\tau \geq 0\):
Theorem 6. For any $\tau \geq 0$, the ideal truncated estimator attains the minimax risk over $|\theta| \leq \tau$ within a multiplicative factor of 2.22:

$$\min\{\tau^2, \epsilon^2\} = \sup_{|\theta| \leq \tau} E_\theta(\hat{\theta}_{\text{ITE}} - \theta)^2 \leq 2.22 \cdot \inf_{\theta} \sup_{|\theta| \leq \tau} E_\theta(\hat{\theta} - \theta)^2. \quad (31)$$

It’s straightforward to generalize this result to the sequence case. Consider the Gaussian sequence model (12) with $\theta \in R(\tau)$, where $\tau = (\tau_1, \tau_2, \cdots)$ is a non-negative sequence and $R(\tau)$ is a orthosymmetric rectangle with one vertex $\tau$:

$$R(\tau) \triangleq \{\theta = (\theta_1, \theta_2, \cdots): |\theta_i| \leq \tau_i, \ \forall i = 1, 2, \cdots\}. \quad (32)$$

The following corollary is immediate:

Corollary 7. For any non-negative vector $\tau = (\tau_1, \tau_2, \cdots)$, for Gaussian sequence model (12) we have

$$\sum_{i=1}^{\infty} \min\{\tau^2, \epsilon^2\} = \sup_{\theta \in R(\tau)} E_\theta||\hat{\theta}_{\text{ITE}} - \theta||_2^2 \leq 2.22 \cdot \inf_{\theta} \sup_{\theta \in R(\tau)} E_\theta||\hat{\theta} - \theta||_2^2. \quad (33)$$

4.2 Soft- and Hard-Thresholding Estimator

Note that the ideal truncated estimator is not indeed an estimator since it requires the unavailable information $\theta$. However, the previous corollary motivates us to find another estimator $\hat{\theta}$ whose performance is comparable to that of $\hat{\theta}_{\text{ITE}}$, which has been shown to be nearly minimax. Specifically, we want to find an estimator $\hat{\theta} = (\hat{\theta}_1, \cdots, \hat{\theta}_m)$ such that in the Gaussian sequence model (12) of length $m$, the inequality

$$E_\theta||\hat{\theta} - \theta||_2^2 \leq \text{something} \times \left(\epsilon^2 + \sum_{i=1}^{m} \min\{\theta_i^2, \epsilon^2\}\right) \quad (34)$$

holds for any $\theta \in R^m$. Note that for minor technical reasons we need to have an additional $\epsilon^2$ term in the RHS.

Now we take a careful look at our requirement. As a sanity check, the RHS is really small when $\theta = 0$, which forces the LHS to be small as well. In other words, when the true parameter $\theta$ is the zero vector, our estimator $\hat{\theta}$ must be close to the zero vector as well. Prove the following result:

Exercise 8. For $X_1, \cdots, X_m \overset{i.i.d.}{\sim} N(0, \epsilon^2)$, we have $P(\max_{1 \leq i \leq m} |X_i| \geq \epsilon \sqrt{2 \log m}) \rightarrow 0$ as $m \rightarrow \infty$.

Using this exercise, a natural constraint on the estimator $\hat{\theta}$ can be that, $\hat{\theta} = 0$ whenever $\max_{1 \leq i \leq m} |y_i| \leq \epsilon \sqrt{2 \log m}$. This observation motivates us to do some type of thresholding: specifically, we can define the soft-thresholding and hard-thresholding functions as follows:

$$\eta_i^s(y) = \text{sign}(y) \cdot (|y| - t)_+ \quad (35)$$

$$\eta_i^h(y) = y \cdot I(|y| \geq t). \quad (36)$$

Note that these thresholding functions are close to truncation: when $|y|$ is small the functions return zero, and when $|y|$ is large the functions return something close to $y$.

By acting coordinatewisely, we may also define the soft-thresholding estimator $\hat{\theta}_i^s(y) = (\eta_1^s(y), \cdots, \eta_m^s(y))$ and the hard-thresholding estimator $\hat{\theta}_i^h(y) = (\eta_1^h(y), \cdots, \eta_m^h(y))$, and the previous analysis motivates us to choose the threshold $t$ to be roughly $\epsilon \sqrt{2 \log m}$. The following theorem shows that the oracle inequality (34) holds for thresholding estimators:

---

Theorem 9. For $t = c\sqrt{2\log m}$, the soft-thresholding estimator $\hat{\theta}_i^s(y)$ satisfies the following oracle inequality in the Gaussian sequence model (12):

$$E_\theta\|\hat{\theta}_i^s(y) - \theta\|_2^2 \leq (2\log m + 1) \left( \epsilon^2 + \sum_{i=1}^m \min\{\theta_i^2, \epsilon^2\} \right).$$

The same result holds for the hard-thresholding estimator $\hat{\theta}_i^h(y)$ with $t = c\sqrt{2\log m + \log\log m}$.

4.3 VisuShrink Estimator

Now we are about to describe the VisuShrink estimator for the Gaussian sequence model (12) with $\Theta = \Theta_{p,q}^s$. The parameters in this model are $\theta = ((\alpha_{jk}), (\beta_{jk}) : j \geq j_0, 0 \leq k \leq 2^j - 1)$, and we rewrite this vector as $\theta = (\theta_1, \theta_2, \cdots)$. The VisuShrink estimator does the following:

$$\hat{\theta}_i^\text{VISU} = \begin{cases} \eta_i^s(y_i) & \text{if } i \leq m \\ 0 & \text{if } i > m \end{cases}$$

where $m$ is a parameter to be chosen later, and the threshold $t$ is chosen to be $t = c\sqrt{2\log m}$. Note that compared with the projection estimator, the only difference is that when $i \leq m$ we replaced the raw data $y_i$ by its soft-thresholding $\eta_i^s(y_i)$. Similarly, we can do hard-thresholding as well with $t = c\sqrt{2\log m + \log\log m}$ according to Theorem 9.

Now we’re about to analyze the performance of the VisuShrink estimator. According to Theorem 9, we know that

$$E_\theta\|\hat{\theta}_i^\text{VISU} - \theta\|_2^2 \leq (2\log m + 1) \left( \epsilon^2 + \sum_{i=1}^m \min\{\theta_i, \epsilon^2\} \right) + \sum_{i>m} |\theta_i|^2$$

$$\leq (2\log m + 1) \left( \epsilon^2 + \sum_{i=1}^\infty \min\{\theta_i, \epsilon^2\} \right) + \sum_{i>m} |\theta_i|^2. \quad (40)$$

By the definition of $\Theta_{p,q}^s$, by choosing $m = e^{-A}$ for some large enough constant $A > 0$, we may have the tail bound: $\sup_{\theta \in \Theta_{p,q}^s} \sum_{i>m} |\theta_i|^2 \ll \epsilon^2$. As a result, for this choice of $m$, we have

$$E_\theta\|\hat{\theta}_i^\text{VISU} - \theta\|_2^2 \lesssim \log\left(\frac{1}{\epsilon}\right) \cdot \sum_{i=1}^\infty \min\{\theta_i, \epsilon^2\} + \epsilon^2 \log\left(\frac{1}{\epsilon}\right). \quad (41)$$

Now we come to the crucial property of the parameter set $\Theta_{p,q}^s$: it is solid and orthosymmetric. Equivalently, this means that for any $\theta \in \Theta_{p,q}^s$, the orthosymmetric hyperrectangle $R(|\theta|)$ with a vertex $\theta$ is contained again in the parameter set $\Theta_{p,q}^s$. As a result, by Corollary 7 we know that

$$E_\theta\|\hat{\theta}_i^\text{VISU} - \theta\|_2^2 \lesssim \log\left(\frac{1}{\epsilon}\right) \cdot \sum_{i=1}^\infty \min\{\theta_i, \epsilon^2\} + \epsilon^2 \log\left(\frac{1}{\epsilon}\right)$$

$$\leq 2.22 \log\left(\frac{1}{\epsilon}\right) \cdot \inf_{\theta'} \sup_\theta E_\theta \|\hat{\theta}' - \theta\|_2^2 + \epsilon^2 \log\left(\frac{1}{\epsilon}\right)$$

$$\leq 2.22 \log\left(\frac{1}{\epsilon}\right) \cdot \inf_{\theta'} \sup_{\theta'' \in \Theta_{p,q}^s} E_\theta \|\hat{\theta} - \theta''\|_2^2 + \epsilon^2 \log\left(\frac{1}{\epsilon}\right) \quad (43)$$

$$\leq 2.22 \log\left(\frac{1}{\epsilon}\right) \cdot \inf_{\theta'} \sup_{\theta'' \in \Theta_{p,q}^s} E_\theta \|\hat{\theta} - \theta''\|_2^2 + \epsilon^2 \log\left(\frac{1}{\epsilon}\right) \quad (44)$$

where in the last inequality we have used the property $R(|\theta|) \subset \Theta_{p,q}^s$. As a result, the VisuShrink estimator attains the minimax risk within a logarithmic factor.

Finally, we note the relationship between $\Theta_{p,q}^s$ and $B_{p,q}^s(L)$, and summarize the VisuShrink estimator as follows:

---

1. Fix some initial level $j_0$ (which is of the constant order) and termination level $j_\epsilon \approx \log(\frac{1}{\epsilon})$;

2. Transform the observation process $(Y_t)_{t \in [0, 1]}$ to the wavelet domain with initial level $j_0$, and obtain the corresponding $\alpha$-coefficients and $\beta$-coefficients empirically;

3. Use the following procedure to obtain new coefficient estimates:
   
   (a) For $\alpha$-coefficients (which are only at level $j_0$), keep them all;
   
   (b) For $\beta$-coefficients, for $j > j_\epsilon$ discard them all, and for $j_0 \leq j \leq j_\epsilon$ apply the thresholding estimator (either soft or hard one) with suitable threshold given in Theorem 9 to the observation vector.

