10-704: Information Processing and Learning **Spring 2015**

Lecture 3: January 20

Lecturer: Aarti Singh *Scribes: Giuseppe Vinci*

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

3.1 Submodularity of Entropy and Mutual information

In Lecture 2 we saw the definition of *submodularity*, consisting of three equivalent statements. In particular in this lecture we will use the following:

$f : 2^{\Omega} \to \mathbb{R}$ is *submodular* if $\forall X \subseteq Y \subseteq \Omega$ and $\forall z \notin Y$ we have

$$f(X \cup \{z\}) - f(X) \geq f(Y \cup \{z\}) - f(Y) \tag{3.1}$$

The intuition is that if f, for instance, is a *utility* function, it is submodular if the marginal utility from adding a new element $\{z\}$ to a set of goods X (this marginal utility is $f(X \cup \{z\}) - f(X)$) is larger than the one obtained when $\{z\}$ is added to a larger set $Y \supseteq X$ (this marginal utility is $f(Y \cup \{z\}) - f(Y)$). Submodularity in some way generalizes the idea of decreasing (positive) first derivative of an increasing function. In fact, submodularity is a useful property of functions in optimization problems.

In the following examples we use the same notation as above.

3.1.1 Entropy

In this case Ω is a set of random variables. We use capital letters to denote sets of random variables (i.e. $X \subset \Omega$) and lower case letters to denote individual random variables $x \in \Omega$. Note that this contrasts with the usual notation of using capital letters for random variables and lower case letters for realizations of a random variable.

To show that $H : 2^{\Omega} \to [0, \infty)$ is submodular consider:

$$\underbrace{H(X, z) - H(X)}_{= \underbrace{H(z|X)}_{\text{cond.entropy}}} \geq \underbrace{H(Y, z) - H(Y)}_{= \underbrace{H(z|Y)}_{\text{cond.entropy}}} \tag{3.2}$$

where $H(z|Y) = H(z|X \cup (Y \setminus X)) \leq H(z|X)$, since conditioning cannot increase entropy.

3.1.2 Mutual information

By Property 6 of Lecture notes 2, Section 2.3

- $I(X, \Omega) = H(X) - H(X|\Omega) = H(X)$, which is therefore submodular (as function of X)

- $I(X, \Omega \setminus X) = H(X) + H(\Omega \setminus X) - H(\Omega)$ is submodular (as function of X). In fact we have:

$$\begin{aligned} A_X &\equiv I(X \cup \{z\}, \Omega \setminus (X \cup \{z\})) - I(X, \Omega \setminus X) \\ &= H(X \cup \{z\}) + H(\Omega \setminus (X \cup \{z\})) - H(X) - H(\Omega \setminus X) \\ &= [H(X \cup \{z\}) - H(X)] + [H(\Omega \setminus (X \cup \{z\})) - H(\Omega \setminus X)] \end{aligned}$$

  and similarly for A_Y. By submodularity of entropy, we have

$$H(X \cup \{z\}) - H(X) \geq H(Y \cup \{z\}) - H(Y)$$

  and

$$H(\Omega \setminus Y) - H(\Omega \setminus (Y \cup \{z\})) \geq H(\Omega \setminus X) - H(\Omega \setminus (X \cup \{z\}))$$

  since $\Omega \setminus (Y \cup \{z\}) \subseteq \Omega \setminus (X \cup \{z\})$. Therefore $A_X \geq A_Y$.

3.1.3 Application to Machine Learning: Sensor placement problem

Suppose we want to monitor the temperature in a room by using k sensors. Let Ω be a set of random variables corresponding to specific (feasible) locations in the room to place sensors We want to choose the subset $X_k \subseteq \Omega$ with $|X_k| = k$ that would collect information about the temperature of the room at best. Let us consider two possible ways to do it:

1. "Maximum entropy subset selection problem"

$$X_k^* \in \underset{X \subseteq \Omega,\ \mathrm{card(X)=k}}{\operatorname{argmax}} I(X, \Omega) = \underset{X \subseteq \Omega,\ \mathrm{card(X)=k}}{\operatorname{argmax}} H(X) \tag{3.3}$$

   where we find the optimal solution (sensors locations) by maximizing the mutual information of X_k and the total space of sensors. Thus X_k^* is the subset with the largest uncertainty. **Disadvantages**: NP-hard; tendency to place sensors near the boundary (walls of the room). **Advantages**: submodularity of the objective function.

2. "Maximum mutual information subset selection problem"

$$X_k^* \in \underset{X \subseteq \Omega,\ \mathrm{card(X)=k}}{\operatorname{argmax}} I(X, \Omega \setminus X) \tag{3.4}$$

   Solutions to (3.4) might be better than solutions to (3.3), since we try to put sensors on locations to be most informative about the unsensed locations ($\Omega \setminus X$). **Disadvantages**: NP-hard. **Advantages**: submodularity of the objective function.

In the next section we deal with the actual maximization of submodular functions, useful to solve problems (3.3) and (3.4).