4. Transform the estimated wavelet coefficients back to the function space, and obtain $\hat{f}^{\text{VISU}}$.

The property of the VisuShrink estimator $\hat{f}^{\text{VISU}}$ is summarized in the following theorem:

**Theorem 10.** \(^6\) For any $s > 0, p, q \in [1, \infty]$, the VisuShrink estimator $\hat{f}^{\text{VISU}}$ attains the minimax risk over Besov balls $\mathcal{B}_{p,q}^s(L)$ within logarithmic factors:

$$
\sup_{f \in \mathcal{B}_{p,q}^s(L)} \mathbb{E}_f \|\hat{f}^{\text{VISU}} - f\|_2^2 \lesssim \frac{1}{\epsilon} \cdot \inf_{f \in \mathcal{B}_{p,q}^s(L)} \mathbb{E}_f \|\hat{f} - f\|_2^2.
$$

4.4 Discussions

We make some discussions on the previous VisuShrink estimator.

First of all, the VisuShrink estimator is almost a linear estimator (i.e., close to the projection estimator), while there is also a little bit non-linearity here, i.e., the thresholding idea. We can think of the thresholding approach as a selector which selects the coefficients to keep in a data-dependent manner: when the empirical coefficient is large, we expect it to be useful signal and keep it; when the empirical coefficient is small, we expect it to be the noise and discard it. We can compare this idea with Lepski’s trick to deal with the sparse regime we defined in the last lecture: in the sparse regime, the signal is supported on a small interval and all others are noise. In this case, Lepski’s trick selects a large bandwidth in the noise regime to essentially neglect all noise, and the VisuShrink estimator simply selects the peak and neglects all others of the transformed signal in the wavelet domain.

Secondly, the VisuShrink estimator employs the shrinkage idea, which means that reduce the variance significantly with a little bit increase on the bias in statistics. Actually, this is where the term “shrink” in the name “VisuShrink” comes from. Specifically, compared with the projection estimator which keeps the raw observation, the thresholding idea incurs a larger bias (note that the previous one is indeed unbiased!), while reduces the variance significantly (e.g., from $\epsilon^2$ to $\min\{\epsilon^2, \theta_i^2\}$ per symbol). We remark that the variance is a main issue in the nonparametric function estimation, while we will see in the next lecture that in functional estimation problems, bias becomes dominant!

Thirdly, by inspecting the proof we find that we do not use any specific properties of $\Theta_{p,q}^s$ (which is of a complicated form) other than that it is solid and orthosymmetric. Also, we prove that the VisuShrink estimator is nearly minimax without even figuring out what the minimax risk is. These observations indicate that the geometry of the parameter set is really important, and the reason why we choose the wavelet basis for the Besov ball is that the Besov ball becomes solid and orthosymmetric in the wavelet domain! In other words, for any orthonormal basis $(\phi_i)_{i \in I}$, the VisuShrink idea still works as long as the associated parameter set $\Theta$ is solid and orthosymmetric in the transformed space. In fact, this property requires that the basis be an unconditional basis:

---

Definition 11 (Unconditional Basis). An orthonormal basis $(\phi_i)_{i \in I}$ is an unconditional basis of the real normed vector space $(X, \| \cdot \|)$ if and only if there exists a universal constant $C > 0$ such that

$$\left\| \sum_{i \in J} \epsilon_i \phi_i \right\| \leq C \left\| \sum_{i \in J} \phi_i \right\|$$

holds for any finite $J \subset I$.

The main messages are that:

1. Unconditional basis is the optimal basis in nonparametric function estimation;

2. Wavelet basis is an unconditional basis for the Besov norm $(L^2[0,1], \| \cdot \|_{B^{s,p,q}})$ (and Fourier basis is not).

Finally, we remark that by construction, the VisuShrink estimator does not require the knowledge of parameters $s, p, q, L$ and is thus an adaptive estimator. Similar to the Lepski’s estimator, the only knowledge the VisuShrink requires is an upper bound of the smoothness parameter $s$, for the termination level $j_\epsilon$ depends on this upper bound.

5 Thresholding and SureShrink Estimator

In the previous section we have validated the thresholding idea by proving an oracle inequality (Theorem 9), and use the geometry of the parameter set $\Theta_{p,q}^s$ to relate the risk of the ideal truncated estimator to the minimax risk. In this section we will validate the thresholding idea from a different viewpoint, and introduce the resulting SureShrink estimator.

5.1 Gaussian Mean Estimation over $\ell_p$ Balls with $\ell_q$ Error

Consider the Gaussian sequence estimation problem (12) with a simpler parameter set: $\Theta = \{ \theta : \| \theta \|_p \leq R \}$ is the $\ell_p$ ball. Also, instead of the mean squared error loss, we consider the $\ell_q$ loss as the general loss functions where $p \in (0, \infty], q \in [1, \infty)$. The question is that: for this simple example, which estimator is nearly minimax under different parameter configurations $(p, q, R, \epsilon)$?

We first begin with some insights. When $R$ is large (or equivalently $\epsilon$ is small), the constraint on $\theta$ is quite loose, and thus we should use an estimator close to the natural one (i.e., the empirical observation). When $R$ is small ($\epsilon$ is large), the vector is close to zero and we may directly apply a zero estimator. When $p \in (0, \infty]$ is small, we know that the parameter $\theta$ is somehow quite sparse, and thus the resulting estimator should have many zero entries. In contrast, when $p$ is large, the parameter $\theta$ can be quite dense, and the natural estimator is expected to work here.

As a result, if we treat the natural estimator $\hat{\theta} = y$ as the thresholding estimator $\hat{\theta} = \eta_s^t(y)$ with threshold $t = 0$, and the zero estimator $\hat{\theta} = 0$ as the thresholding estimator $\hat{\theta} = \eta_s^\infty(y)$ with threshold $t = \infty$, it seems that the thresholding estimator with some suitable threshold should work. This intuition turns out to be correct, which is shown in the following theorem:

Theorem 12. 7 For most parameter configurations $(p, q, R, \epsilon)$, there exists a universal constant $C_{p,q} > 0$ such that for the Gaussian sequence model (12) with $\Theta = \{ \theta : \| \theta \|_p \leq R \}$,

$$\inf_t \sup_{\theta \in \Theta} \mathbb{E}_\theta \| \eta_s^t(y) - \theta \|_q \leq C_{p,q} \cdot \inf_{\hat{\theta}} \sup_{\theta \in \Theta} \mathbb{E}_\theta \| \hat{\theta} - \theta \|_q$$

The same result also holds for the hard-thresholding estimator $\eta_s^h(y)$.

---

The implication of Theorem 12 to our case is that: although the parameter space $\Theta_{p,q}^s$ is of a very complicated form, it only involves the the combination of $\ell_p$ and $\ell_q$ norms! Hence, we expect that the thresholding idea also works in our case over $\Theta_{p,q}^s$. Specifically, we consider the following estimator:

1. Transform the observation to wavelet coefficients starting from initial level $j_0$ (of a constant order);
2. Keep all father wavelet coefficients, and for each level $j$, apply the (soft- or hard-)thresholding estimator to the mother wavelet coefficients with threshold $t_j$;
3. Transform back the coefficients into functions to yield $\hat{f}$.

We write the resulting estimator as $\hat{f}_t$, where $t = (t_{j_0}, t_{j_0+1}, \cdots)$ is the threshold sequence. Based on the previous insights and with the help of Theorem 12, the following result holds:

**Theorem 13.** For the nonparametric function estimation over Besov balls, the thresholding estimator with appropriate thresholds attains the minimax risk within a multiplicative factor:

$$\inf_{t=(t_{j_0}, t_{j_0+1}, \cdots)} \sup_{f \in B_{p,q}(L)} \mathbb{E}_f \| \hat{f}_t - f \|_2^2 \lesssim \inf_{f} \sup_{f \in B_{p,q}(L)} \mathbb{E}_f \| \hat{f} - f \|_2^2. \quad (48)$$

**5.2 SURE (Stein’s Unbiased Risk Estimate)**

The previous Theorem ensures that some thresholding estimator works, but does not specify which threshold we should choose. A na"ıve thought is that, if we could compare the performances of $\hat{f}_t$ with different $t$’s, then we should choose the one with the minimum error:

$$t^* = \arg \min_t \mathbb{E}_f \| \hat{f}_t - f \|_2^2. \quad (49)$$

However, this approach is infeasible since we do not know the true function $f$. Despite this difficulty, the good news is that we can still apply this idea and use an unbiased estimator of $\mathbb{E}_f \| \hat{f}_t - f \|_2^2$ without knowing $f$.

Now we start to illustrate the idea. Consider the Gaussian sequence model in (12) with length $m$, and fix any estimator $\hat{\theta}(y)$ of $\theta$. Note that $g(y) = \hat{\theta}(y) - y$ only depends on $y$ but not on $\theta$, and

$$\mathbb{E}_\theta \| \hat{\theta} - \theta \|_2^2 = \mathbb{E}_\theta \| g(y) + y - \theta \|_2^2 = \mathbb{E}_\theta \| g(y) \|_2^2 + \mathbb{E}_\theta \| y - \theta \|_2^2 + 2\mathbb{E}_\theta [(y - \theta)^T g(y)]. \quad (50)$$

**Exercise 14.** Prove Stein’s identity: for $X \sim \mathcal{N}(\mu, \sigma^2)$ and any (weakly) differentiable function $f$, we have $\mathbb{E}[(X - \mu)f(X)] = \sigma^2 \mathbb{E}[f'(X)]$.

By Stein’s identity, we further have

$$\mathbb{E}_\theta \| \hat{\theta} - \theta \|_2^2 = \mathbb{E}_\theta \| g(y) \|_2^2 + \mathbb{E}_\theta \| y - \theta \|_2^2 + 2\mathbb{E}_\theta [(y - \theta)^T g(y)] \quad (52)$$

$$= \mathbb{E}_\theta \| g(y) \|_2^2 + m\epsilon^2 + 2\mathbb{E}_\theta [\nabla \cdot g(y)] \quad (53)$$

$$= \mathbb{E}_\theta [(m + 2\nabla \cdot g(y))\epsilon^2 + \| g(y) \|_2^2]. \quad (54)$$

As a result, we have the following definition of the Stein’s Unbiased Risk Estimator:

**Definition 15 (SURE).** For the Gaussian sequence model in (12) with length $m$, then for any estimator $\hat{\theta}(y)$ with a weakly differentiable $g(y) \triangleq \hat{\theta}(y) - y$, the Stein’s Unbiased Risk Estimator (SURE) is defined by

$$r_{\text{SURE}}(y) \triangleq (m + 2\nabla \cdot g(y))\epsilon^2 + \| g(y) \|_2^2. \quad (55)$$

The SURE satisfies $\mathbb{E}_\theta[r_{\text{SURE}}(y)] = \mathbb{E}_\theta[\| \hat{\theta} - \theta \|_2^2]$ for any $\theta \in \Theta$.

---

5.3 The SureShrink Estimator

With the SURE in hand, we may introduce the SureShrink estimator, which chooses the threshold sequence \( t = (t_j)_{j \geq j_0} \) based on the risk estimate. Specifically, on each level \( j \geq j_0 \), we randomly divide the index set \( \{0, 1, \ldots, 2^j - 1\} \) into two halves \( I, I' \), and for the soft-thresholding we define

\[
t_I \triangleq \arg \min_{t \geq 0} \sum_{k \in I} (1 - 2 \cdot \mathbb{1}(|y_{j,k}| \leq t))t^2 + (\min\{t, |y_{j,k}|\})^2
\]

(56)

\[
t_{I'} \triangleq \arg \min_{t \geq 0} \sum_{k \in I} (1 - 2 \cdot \mathbb{1}(|y_{j,k}| \leq t))t^2 + (\min\{t, |y_{j,k}|\})^2
\]

(57)

where \((y_{j,k})_{0 \leq k < 2^j - 1}\) are the empirical wavelet coefficients on level \( j \). Then we apply the soft-thresholding estimator with threshold \( t_I \) to all \( y_{j,k} \) for \( k \in I \), and apply it with threshold \( t_{I'} \) to all \( y_{j,k} \) for \( k \in I' \). The same idea can also be applied to the hard-thresholding estimator. We remark that the random sample splitting approach is purely for technical purposes (to gain independence), which is not necessary in practice. The SureShrink estimator is defined to be \( \hat{f}_I \) with the thresholds given by the previous equation.

The theoretical performance of the SureShrink estimator is summarized in the following theorem.

**Theorem 16.** For \( s > \frac{1}{r} - \frac{1}{2} \), the SureShrink estimator \( \hat{f}^{SURE} = \hat{f}_I \) essentially performs as well as the optimal thresholding estimator, and thus attains the minimax risk over Besov balls \( \mathcal{B}^r_{p,q}(L) \) within multiplicative constants:

\[
\sup_{f \in \mathcal{B}^r_{p,q}(L)} \mathbb{E}_{f} \| \hat{f}^{SURE} - f \|_2^2 \leq (1 + o(1)) \cdot \inf_{(t_j)_{j \geq j_0} \in \mathbb{B}^r_{p,q}(L)} \sup_{f \in \mathcal{B}^r_{p,q}(L)} \mathbb{E}_{f} \| \hat{f}_I - f \|_2^2
\]

(58)

\[
\lesssim \inf_{\hat{f}} \sup_{f \in \mathcal{B}^r_{p,q}(L)} \mathbb{E}_{f} \| \hat{f} - f \|_2^2.
\]

(59)

6 Minimax \( L_r \) Risk over Besov Balls

In the previous sections we have studied the nonparametric function estimation problem over Besov balls using the mean squared error loss, where we have used the \( L_2 \) isometry to establish the transformation from the function domain to the wavelet domain. However, we remark that the \( L_2 \) isometry is not essential here: the same thresholding idea also applies to general \( L_r \) error, for \( r \in [1, \infty] \). Furthermore, note that we have shown the minimax optimality of various estimators without specifying the value of the minimax risk, for completeness we give the most general result here.

**Theorem 17.** For any \( s > \frac{1}{r} - \frac{1}{2} \), \( p, q, r \in [1, \infty] \), the minimax \( L_r \) risk in estimating the function \( f \) over Besov balls \( \mathcal{B}^r_{p,q}(L) \) is given by

\[
\left( \inf_{\hat{f}} \sup_{f \in \mathcal{B}^r_{p,q}(L)} \mathbb{E}_{f} \| \hat{f} - f \|_r \right)^\frac{1}{r} \asymp \begin{cases} (2^s \log(1/\epsilon))^{\frac{sp}{2^s+1}}(\log(1/\epsilon))^{\frac{1}{2^s+1}} & \text{if } r < (2s + 1)p \\ (2^s \log(1/\epsilon))^{\frac{sp}{2^s+1}}(\log(1/\epsilon))^{\frac{1}{2^s+1}} & \text{if } r = (2s + 1)p \\ (2^s \log(1/\epsilon))^{\frac{sp}{2^s+1}}(\log(1/\epsilon))^{\frac{1}{2^s+1}} & \text{if } r > (2s + 1)p \end{cases}
\]


In contrast, the minimax linear risk in estimating the function $f$ over Besov balls $\mathcal{B}_{p,q}^s(L)$ is given by

$$\left( \inf_{\hat{f}} \sup_{f \in \mathcal{B}_{p,q}^s(L)} \mathbb{E}_f \|\hat{f} - f\|_r \right)^{\frac{1}{r}} \times \begin{cases} \frac{(\epsilon^2)^{\frac{s}{2^r-1}}}{} & \text{if } r \leq p \\ \frac{(\epsilon^2)^{\frac{s}{2^r-1}}}{(2(\epsilon-\frac{1}{2})^{r-1}}} & \text{if } p < r < \infty \\ \frac{(\epsilon^2 \log(1/\epsilon))^{\frac{s}{2^r-1}}}{} & \text{if } r = \infty \end{cases}$$

(61)

where the infimum is taken over all possible linear estimators.
In this lecture, we formally define the two distinct problems of achieving and estimating the fundamental limit, and show that under the logarithmic loss, it is easier to estimate the fundamental limit than to achieve it.

1 The Bayes envelope

The Bayes envelope introduced in the previous lectures can be viewed as the fundamental limit of prediction. Indeed, for a specified loss function $\Lambda(x, \hat{x})$, the minimum average loss in predicting $X \sim P_X$ is given by the Bayes envelope:

\[ U(P_X) \equiv \min_{\hat{x}} E_{X \sim P_X} [\Lambda(X, \hat{x})]. \quad (1) \]

Throughout this lecture, we observe $X_1, X_2, \ldots, X_n \in X$ i.i.d. $P_X$, where $X \in X = \{1, 2, \ldots, S\}$. In other words, the alphabet size of $X$ is $|X| = S$. We denote by $M_S$ the space of probability measures on $X$. We take $\Lambda(x, \hat{x})$ to be the logarithmic loss in the sequel, in other words, we have

\[ \Lambda(x, \hat{x}) = \Lambda(x, \hat{P}) = \log \frac{1}{\hat{P}(x)}, \quad (2) \]

for any $x \in X, \hat{P} \in M_S$.

For non-negative sequences $a_\gamma, b_\gamma$, we use the notation $a_\gamma \lesssim b_\gamma$ to denote that there exists a universal constant $C$ such that $\sup_\gamma \frac{a_\gamma}{b_\gamma} \leq C$, and $a_\gamma \gtrsim b_\gamma$ is equivalent to $b_\gamma \lesssim a_\gamma$. Notation $a_\gamma \asymp b_\gamma$ is equivalent to $a_\gamma \lesssim b_\gamma$ and $b_\gamma \lesssim a_\gamma$. Notation $a_\gamma \gg b_\gamma$ means that $\liminf_\gamma \frac{a_\gamma}{b_\gamma} = \infty$, and $a_\gamma \ll b_\gamma$ is equivalent to $b_\gamma \gg a_\gamma$.

We write $a \land b = \min\{a, b\}$ and $a \lor b = \max\{a, b\}$. Moreover, $\text{poly}_K$ denotes the set of all polynomials of degree no more than $K$.

2 Achieving the fundamental limit

Given $n$ i.i.d. observations $X_1, X_2, \ldots, X_n \in X$ i.i.d. $P_X$, we would like to construct a predictor $\hat{P} = \hat{P}(X_1, X_2, \ldots, X_n)$ to predict a fresh new independent random variable $X \sim P_X$, where $X$ is independent of the training data $\{X_i\}_{i=1}^n$.

The average risk of predicting $X$ using the predictor $\hat{P}$ under the logarithmic loss is given by

\[ E_{\hat{P}} \left[ \log \frac{1}{\hat{P}(X)} \right], \quad (3) \]

where the expectation is over the randomness of $(X_1, X_2, \ldots, X_n, X) \sim P_X^{\otimes (n+1)}$. 

\[ \text{poly}_K \]
2.1 The inappropriate question of minimax risk

Since the distribution $P_X$ is unknown, we may take the minimax approach in decision theory and aim at solving the minimax risk. In other words, we aim at solving

$$\inf_{\hat{P}} \sup_{P_X \in \mathcal{M}_S} \mathbb{E}_P \left[ \log \frac{1}{P(X)} \right].$$

(4)

We now show that this question leads to a degenerate answer that may not be what we want.

**Theorem 1.** The minimax risk is given by

$$\inf_{\hat{P}} \sup_{P_X \in \mathcal{M}_S} \mathbb{E}_P \left[ \log \frac{1}{P(X)} \right] = \log(S),$$

(5)

and the minimax risk achieving $\hat{P}$ can be taken to be $U_S = (\frac{1}{S}, \frac{1}{S}, \ldots, \frac{1}{S})$, where $U_S$ is the uniform distribution on $X$.