3.2 Maximizing submodular functions

Theorem 1 (Nemhauser et al. (1978)). *Let f be a function such that:*

1. *f is submodular over finite set Ω*
2. *f is monotone, i.e. $\forall X \subseteq Y \subseteq \Omega$, we have $f(Y) \geq f(X)$*

Algorithm 1 Greedy-Algorithm(Ω, f, k)

Input: A set Ω, A set function $f : 2^{\Omega} \to \mathbb{R}$, Size of subset k.
Output: A subset $A_k \subset \Omega$ of size k.

$A_0 \leftarrow \emptyset$
For $i = 1, \ldots, k$

1. for $x \in \Omega \backslash A_{i-1}$, set $\delta_x \leftarrow f(A_{i-1} \cup \{x\}) - f(A_{i-1})$
2. $x_* \leftarrow \text{argmax}_{x \in \Omega \backslash A_{i-1}} \delta_x$
3. $A_i \leftarrow A_{i-1} \cup \{x_*\}$

3. $f(\emptyset) = 0$

Let $A_k \subseteq \Omega$ be the first k elements chosen by ***Greedy-Algorithm(Ω, f, k)*** *(see Algorithm 1). Then*

$$f(A_k) \geq \left(1 - \frac{1}{e}\right) f(A_{\text{opt}})$$

where $A_{\text{opt}} = \underset{A \subseteq \Omega,\ \text{card(A)=k}}{\text{argmax}} f(A)$.

Proof. We prove the theorem by induction. Define $A_i = \{a_1, ..., a_i\}$, with $A_0 \equiv \emptyset$. We claim that $\forall 0 \leq j \leq k$

$$f(A_{\text{opt}}) - f(A_j) \leq \left(1 - \frac{1}{k}\right)^j f(A_{\text{opt}}) \tag{3.5}$$

1. At step $j = 0$ we have $f(A_{\text{opt}}) - \underbrace{f(A_0)}_{=f(\emptyset)=0} \leq f(A_{\text{opt}})$

2. Suppose (3.5) is true at step $j = i - 1$. Let $\delta_i = f(A_i) - f(A_{i-1})$. Thus

$$f(A_{\text{opt}}) - f(A_i) = f(A_{\text{opt}}) - f(A_{i-1}) - \delta_i \tag{3.6}$$

Let $A_{\text{opt}} \setminus A_{i-1} = \{x_1, ..., x_m\}, m \leq k$. We have

$$\begin{aligned}
f(A_{\text{opt}}) - f(A_{i-1}) &\leq f(A_{\text{opt}} \cup A_{i-1}) - f(A_{i-1}) \quad \text{[monotonicity]} \\
&= f(A_{i-1} \cup (A_{\text{opt}} \setminus A_{i-1})) - f(A_{i-1}) \\
&= \sum_{j=1}^{m} [f(A_{i-1} \cup \{x_1, ..., x_j\}) - f(A_{i-1} \cup \{x_1, ..., x_{j-1}\})] \\
\text{[submodularity]} &\leq \sum_{j=1}^{m} [f(A_{i-1} \cup x_j) - f(A_{i-1})] \\
&\leq \sum_{j=1}^{m} [f(A_i) - f(A_{i-1})] = m\delta_i \leq k\delta_i
\end{aligned}$$

$\Rightarrow \delta_i \geq \frac{1}{k}(f(A_{\text{opt}}) - f(A_{i-1}))$. Hence, equation (3.6) can be completed as follows

$$\begin{aligned} f(A_{\text{opt}}) - f(A_i) &= f(A_{\text{opt}}) - f(A_{i-1}) - \delta_i \\ &\leq \left(1 - \frac{1}{k}\right)(f(A_{\text{opt}}) - f(A_{i-1})) \\ &\leq \left(1 - \frac{1}{k}\right)^i f(A_{\text{opt}}) \end{aligned}$$

Therefore (3.5) holds also at step i.

3. Finally notice that $\left(1 - \frac{1}{k}\right)^k \leq \lim_{k\to\infty} = \frac{1}{e}$, which completes the proof.

□

The following theorem works under the more general assumption of "approximate monotonicity".

Theorem 2 (Krause et al. (2008)). *If condition 2 of Theorem 1 is replaced by*

2^*. $\forall X \subseteq \Omega$ *s.t.* $\text{card}(X) \leq 2k$

$$f(X) \leq f(X \cup \{z\}) + \epsilon \quad \textit{(approximate monotonicity)} \tag{3.7}$$

then $f(A_k) \geq \left(1 - \frac{1}{e}\right)(f(A_{\text{opt}}) - k\epsilon)$.

Proof. See Krause et al. (2008), Appendix. □

Clearly notice that if $\epsilon = 0$, then f is monotone.