**Proof** We first show that the minimax risk is at least $\log(S)$. Indeed, for any predictor $\hat{P} \in \mathcal{M}_S$, we have

$$\mathbb{E}_P \left[ \log \frac{1}{P(X)} \bigg| X_1, X_2, \ldots, X_n \right] = \sum_{x \in X} P_X(x) \log \frac{1}{P_X(x)} \geq \sum_{x \in X} P_X(x) \log \frac{1}{P_X(x)} = H(P_X),$$

(6)

where we used the non-negativity of the KL divergence, and $H(P_X)$ is the Shannon entropy. Taking $P_X = U_S$, we have $H(P_X) = \log S$. Taking expectations on both sides with respect to $X_1, X_2, \ldots, X_n$, we know

$$\mathbb{E}_P \left[ \log \frac{1}{P(X)} \right] \geq \log S$$

(9)

for any predictor $\hat{P}$.

On the other hand, taking $\hat{P} \equiv U_S$, we have

$$\mathbb{E}_P \left[ \log \frac{1}{P(X)} \right] = \log S,$$

(10)

which proves that the minimax risk is at most $\log S$. 

Theorem 1 shows that solving the minimax risk in prediction may lead to inappropriate answers. Indeed, the minimax optimal solutions turns out to be a degenerate answer that ignores all the training data. What we show next is that focusing on the minimax regret solves this problem in a meaningful way.

2.2 Achieving the fundamental limit: minimax regret

As we argued in the proof of Theorem 1, for any predictor $\hat{P} = \hat{P}(X_1, X_2, \ldots, X_n)$, we have

$$\mathbb{E}_P \left[ \log \frac{1}{P(X)} \right] \geq H(P_X).$$

(11)
It motivates us to define the minimax regret as follows:

\[
\inf_{\hat{P}} \sup_{P_X \in \mathcal{M}_s} \mathbb{E}_P \left[ \log \frac{1}{P(X)} \right] - H(P_X). \tag{12}
\]

We have the following algebraic manipulations for any predictor \(\hat{P}\):

\[
\mathbb{E}_P \left[ \log \frac{1}{P(X)} \right] - H(P_X) = \mathbb{E}_P \left[ \sum_{x \in \mathcal{X}} P_X(x) \log \frac{1}{P(x)} - \sum_{x \in \mathcal{X}} P_X(x) \log \frac{1}{\hat{P}_X(x)} \left| X_1, X_2, \ldots, X_n \right\} \right] \tag{13}
\]

\[
= \mathbb{E}_P \left[ \sum_{x \in \mathcal{X}} P_X(x) \log \frac{P_X(x)}{\hat{P}(x)} \right] \tag{14}
\]

\[
= \mathbb{E}_P D(P_X \| \hat{P}), \tag{15}
\]

where \(D(P \| Q) = \sum_{x \in \mathcal{X}} P(x) \log \frac{P(x)}{Q(x)}\) is the KL divergence between \(P\) and \(Q\).

In other words, solving the minimax regret of predicting a fresh new independent random variable \(X\) based on \(n\) i.i.d. training samples \(X_1, X_2, \ldots, X_n\) is equivalent to solving the problem of estimating the discrete distribution \(P_X\) under the KL divergence loss.

The minimax regret is characterized by the following theorem.

**Theorem 2.**

\[
\inf_{\hat{P}} \sup_{P_X \in \mathcal{M}_s} \mathbb{E}_P \left[ \log \frac{1}{P(X)} \right] - H(P_X) = \begin{cases} 
(1 + o(1)) \frac{S-1}{2n} \log(e) & \text{if } n \gg S \\
(1 + o(1)) \log \frac{S}{n} & \text{if } n \ll S
\end{cases} \tag{16}
\]

Moreover, if \(\limsup \frac{n}{S} \leq c \in (0, \infty)\), the minimax regret is bounded away from zero.

The predictor \(\hat{P}\) that achieves the performance above in the regime of \(n \gg S\) is:

\[
\hat{P}(x) = \frac{n(x) + \beta(n(x))}{n + \sum_{j=1}^{S} \beta(n(\tilde{x}_j))} \text{ for any } x \in \mathcal{X}, \tag{17}
\]

where

\[
n(x) = \sum_{i=1}^{n} 1(X_i = x), \tag{18}
\]

and \((X_1, X_2, \ldots, X_n)\) is the training data. Here

\[
\beta(k) = \begin{cases} 
\frac{1}{2} & \text{if } k = 0 \\
1 & \text{if } k = 1 \\
\frac{3}{4} & \text{o.w.}
\end{cases} \tag{19}
\]

The predictor \(\hat{P}\) that achieves the performance above in the regime of \(n \ll S\) is:

\[
\hat{P}(x) = \frac{n(x) + \frac{n}{S} \log \frac{S}{n}}{n + n \log \frac{S}{n}} \text{ for any } x \in \mathcal{X}. \tag{20}
\]

---


Vanishing minimax regret implies that there exists a predictor $\hat{P}$ such that its average prediction error $\mathbb{E}_P \left[ \log \frac{1}{P(X)} \right]$ on the test set approaches the fundamental limit $H(P_X)$. Theorem 2 shows that it takes at least $n \gg S$ samples to achieve vanishing minimax regret. It can be understood intuitively that one needs at least to see all the symbols at least once to be able to construct a predictor whose performance is able to approach the fundamental limit.

The minimax regret definition reflects the traditional way of understanding of the difficulty of machine learning tasks. In machine learning practice, we iteratively improve our training algorithm, and use its prediction accuracy on the test set to measure the performance of our prediction algorithm. The best performance achieved by existing schemes on the test set is usually understood as the “limit” of prediction for a specific dataset. In this context, Theorem 2 can be interpreted in the way that with $n \ll S$ samples, there does not exist any prediction algorithm based on $n$ training samples whose performance on the test set can approach the Bayes envelope in the worst case.

As we show in the next section, there exist algorithms that can estimate the fundamental limit with $n \ll S$ samples without explicitly constructing a prediction algorithm.

3 Estimating the fundamental limit

We define the problem of estimating the fundamental limit as solving the following minimax problem:

$$\inf_{\hat{H}} \sup_{P_X \in \mathcal{M}_S} \mathbb{E}_P |\hat{H} - H(P_X)|,$$

where the infimum is taken over all possible estimators $\hat{H} = \hat{H}(X_1, X_2, \ldots, X_n)$ that are functions of the empirical training data. The materials in this section are mainly taken from


3.1 The minimax rates

We have the following theorem.

**Theorem 3.** Suppose $n \gtrsim \frac{S}{\log S}$. Then,

$$\inf_{\hat{H}} \sup_{P_X \in \mathcal{M}_S} \mathbb{E}_P |\hat{H} - H(P_X)| \approx \frac{S}{n \ln n} + \frac{\ln S}{\sqrt{n}}. \tag{22}$$

Theorem 3 shows that it suffices to take $n \gg \frac{S}{\log S}$ samples to consistently estimate the fundamental limit $H(P_X)$. It is very surprising that the number of samples required is in fact sublinear in $S$: one can estimate the Shannon entropy uniformly over all $P_X \in \mathcal{M}_S$ even if one has not seen most of the symbols in the alphabet $X$ in the empirical samples.

---


3.2 Natural candidate: the empirical entropy

One of the most natural estimators for the Shannon entropy $H(P_X)$ given $n$ i.i.d. samples is the empirical entropy, which is defined as the following.

Denote the empirical distribution by $\hat{P}_n = (\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_S)$, where $\hat{p}_i = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}(X_i = i)$ is the empirical frequency of symbol $i$ in the training set. The empirical entropy is defined as $H(\hat{P}_n)$, which plugs-in the empirical distribution into the Shannon entropy functional. Intuitively, since the Shannon entropy is a continuous functional for finite alphabet distributions, and $\hat{P}_n$ converges to the true distribution $P_X$ as $n \to \infty$, the plug-in estimate $H(\hat{P}_n)$ should be a decent estimator for $H(P_X)$ if $S$ is fixed and $n \to \infty$. It is indeed true: it is only in the high dimensions that the empirical entropy starts to behave poorly as an estimate for the Shannon entropy.

We have the following theorem quantifying the performance of the empirical entropy in estimating $H(P_X)$.

**Theorem 4.** Suppose $n \geq S$. Then,

$$\sup_{P_X \in M_S} \mathbb{E}_P[H(\hat{P}_n) - H(P_X)] \geq \frac{S}{n} + \frac{\ln S}{\sqrt{n}}. \quad (23)$$

Comparing Theorem 4 and 3, it seems that the main difference is that one has improved the term $\frac{S}{n}$ to $\frac{S}{n \ln n}$ in the minimax rate-optimal entropy estimator, while keeping the second term unchanged. We now investigate where the two terms come from, and how one may construct the minimax rate-optimal estimators based on the empirical entropy.

3.3 Analysis of the empirical entropy

For any estimator $\hat{H}$, its performance in estimating $H(P_X)$ can be characterized via its bias defined as $\mathbb{E}_P \hat{H} - H(P_X)$, and the concentration of $\hat{H}$ around its expectation $\mathbb{E}_P \hat{H}$. The concentration property may be partially characterized by the variance of the estimator $\hat{H}$, namely $\text{Var}(\hat{H}) = \mathbb{E}_P (\hat{H} - \mathbb{E}_P \hat{H})^2$.

We now argue that in Theorem 4, the term $\frac{S}{n}$ comes from the bias, and the term $\frac{\ln S}{\sqrt{n}}$ comes from the variance.

Introduce the concave function $f(x) = x \ln(\frac{1}{x})$ on $[0, 1]$. It is clear that

$$H(\hat{P}_n) = \sum_{i=1}^{S} f(\hat{p}_i). \quad (24)$$

We have the following claim.

**Claim 5.** If $p_i \geq \frac{1}{n}$, then

$$0 \leq f(p_i) - \mathbb{E}_f(\hat{p}_i) \leq \frac{1}{n}. \quad (25)$$

Moreover,

$$\text{Var}(H(\hat{P}_n)) \leq \frac{(\ln(n))^2}{n} \wedge \frac{2(\ln(S) + 3)^2}{n}. \quad (26)$$

The results in Claim 5 are inspiring. It shows that the variance of the empirical entropy can be universally bounded regardless of the support size $S$. Moreover, the bias contributed by each symbol will be linearly added up together, contributing the term $\frac{S}{n}$. It is clear that in the regime of $S$ fixed and $n \to \infty$, the variance dominates, but in the high dimensions the bias dominates. Hence, the key to improving the empirical entropy would be to reduce the bias in high dimensions without incurring too much additional variance.

---


3.4 How can we improve the empirical entropy?

It has been a long journey to find the minimax rate-optimal estimators. Harris in 1975 proposed expanding $\mathbb{E}_p H(\hat{P}_n)$ using a Taylor expansion and obtained

$$\mathbb{E}_p H(\hat{P}_n) = H(P_X) - \frac{S - 1}{2n} + \frac{1}{12n^2} \left( 1 - \sum_{i=1}^{S} \frac{1}{p_i} \right) + o\left( \frac{1}{n^2} \right).$$

(27)

The Taylor expansion result looks decent in the regime where $p_i$’s are not too small. Indeed, for very small $p_i$ the remainder term $\sum_{i=1}^{S} \frac{1}{p_i}$ may be much larger than the true entropy $H(P_X)$ itself. This intuition turns out to be correct: it suffices to do a first-order bias correction using Taylor series in the regime of “not too small” $p_i$.

In general, for $n \cdot \hat{p}_n \sim \mathcal{B}(n, p)$, we may write

$$\mathbb{E}_p[f(\hat{p}_n)] = f(p) + \frac{1}{2} f''(p) \frac{p(1-p)}{n} + \mathcal{O}_p \left( \frac{1}{n^2} \right),$$

which motivates the bias correction:

$$\hat{f}^\prime = f(\hat{p}_n) - \frac{1}{2} f''(\hat{p}_n) \hat{p}_n(1 - \hat{p}_n).$$

In the entropy estimation case, we follow the bias correction above and do the following \footnote{Note that this bias correction intuition does not easily generalize to higher order corrections. For a systematic approach to do higher order bias correction with Taylor series, we refer the readers to Yanjun Han, Jiantao Jiao, Tsachy Weissman, “Minimax Rate-Optimal Estimation of Divergences between Discrete Distributions”, arXiv preprint arXiv:1605.09124 (2016).}

**Construction 6.** If the true $p_i \gtrsim \frac{\ln n}{n}$, we use $f(\hat{p}_i) + \frac{1}{2n}$ instead of $f(\hat{p}_i)$ to estimate $f(p_i)$.

Now the focus is on the small $p_i$ regime. We need to understand precisely which term contributed the $\frac{2}{n}$ bias bound. Assume for now that all $p_i \gtrsim \frac{\ln n}{n}$. We have the following manipulations:

$$H(\hat{P}_n) - H(P_X) = \sum_{i=1}^{S} f(\hat{p}_i) - f(p_i)$$

(28)

$$= \sum_{i=1}^{S} (f(\hat{p}_i) - P_K(\hat{p}_i)) - \sum_{i=1}^{S} (f(p_i) - P_K(p_i)) + \sum_{i=1}^{S} (P_K(\hat{p}_i) - P_K(p_i)),$$

(29)

where $P_K(\cdot)$ is an arbitrary polynomial with order no more than $K$.

The following two observations are crucial for the improvements of empirical entropy.

**Claim 7.** If $p_i \lesssim \frac{\ln n}{n}$, we have $\hat{p}_i \lesssim \frac{\ln n}{n}$ with probability at least $1 - \frac{1}{n^4}$.

**Claim 8.** Suppose $K \asymp \ln n$. Then for any constant $c > 0$,

$$\inf_{P_K \in \text{poly}_K} \sup_{x \in [0, \frac{\ln n}{n}]} |f(x) - P_K(x)| \asymp \frac{c}{n \ln n}.$$  

(30)

Utilizing those two claims, and conditioning on the event that all $\hat{p}_i \leq \frac{c \ln n}{n}$, $p_i \leq \frac{c \ln n}{n}$, we immediately obtain that

$$\sum_{i=1}^{S} (f(\hat{p}_i) - P_K(\hat{p}_i)) \lesssim \frac{S}{n \ln n}$$

(31)

$$\sum_{i=1}^{S} (f(p_i) - P_K(p_i)) \lesssim \frac{S}{n \ln n},$$

(32)
which implies that
\[ \left| E_P \left[ \sum_{i=1}^{S} (P_K(\hat{p}_i) - P_K(p_i)) \right] \right| \geq \frac{S}{n} \] (33)
since we already know in Claim 5 that \[ \left| E_P H(\hat{P}_n) - H(P_X) \right| \geq \frac{S}{n}. \]

Thus, we have identified the reason of the poor bias of the empirical entropy: it is because the plug-in approach in estimating the polynomial \( P_K \) incurs too much bias. Realizing this turns out to be the crucial factor that leads to the minimax rate-optimal estimator: under the multinomial model there exists unbiased estimators for any polynomial \( P_K \) whose order is no more than \( n \). Indeed, when \( X \sim B(n, p) \), for any integer \( r \in \{1, \ldots, n\} \):
\[ E \left[ \frac{X(X-1) \ldots (X-r+1)}{n(n-1) \ldots (n-r+1)} \right] = p^r. \]

We complete the construction of the minimax rate-optimal estimator by doing the following:

**Construction 9.** If the true \( p_i \lesssim \frac{\ln n}{n} \), we use the unbiased estimator of polynomial \( P_K(p_i) \) to estimate \( f(p_i) \). Here \( P_K(\cdot) \) is the best approximation polynomial of \( f(p_i) \) over the interval \( \left[ 0, \frac{c \ln n}{n} \right] \) introduced in Claim 8.

As for the last step, we need to use the Chernoff bound to show the following results on confidence intervals in the binomial model:

**Claim 10.** There exist \( c_1, c_2, c_3, c_4 \) positive real numbers such that:

- if \( \hat{p}_i \in \left[ 0, c_1 \frac{\log(n)}{n} \right] \) then \( p_i \in \left[ 0, c_2 \frac{\log(n)}{n} \right] \) with probability at least \( 1 - \frac{1}{n^4} \).
- if \( \hat{p}_i \in \left[ c_3 \frac{\log(n)}{n}, 1 \right] \) then \( p_i \in \left[ c_4 \frac{\log(n)}{n}, 1 \right] \) with probability at least \( 1 - \frac{1}{n^4} \).

There are other details needed to make the whole proof work: for example, one needs to argue that this approach does not increase the variance by too much, and also show minimax lower bounds. In practice one may also remove the constant term in \( P_K(\cdot) \) to ensure that one assigns zero to symbols that have never appeared in the training data. Thus, we have constructed a minimax rate-optimal estimator that does not require the knowledge of the support size \( S \), but behaves nearly as well as the exact minimax estimator with the knowledge of the support size \( S \).
1 Estimating the fundamental limit is easier than achieving it: other loss functions

We emphasize that it is a general phenomenon that estimating the fundamental limit is easier than achieving it. Recall the definition of the two problems of achieving and estimating the fundamental limit:

- Achieving the fundamental limit:
  \[
  \inf_{\hat{X}(X_1, \ldots, X_n)} \sup_{P_X \in \mathcal{P}} \mathbb{E}_P[\Lambda(X, \hat{X}) - U(P_X)]
  \]

- Estimating the fundamental limit:
  \[
  \inf_{\hat{U}(X_1, \ldots, X_n)} \sup_{P_X \in \mathcal{P}} \mathbb{E}_P[|\hat{U} - U(P_X)|]
  \]

Here we observe \( n \) i.i.d. samples \( X_1, X_2, \ldots, X_n \in \mathcal{X} \) with distribution \( P_X \in \mathcal{P} \), where \( \mathcal{P} \) is a collection of probability distribution on \( \mathcal{X} \). The \( U(P_X) = \arg \min_z \mathbb{E}_P[\Lambda(X, z)] \) is the Bayes envelope.

In last lecture, we show that in the case of \( \Lambda(X, \hat{x}) = \Lambda(X, \hat{P}) = \log \frac{1}{\hat{P}(X)} \) and \( \mathcal{P} = \mathcal{M}_S \), it takes \( n \gg S \) samples to achieve the fundamental limit, and \( n \gg S \log(S) \) samples to estimate the fundamental limit.

In this lecture we show that similar phenomenon happens for another widely used loss function: the Hamming loss. Suppose \( X = (Y, Z) \) where \( Y \in \mathcal{Y}, |\mathcal{Y}| = S \) and \( Z \in \{0, 1\} \). One may interpret \( Y \) as the feature, and \( Z \) as the label while casting it as a binary classification problem. Consider the Hamming loss:

\[
\Lambda(x, t) = 1_{(t(Y) \neq Z)}
\]

The Bayes envelope in this case is given by

\[
\hat{U}(X_1, \ldots, X_n) = \mathbb{E}_P[\min(\eta(Y), 1 - \eta(Y))],
\]

where \( \eta(y) = \mathbb{P}[Z = 1 | Y = y] \).

**Theorem 1.** Consider \( \mathcal{P} = \{ P_X | P_Z(0) = \frac{1}{2} \} \). Then, it requires \( n \gg S \) samples to achieve \( U(P_X) \) and \( n \gg \frac{S}{\log(S)} \) samples to estimate \( U(P_X) \).