Let's check if the assumptions of Theorem 1 are satisfied by entropy $H(X)$ and mutual information $I(X, \Omega \setminus X)$. We have

1. submodularity: $H(X)$ and $I(X, \Omega \setminus X)$ are submodular
2. monotonicity: $H(X) \leq H(Y)$, **but** $I(X, \Omega \setminus X) \not\leq I(Y, \Omega \setminus Y)$
3. $H(\emptyset) = I(\emptyset, \Omega \setminus \emptyset) = 0$

Thus for the mutual information $I(X, \Omega \setminus X)$ Theorem 1 cannot be directly applied. However Theorem 2 can be used because

$$I(X, \Omega \setminus X) \leq I(X \cup \{z\}, \Omega \setminus (X \cup \{z\})) + \epsilon \tag{3.8}$$

for $\text{card}(X)$ small enough.

3.3 Differential Entropy

Definition 3 (Differential Entropy). *Let X be a continuous random variable with pdf f. Then the differential entropy of X is defined as*

$$H(X) = -\int f(x) \ln f(x) dx$$

Notice that:

1. The differential entropy is based on the natural logarithm $\ln = \log_e$, instead of $\log_2$ as for entropy
2. The differential entropy can be negative!

Example 1. *$X \sim \text{Uniform}[0, a)$. Then $H(X) = -\int_0^a \frac{1}{a} \ln \frac{1}{a} dx = \ln a$ such that $H(X) < 0, \forall a \in (0, 1)$.*

Example 2. *$X \sim N(0, \sigma^2)$. Then the pdf is $f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$ and*

$$\begin{aligned} H(X) &= -\int_{\mathbb{R}} f(x) \ln f(x) dx \\ &= -\int_{\mathbb{R}} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \ln \left(\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \right) dx \\ &= \int_{\mathbb{R}} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \left(\frac{x^2}{2\sigma^2} + \ln(\sqrt{2\pi}\sigma) \right) dx \\ &= \frac{1}{2} + \ln(\sqrt{2\pi}\sigma) \\ &= \frac{1}{2} \ln(2\pi e \sigma^2) \end{aligned}$$

such that $H(X) < 0$ for $\sigma < \sqrt{\frac{1}{2\pi e}}$.

3.3.1 Application to Machine Learning: Clustering

Let $X = \{X_1, ..., X_n\}$ be a set of random variables $X_i \in \mathbb{R}^d$. Let $C : X \to \{1, ..., k\}$ define a cluster assignment which put each X_i into one of k classes. Denote with C_i the class assigned to X_i. An *entropy-based clustering* is performed by solving:

$$\max_C I(X, C) = \max_C H(X|C) \tag{3.9}$$

Connection to k-means clustering

We might wonder if criterion (3.9) differs from k-means clustering. Suppose $X|C = j \sim N(\mu_j, \sigma_j^2 I)$, where I is the $d \times d$ identity matrix. Thus, the differential entropy of $X|C = j$ is

$$H(X|C = j) = \frac{d}{2} \ln(2\pi e \sigma_j^2) \tag{3.10}$$

An estimator of $H(X|C = j)$ (when μ_j, σ_j are unknown) is the *plug-in* estimator

$$\hat{H}(X|C = j) = \frac{d}{2} \ln \left(2\pi e \frac{1}{n_j} \sum_{i:C_i=j} \|X_i - \hat{\mu}_j\|_2^2 \right) \tag{3.11}$$

obtained by replacing σ_j^2 with $\hat{\sigma}_j^2 = \frac{1}{n_j} \sum_{i:C_i=j} \|X_i - \hat{\mu}_j\|^2$, where $\hat{\mu}_j = \frac{1}{n_j} \sum_{i:C_i=j} X_i$ and $n_j = \text{card}(\{i : C_i = j\})$. Thus, an estimate of the conditional (differential) entropy $H(X|C)$ is

$$\hat{H}(X|C) = \sum_{j=1}^k \hat{H}(X|C = j) \underbrace{\hat{P}(C = j)}_{n_j/n} = \sum_{j=1}^k \frac{d}{2} \ln \left(2\pi e \frac{1}{n_j} \sum_{i:C_i=j} \|X_i - \hat{\mu}_j\|_2^2 \right) \frac{n_j}{n} \tag{3.12}$$

Therefore the *entropy-based clustering* (3.9) is implemented by solving

$$\min_C \hat{H}(X|C) = \min_C \sum_{j=1}^{k} \ln \left(\frac{1}{n_j} \sum_{i:C_i=j} \|X_i - \hat{\mu}_j\|_2^2 \right) n_j \tag{3.13}$$

The k-means clustering is performed by

$$\min_C \sum_{j=1}^{k} \sum_{i:C_i=j} \|X_i - \hat{\mu}_j\|_2^2 \tag{3.14}$$

We can easily see that the optimization problem of the entropy-based clustering (3.13) differs from the k-means clustering optimization problem (3.14) just because of the logarithm. In fact if ln is replaced by the identity function, (3.13) and (3.14) are equivalent. However, the logarithm makes the entropy-based clustering *more robust* than k-means.

References

Krause, Andreas, Singh, A., & Guestrin, C. (2008). Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *The Journal of Machine Learning Research*, *9*, 235-284.

Nemhauser, G. L., Wolsey, L. A., & Fisher, M. L. (1978). An analysis of approximations for maximizing submodular set functionsI. *Mathematical Programming*, *14.1*, 265-294.