2 Boosting the Chow-Liu algorithm with improved mutual information estimates

Graphical models provide us with efficient computational tools to conduct inference in high dimensional data with potential structure, cf. [1] and references therein. Learning the structure and parameters of graphical models from empirical data is therefore the starting point for all these applications. It has been known that exact learning of a general graphical model is NP-hard [2], and there exist tractable sub-classes among which tree graphical models are the most famous. The seminal work of Chow and Liu [3] contributed an efficient algorithm to compute the Maximum Likelihood Estimator (MLE) of tree structured graphical model based on empirical data, and constitutes one of the very few cases where the exact MLE can be solved efficiently. There are various approaches towards learning more complex structures, for which we refer the reader to [4] for a review.
Concretely, the Chow–Liu algorithm (CL) addresses the following question. Given \( n \) i.i.d. samples of a random vector \( \mathbf{X} = (X_1, X_2, \ldots, X_d) \), where \( X_i \in \mathcal{X}, |\mathcal{X}| < \infty \), we want to estimate the joint distribution of \( \mathbf{X} \). Chow and Liu [3] assumed that \( P_X \) can be factorized as:

\[
P_X = \prod_{i=1}^{d} P_{X_{m_i}\mid X_{m_{j(i)}}}, \quad 0 \leq j(i) < i,
\]

where \((m_1, m_2, \ldots, m_d)\) is an unknown permutation of integers \(1, 2, \ldots, d\), and \( P_{X_{m_i}\mid X_{m_{j(i)}}} \) is by definition equal to \( P_{X_i} \). Then, CL outputs the distribution \( P_{\hat{X}} \) that maximizes the likelihood of the observed data. Interestingly, this optimization problem can be efficiently solved after being transformed into a Maximum Weight Spanning Tree (MWST) problem, which can be solved using the Prim or Kruskal algorithm. In particular, they showed that the MLE of the tree structure boils down to the following expression:

\[
E_{\text{ML}} = \arg \max_{E_{Q}\subseteq Q \text{ is a tree}} \sum_{e \in E_{Q}} I(\hat{P}_e),
\]

where \( I(\hat{P}_e) \) is the mutual information associated with the empirical distribution of the two nodes connected via edge \( e \), and \( E_{Q} \) is the set of edges of a tree distribution \( Q \) (i.e., \( Q \) factors as a tree). In words, it suffices to first compute the empirical mutual information between any two nodes (in total \( \binom{d}{2} \) pairs), and the maximum weight spanning tree is the tree structure that maximizes the likelihood. To obtain estimates of distributions on each edge, Chow and Liu [3] simply assigned the empirical distribution while picking an arbitrary node as the tree root.

We begin by asking the following natural question:

**Question 2. Is the Chow–Liu algorithm optimal for learning tree graphical models?**

Since the Chow–Liu algorithm exactly solves the MLE, and has been widely used in many applications, its optimality seems to be tacitly assumed in much of the literature. However, a closer inspection of the statistical theory [5, 6] reveals that it is only known that the Chow–Liu algorithm performs essentially optimally when the number of samples \( n \) grows to \( \infty \), while the number of states of the tree has fixed size. Indeed, the modern theory of the maximum likelihood estimation paradigm [7] only justifies the asymptotic efficiency of MLE, without general non-asymptotic guarantees when we have finitely many samples. In contrast, various modern data-analytic applications deal with datasets that do not have the luxury of too many observations compared to the alphabet size.

To explain the insights underlying our improved algorithm, we revisit equation (2) and note that if we were to replace the empirical mutual information with the true mutual information, the output of the MWST would be the true edges of the tree. In light of this, the CL algorithm can be viewed as a “plug-in” estimator that replaces the true mutual information with an estimate of it, namely the empirical mutual information. Naturally then, it is to be expected that a better estimate of the mutual information would lead to smaller probability of error in identifying the tree. However, how bad can the empirical mutual information be as an estimate for the true mutual information? The following theorem in [8] implies that it can be highly sub-optimal in high dimensional regimes.

**Theorem 3.** Suppose we have two random variables \( X_1, X_2 \in \mathcal{X}, |\mathcal{X}| < \infty \). The minimax sample complexity in estimating the mutual information \( I(X_1; X_2) \) under mean squared error is \( \Theta(|\mathcal{X}|^2/\ln |\mathcal{X}|) \), while the sample complexity required by the empirical mutual information to be consistent is \( \Theta(|\mathcal{X}|^2) \).

In words, Theorem 3 implies that for the minimax rate-optimal estimator, it suffices to take \( n \gg |\mathcal{X}|^2/\ln |\mathcal{X}| \) samples to consistently estimate the mutual information \( I(X_1; X_2) \) for any underlying distributions. At the same time, unless \( n \gg |\mathcal{X}|^2 \), there exist distributions for which the error of the empirical mutual information would be bounded away from zero. Furthermore, when \( |\mathcal{X}| \) is large, the approach of estimating the three entropy terms in

\[
I(X_1; X_2) = H(X_1) + H(X_2) - H(X_1, X_2)
\]
separately using the minimax rate-optimal entropy estimators is minimax rate-optimal for estimating the mutual information.

Now we apply the improved mutual information estimates to improve the Chow–Liu algorithm. In the following experiment, we fix $d = 7, |X| = 300$, construct a star tree (i.e. all random variables are conditionally independent given $X_1$), and generate a random joint distribution by assigning independent Beta$(1/2, 1/2)$-distributed random variables to each entry of the marginal distribution $P_{X_1}$ and the transition probabilities $P_{X_k|X_{1:2}}, 2 \leq k \leq d$ (with normalization). Then, we increase the sample size $n$ from $10^3$ to $5.5 \times 10^4$, and for each $n$ we conduct 20 Monte Carlo simulations.

Note that the true tree has $d - 1 = 6$ edges, and any estimated set of edges will have at least one overlap with these 6 edges because the true tree is a star graph. We define the wrong-edges-ratio in this case as the number of edges different from the true set of edges divided by $d - 2 = 5$. Thus, if the wrong-edges-ratio equals one, it means that the estimated tree is maximally different from the true tree and, in the other extreme, a ratio of zero corresponds to perfect reconstruction. We compute the expected wrong-edges-ratio over 20 Monte Carlo simulations for each $n$, and the results are exhibited in Figure 1.

![Figure 1: The expected wrong-edges-ratio of our modified algorithm and the original CL algorithm for sample sizes ranging from $10^3$ to $5.5 \times 10^4$.](image)

Figure 1 reveals intriguing phase transitions for both the modified and the original CL algorithm. When we have fewer than $3 \times 10^3$ samples, both algorithms yield a wrong-edges-ratio of 1, but soon after the sample size exceeds $6 \times 10^3$, the modified CL algorithm begins to reconstruct the network perfectly, while the original CL algorithm continues to fail maximally until the sample size exceeds $47 \times 10^3$, 8 times the sample size required by the new algorithm, which we temporarily call “Modified Chow–Liu” algorithm.

**Open Question 4.** The exact sample complexity for learning tree graphical models is open. Although it is
clear that using improved mutual information estimates improves the tree learning process, it is not clear that it is the optimal way for tree learning. One may start from asking the following easier question: for any jointly distributed random variables \((X_1, X_2, X_3)\), what is the sample complexity of testing \(I(X_1; X_2) - I(X_1; X_3) \geq \epsilon_2\) against \(I(X_1; X_2) - I(X_1; X_3) \leq \epsilon_1\)? Here \(0 < \epsilon_1 < \epsilon_2\) are fixed constants.

3 Support size estimation

Given a discrete probability distribution \(P\), we want to estimate its support size:

\[
S(P) = \sum_i 1_{(p_i > 0)}
\]

under the assumption that \(\min_i p_i \geq \frac{1}{S}\). Note that this assumption immediately implies that the support size of \(P\) can be at most \(S\). The question is: how can we design the minimax rate-optimal estimator for estimating the true support size? The materials in this section come from [9].

We demonstrate in this section that using the approximation based methodology in last lecture, one can intuitively obtain the sample complexity in a systematic fashion.

It was shown in the last lecture that we can distinguish the cases \(p_i \approx \frac{\log n}{n}\) and \(p_i \gg \frac{\log n}{n}\) with overwhelming probability. It motivates us to separate the problem into two regimes:

1. The case of \(\frac{1}{S} \gg \frac{\log n}{n}\): in this case, the functional we want to estimate is a constant, and it suffices to use the plug-in approach. The resulting bias for each \(p_i\) is:

\[
|E[1(\hat{p}_i \neq 0) - 1_{p_i > 0}]| = (1 - p_i)^n \leq e^{-np_i} \leq e^{-\frac{n}{S}}. \tag{4}
\]

2. The case of \(\frac{1}{S} \lesssim \frac{\log n}{n}\): in this case, there might be some \(p_i\) that falls in the interval \(\left[0, \frac{\log n}{n}\right]\), while others fall in the interval \(\left[\frac{\log n}{n}, 1\right]\). Those \(p_i\) that fall into \(\left[\frac{\log n}{n}, 1\right]\) are handled with the MLE, and it suffices to look at the \(p_i\)'s that are in \(\left[0, \frac{\log n}{n}\right]\).

Claim 5. Suppose \(q \in (0, 1)\). Then,

\[
\inf_{P_K \in \text{poly}_K, P_K(0)=0} \sup_{x \in [q, 1]} |P_K(x) - 1| \approx \left(\frac{1 - \sqrt{q}}{1 + \sqrt{q}}\right)^K \approx e^{-K\sqrt{q}}. \tag{5}
\]

Applying Claim 5 to the scaled interval \(\left[0, \frac{\log n}{n}\right]\), where \(q = \frac{\frac{1}{S}}{\frac{\log n}{n}}\), \(K \propto \log n\), we obtain that the bias for \(p_i \in \left[\frac{1}{S}, \frac{\log n}{n}\right]\) is

\[
e^{-K\sqrt{q}} = e^{-\log n \sqrt{\frac{\log n}{S}}} = e^{-\sqrt{\frac{n\log n}{S}}}, \tag{6}
\]

which vanishes at long as \(n \gg \frac{q}{\log n}\).

Hence, we have (intuitively) shown that the bias of the minimax optimal approach should be of order

\[
S \cdot e^{-\Theta\left(\sqrt{\frac{n\log n}{S}}\right)}. \tag{7}
\]
References


Effective sample size enlargement

Let $M_S$ be the set of all possible discrete distributions with support size $S$. Consider a random variable $X$ with distribution $P \in M_S$. We desire to estimate some functional $F(P)$ from $n$ independent samples $X_1, \ldots, X_n$ of $P$. Examples of $F$ include the entropy or the $L_1$-distance from another distribution $Q \in M_S$. Let $\hat{F} = \hat{F}(X_1, \ldots, X_n)$ denote our estimator of $F$ and let $\hat{P}_n$ denote the empirical distribution as calculated from the $n$ samples. Consider the minimax and the plug-in risk functions defined as

$$R_{\text{minmax}}(F, P, n) = \inf_{\hat{F}(X_1, \ldots, X_n)} \sup_{P \in \mathcal{P}} E|\hat{F} - F(P)|$$

$$R_{\text{plug-in}}(F, P, n) = \sup_{P \in \mathcal{P}} E|F(\hat{P}_n) - F(P)|.$$ 

In the following table we compare the minmax and plug-in risk for several different functionals $F$. There are certain conditions on the parameters to make the risk expression hold, and we ignore them here for simplicity in presentation.

<table>
<thead>
<tr>
<th>$F(P)$</th>
<th>$\mathcal{P}$</th>
<th>$R_{\text{minmax}}(F, P, n)$</th>
<th>$R_{\text{plug-in}}(F, P, n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sum_{i=1}^{S} p_i \log \left( \frac{1}{p_i} \right)$</td>
<td>$M_S$</td>
<td>$\frac{S}{n \log(n)} + \frac{\log(S)}{\sqrt{n}}$</td>
<td>$\frac{S}{n} + \frac{\log(S)}{\sqrt{n}}$</td>
</tr>
<tr>
<td>$F_\alpha(P)$, $0 &lt; \alpha \leq \frac{1}{2}$</td>
<td>$M_S$</td>
<td>$\frac{S}{(n \log(n))^\alpha}$</td>
<td>$\frac{S}{n^\alpha}$</td>
</tr>
<tr>
<td>$F_\alpha(P)$, $\frac{1}{2} &lt; \alpha &lt; 1$</td>
<td>$M_S$</td>
<td>$\frac{S}{(n \log(n))^\alpha} + \frac{S^{1-\alpha}}{\sqrt{n}}$</td>
<td>$\frac{S}{n^\alpha} + \frac{S^{1-\alpha}}{\sqrt{n}}$</td>
</tr>
<tr>
<td>$F_\alpha(P)$, $1 &lt; \alpha &lt; \frac{3}{2}$</td>
<td>$M_S$</td>
<td>$\frac{S}{(n \log(n))^{-(\alpha-1)}}$</td>
<td>$\frac{S}{n^{-(\alpha-1)}}$</td>
</tr>
<tr>
<td>$F_\alpha(P)$, $\alpha \geq \frac{3}{2}$</td>
<td>$M_S$</td>
<td>$\frac{1}{\sqrt{n}}$</td>
<td>$\frac{1}{\sqrt{n}}$</td>
</tr>
<tr>
<td>$\sum_{i=1}^{S} 1(p_i \neq 0)$</td>
<td>${ P : \min_i p_i \geq \frac{1}{S} }$</td>
<td>$Se^{-\Theta\left(\max\left{\sqrt{\frac{\log(S)}{n}}, \frac{S}{n} \right}\right)}$</td>
<td>$Se^{-\Theta\left(\frac{S}{n}\right)}$</td>
</tr>
<tr>
<td>$\sum_{i=1}^{S}</td>
<td>p_i - q_i</td>
<td>$</td>
<td>$M_S$</td>
</tr>
</tbody>
</table>

In the following table we compare the minmax and plug-in risk for several different functionals of two distributions $P, Q \in M_S$ where we have $m$ samples from $p$ and $n$ samples from $q$. For the Kullback-Leibler and $\chi^2$ divergence estimators we only consider $(P, Q) \in \{(P, Q)|P, Q \in M_S, \frac{P_i Q_i}{P_i} \leq u(S)\}$ where $u(S)$ is some function of $S$. 


The Definition 1. 2 Preliminaries on approximation theory reason for this phenomenon. effective sample size enlargement term this phenomenon the Peetre’s Definition 3. Definition 2. plug-in risk to the minmax risk, we replace ϕ where ω

Example: f,t χ

Properties:
1. ωφφ+1(f, t) ≤ κ · ωφ(f, t), κ a constant
2. lim t→0+ ωφ(f, t)/tφ = 0 ⇒ f ∈ polyr−1

Example: Let f(x) = x log(x). We then have that ω2(f, t) ∼ t and ωφ(f, t) ∼ t2, r ≥ 2.

Definition 3. Peetre’s K-functional

Note: In many cases (with the notable exception of the support size estimator), in passing from the plug-in risk to the minmax risk, we replace m and n in the denominator with m log(m) and n log(n). We term this phenomenon the effective sample size enlargement. In the next section, we examine the underlying reason for this phenomenon.

2 Preliminaries on approximation theory

Definition 1. The r th symmetric difference of f ∈ C[0, 1] is

Example:

Definition 2.

ωr(f, t) = sup ||Δh,f||
0<h≤t

ωf(t, f) = sup ||Δh,φ(f)||
0<h≤t

where φ(x) = √(x(1−x)) and ||f|| = supx∈[0,1] |f(x)|.

Properties:
1. ωφφ+1(f, t) ≤ κ · ωφ(f, t), κ a constant
2. lim t→0+ ωφ(f, t)/tφ = 0 ⇒ f ∈ polyr−1

Example: Let f(x) = x log(x). We then have that ω2(f, t) ∼ t and ωφ(f, t) ∼ t2, r ≥ 2.

Definition 3. Peetre’s K-functional

Kf(f, t) = inf g (||f − g|| + tφ||g(r)||)

Kt,φ(f, t) = inf g (||f − g|| + tφ||φg(r)||)
Lemma 4. For all $f \in C[0,1]$, 

$$K_r(f, t') \asymp \omega^r(f, t)$$

$$K_r(\phi, t') \asymp \omega^r(\phi, t).$$

We present a systematic approach to bound the gap $|Ef(X) - f(\mathbb{E}X)|$ using the $K$-functional approach below.

Example: Suppose $X \in [0,1]$ is a random variable and $f \in C[0,1]$. We then have that

$$|Ef(X) - f(\mathbb{E}X)| = |E(f(X) - g(X) + g(X) - g(\mathbb{E}X) + g(\mathbb{E}X) - f(\mathbb{E}X))|$$

$$\leq 2\|f - g\| + |E(g(X) - g(\mathbb{E}X))|$$

$$\leq 2\|f - g\| + \frac{\text{var}(X)}{2}\|g''\|$$

$$\lesssim \omega^2 \left(f, \sqrt{\text{var}(X)} \frac{1}{2}\right),$$

where we have used the Taylor’s theorem $g(X) - g(\mathbb{E}X) = g(\mathbb{E}X)(X - \mathbb{E}X) + \frac{1}{2}g''(\zeta)(X - \mathbb{E}X)^2$ to create the bound $|E(g(X) - g(\mathbb{E}X))| \leq \frac{1}{2}\|g''\|\text{var}(X)$ followed by an application of lemma 4.

3 The mathematics behind the effective sample size enlargement

Theorem 5. Consider a binomial model, i.e., $\hat{p}_k \sim B(k, p)$ and let $f \in C[0,1]$. Note that

$$\mathbb{E}_p f(\hat{p}_k) = \sum_{i=0}^{k} f \left(\frac{i}{k}\right) \binom{k}{i} p^i (1-p)^{k-i}$$

is a polynomial of degree at most $k$, and is called the Bernstein approximation of the function $f(p)$ of order $k$.

1. Bernstein: For all $f \in C[0,1]$, $\|\mathbb{E}_p f(\hat{p}_k) - f(p)\| \rightarrow 0$ as $k \rightarrow \infty$.

2. Totik’94, Knoop and Zhou’94: For all $f \in C[0,1]$, $\|\mathbb{E}_p f(\hat{p}_k) - f(p)\| \asymp \omega^r_k(f, \frac{1}{\sqrt{k}})$

Theorem 6. For any $r < k$, the best approximation error of $f(p) \in C[0,1]$ is upper bounded by

$$\inf_{p_k \in \text{poly}_k} \sup_{x \in [0,1]} |f(x) - p_k(x)| \lesssim \omega^r_k(f, \frac{1}{k})$$

where $\text{poly}_k$ is the set of all polynomials of degree at most $k$ on $[0,1]$.

Comparing the two theorems above, we see that the key improvement from the Bernstein approximation to the best approximation is reflected on two points: one has changed the argument from $1/\sqrt{k}$ to $1/k$, and the modulus order has improved from 2 to an arbitrary number $r$ that is smaller than $k$. Since increasing the modulus order in various situations does not help reduce the approximation error, in most situations the $1/\sqrt{k} \Rightarrow 1/k$ phenomenon dominates, which can be seen from the following table.

<table>
<thead>
<tr>
<th>$f(p)$</th>
<th>Bernstein approximation error</th>
<th>Best approximation error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p \log \left(\frac{1}{p}\right)$</td>
<td>$\frac{1}{k}$</td>
<td>$\frac{1}{k^2}$</td>
</tr>
<tr>
<td>$p^\alpha, \alpha &gt; 0, \alpha \notin \mathbb{N}$</td>
<td>$\max \left{ \frac{1}{k}, \frac{1}{k^\alpha} \right}$</td>
<td>$\frac{1}{k^{2\alpha}}$</td>
</tr>
<tr>
<td>$0$ if $p = 0$ \ \ $1$ if $p \geq q$</td>
<td>$(1-q)^k$</td>
<td>$\left(\frac{1-\sqrt{q}}{1+\sqrt{q}}\right)^k$</td>
</tr>
<tr>
<td>$</td>
<td>p - q</td>
<td>, \ q \in [0,1]$</td>
</tr>
</tbody>
</table>
Now we present a heuristic argument of the effective sample size enlargement phenomenon. Consider the entropy estimation problem. Since the nonsmooth regime is of order \( [0, \log \frac{n}{n}] \), if we (approximately) conduct Bernstein approximation over this interval with order \( k \approx \log n \), the overall approximation error after scaling would be

\[
\frac{1}{k} \cdot \log \frac{n}{n} \approx \frac{1}{n}.
\]

However, if we use the best approximation with order \( k \approx \log n \) in this interval, we have the approximation error

\[
\frac{1}{k^2} \cdot \log \frac{n}{n} \approx \frac{1}{n \ln n},
\]

which justifies the improvement from \( n \) to \( n \log n \).

Similarly, consider the function \( p^\alpha, 0 < \alpha < 1 \). The nonsmooth regime is still \( [0, \frac{\log n}{n}] \). The Bernstein approximation error would be

\[
\frac{1}{k^\alpha} \left( \frac{\log n}{n} \right)^\alpha \approx \frac{1}{n^\alpha},
\]

and the best approximation error would be

\[
\frac{1}{k^{2\alpha}} \left( \frac{\log n}{n} \right)^\alpha \approx \frac{1}{(n \ln n)^\alpha}.
\]

It is also clear why this phenomenon fails to hold for the support size problem: for this problem the improvement is no longer \( 1/\sqrt{k} \Rightarrow 1/k \), but \( q \Rightarrow \sqrt{q} \).
Both Jackknife and bootstrap are generic methods that can be used to reduce the bias of statistical estimators. However, the traditional theory proves incapable of answering whether the bootstrap or jackknife can help reduce the bias of the empirical entropy so as to achieve the minimax rates of entropy estimation. We provide the answers in this lecture using advanced theory of approximation.

To simplify the presentation we focus on the binomial model $n\hat{p}_n \sim B(n,p)$ throughout this lecture. However, we emphasize that the methodology applies to any statistical models of interest.

1 Jackknife intuition

Let $e_{1,n}(p) = f(p) - \mathbb{E}_p f(\hat{p}_n)$, $f \in C([0;1])$ be the bias term. Construct the jackknife bias corrected estimator as

$$\hat{f}_2 = nf(\hat{p}_n) - (n-1)f(\hat{p}_{n-1})$$

We now heuristically claim that this approach can reduce bias. Suppose that

$$e_{1,n}(p) = \frac{a(p)}{n} + \frac{b(p)}{n^2} + O\left(\frac{1}{n^3}\right),$$

where $a(p), b(p)$ are unknown functions of $p$ which do no depend on $n$. We also have

$$e_{1,n-1}(p) = \frac{a(p)}{n-1} + \frac{b(p)}{(n-1)^2} + O\left(\frac{1}{(n-1)^3}\right).$$

Hence, the overall bias of $\hat{f}_2$ is:

$$f(p) - \mathbb{E}_p \hat{f}_2 = ne_{1,n}(p) - (n-1)e_{1,n-1}(p)$$

$$= \frac{b(p)}{n} - \frac{b(p)}{n-1} + O\left(\frac{1}{n^2}\right)$$

$$= -\frac{b(p)}{n(n-1)} + O\left(\frac{1}{n^2}\right),$$

which shows that the bias has been reduced to order $\frac{1}{n^2}$ instead of order $\frac{1}{n^3}$.

**Note:** although the result seems solid, it does not capture the dependence in $p$. For example, if $f(p) = p \log \frac{1}{p}$, standard Taylor expansion shows that the function $b(p)$ and the final $O\left(\frac{1}{n^2}\right)$ term contain functions of $p$ that explode to infinity as $p \to 0$. Traditional statistical theory fails to analyze the uniform bias reduction properties of the jackknife for functions such as $f(p) = p \log \left(\frac{1}{p}\right)$. This task can be done via the advanced theory of approximation.

2 Approximation-theoretic analysis of Jackknife

Recap:

- $r$-th order symmetric difference: $\Delta^r_h f(x) = \sum_{k=0}^{r} (-1)^k \binom{r}{k} f(x + (r-k)h)$

- $\omega^r_\varphi (f, t) = \sup_{0 \leq h \leq r} \| \Delta^r_{h \varphi} f(x) \|$, $\varphi(x) = \sqrt{x(1-x)}$
\( f(p) = p \log \frac{1}{p} \Rightarrow \omega_p^r(f,t) \asymp t^2 \) for fixed \( r \geq 2 \)

Let \( n_1 < n_2, \ldots < n_r = kn_1 = n \), where \( k \) is fixed and \( n_i \in \mathbb{N} \) and choose \( c_1, \ldots, c_r \) such that \( \sum_{i=1}^{r} c_i = 1 \)

and

for \( \rho \in \mathbb{Z}, 1 \leq \rho \leq r - 1 : \sum_{i=1}^{r} \frac{c_i}{n_i^{\rho}} = 0 \)

Then the estimator would be:

\[
\hat{f}_r = \sum_{i=1}^{r} c_i \hat{f}(\hat{p}_{n_i})
\]

Note that this framework incorporates the one described in section 1 with \( n_1 = n - 1, n_2 = n \), with \( c_1 = -(n-1), c_2 = n \).

**Lemma 1.** The solutions of \( c_i, 1 \leq i \leq r \) are given by

\[
c_i = \prod_{j \neq i} \frac{n_i}{n_i - n_j}, 1 \leq i \leq r.
\]

**Theorem 2.** Suppose that \( \sum_{i=1}^{r} |c_i| \leq C \), where \( C \) is a constant independent of \( n \). Fix \( r > 0 \). Then:

1. \( \| f - \mathbb{E} \hat{f}_r \|_{\infty} \leq C(\omega_p^r(f, \frac{1}{\sqrt{n}}) + n^{-r}\|f\|) \)
2. for fixed \( \alpha \in (0, 2r) : \)
   \[
   \| f - \mathbb{E} \hat{f}_r \| = O(n^{-\alpha/2}) \Leftrightarrow \omega_p^r(f,t) = O(t^{\alpha})
   \]
3. if \( f(p) = p \log \frac{1}{p} \), then \( \| f - \mathbb{E} \hat{f}_2 \| \asymp \frac{1}{n} \)

There are several interesting interpretations of this results:

1. Iterating the jackknife one more time is equivalent to raising the modulus order by 2
2. Raising the modulus order does not reduce the error for functions such as \( f(p) = p \log \frac{1}{p} \), hence doing jackknife does not improve the worst case bias of empirical entropy in terms of order
3. The traditional jackknife, which has \( n_1 = n - 1, n_2 = n \), does not satisfy the conditions of this theorem.

We now show that the reason why the traditional jackknife is not included in the previous theorem is that it does not satisfy the nice properties enlisted in it.

**Example 3.** Denote the \( r \)-th order jackknife with \( n_r = r, n_j - n_{j-1} = 1 \) as \( \hat{f}_r \). Then there exists a fixed function \( f \in C([0,1]) \) and \( \| f \| \leq 1 \), such that \( \| \mathbb{E} \hat{f}_r - f \| \gtrsim n^{r-1} \). Also, there exists \( f \in C([0;1]) \) such that \( \text{Var}(\hat{f}_2) \geq \frac{\pi^2}{6} \).

This result shows that the traditional jackknife may have very bad bias and variance properties if the function is not very smooth. However, this does not imply that the traditional jackknife does not work for functions like \( f(p) = p \log \frac{1}{p} \). Since the general theorem does not apply to this situation, one needs to carefully compute the bias of the jackknife estimator for entropy. It proves to be a highly challenging task.

We have the following conjecture:

**Conjecture 4.** Consider \( f(p) = p \log \frac{1}{p} \). We conjecture that \( \| f - \mathbb{E} \hat{f}_r \| \asymp \frac{1}{n} \) for any fixed \( r \geq 2 \). It is proved to be true when \( r = 2 \).
3 Bootstrap

The bootstrap relies on two principles: the plug-in principle and the Monte–Carlo principle.

1. Plug-in principle: if $\hat{P}$ is a good estimate of $P$ then $F(\hat{P})$ is a good estimate of $F(P)$.

2. Monte–Carlo principle: in order to compute the plug-in estimator $F(\hat{\theta})$ of $F(\theta) = \int g(X)dP_\theta$, where $X \sim P_\theta$, we sample $B$ random objects $X_1, X_2, \ldots, X_B \overset{i.i.d.}{\sim} P_\theta$, and then use

$$\frac{1}{B} \sum_{i=1}^{B} g(X_i)$$

(7)

to approximately compute $F(\hat{\theta})$.

Now we investigate the bias correction problem. Since $e_1(p) = f(p) - E_P \hat{f}(\hat{p}_n)$, the plug-in principle suggests that $e_1(\hat{p}_n) \approx e_1(p)$, which further suggests the bias corrected estimator

$$\hat{f}_2 = f(\hat{p}_n) + e_1(\hat{p}_n).$$

(8)

Note that one in general needs to use the Monte–Carlo principle to approximately compute $e_1(\hat{p}_n)$. It is also clear that one can do this correction infinitely many times. Indeed, we have

$$e_2(p) = e_1(p) - E_{\hat{p}_n} e_1(\hat{p}_n)$$

$$\hat{f}_r = f(\hat{p}_n) + \sum_{i=1}^{r-1} e_i(\hat{p}_n)$$

It begs the question: how can we understand the bias correction properties of this bootstrap approach?

1. Question 1: Does $\hat{f}_r$ converge as $r \to \infty$? The answer is YES!

2. Question 2: What does it converge to?

**Lemma 5.** $\lim_{r \to \infty} \|e_r(p) - (f(p) - L_n[f](p))\| = 0$ for any $f \in C([0; 1])$, where $L_n[f](p)$ is the unique Lagrange interpolating polynomial of $f$ at $n+1$ points $\{0, 1/n, 2/n, \ldots, 1\}$.

3. Question 3: does $L_n[f](p)$ have good approximation properties?

**Example 6** (Bernstein). For $f(p) = |p - 1/2|$, $\lim_{r \to \infty} |f(p) - L_n[f](p)| = \infty$ for any $p$ except $p = 0, p = 1/2$.

These results show that iterating the bootstrap bias correction too many times may not be a wise idea. What about doing it only a few times? We now show that it essentially has the similar performance as doing the jackknife exactly the same number of times.

**Theorem 7.** Fix $r > 0$. Then

1. $\|f(p) - E_p \hat{f}_r\|_\infty \leq C(\omega_\psi^{2r}(f, \frac{1}{\sqrt{n}}) + n^{-r}\|f\|)$

2. for fixed $\alpha \in (0, 2r]$ :

$$\|f(p) - E_p \hat{f}_r\| = O(n^{-\alpha/2}) \iff \omega_\psi^{2r}(f, t) = O(t^{\alpha})$$

3. if $f(p) = p \log \frac{1}{p}$, then $\|f - E \hat{f}_r\| \asymp \frac{1}{n}$