6.883 Learning with Combinatorial Structure
Notes for Lecture 7
Scribe: Matthew Staib

1 Submodularity in graphical models

1.1 Log-submodularity for faster inference

Last time we looked at some graphical or structural properties of graphical models which make inference easier, such as treewidth or other properties relating to clique size. Alternatively, we could ignore the structure and instead focus on properties of the potential functions themselves. Consider a distribution of the form

$$p(x; w) \propto \prod_{C \in \mathcal{C}} \psi_C(x_C; w). \tag{1}$$

Suppose the variables x_i are binary and the density function $p(x; w)$ is log-supermodular, i.e. for all $S, T \subseteq \mathcal{V}$,

$$p(1_S)p(1_T) \leq p(1_{S \cup T})p(1_{S \cap T}). \tag{2}$$

We can then find the mode of the distribution in polynomial time by maximizing a log-supermodular function.

1.2 Checking log-submodularity

One way to get a log-supermodular density function is to have log-supermodular potential functions ψ_C, as the product of log-supermodular functions is log-supermodular.

In the case where we have pairwise potentials, i.e. $|C| \leq 2$ for all $C \in \mathcal{C}$, to check log-supermodularity, we need only check that

$$\psi'_{ij}(0,1) + \psi'_{ij}(1,0) \geq \psi'_{ij}(0,0) + \psi'_{ij}(1,1), \tag{3}$$

where $\psi'_{ij}(x_i, x_j) = -\log \psi_{ij}(x_i, x_j)$.

For maximal cliques of arbitrary size, we can instead check

$$\psi'_C(z,0,1) + \psi'_C(z,1,0) \geq \psi'_C(z,0,0) + \psi'_C(z,1,1) \tag{4}$$

for all assignments $z = x_{C \setminus i,j} \in \{0,1\}^{|C|-2}$.

2 Extensions of submodular functions

Now that we have a submodular optimization problem, it is natural to want to extend the set function $F : \{0,1\}^n \to \mathbb{R}$ [1] to a function $f : \mathbb{R}_+^n \to \mathbb{R}$, so that we can leverage techniques from continuous optimization. Ideally, the functions F and f would agree on $\{0,1\}^n$, i.e. $f(1_S) = F(S)$ for all $S \subseteq \mathcal{V}$. But how might we consistently interpret x when $x \notin \{0,1\}^n$?

2.1 Lovász extension

One idea is to write $x = \sum_{i=1}^k \alpha_i 1_{S_i}$ as a way of interpolating between some collection of subsets S_i. We could then write $f(x) = \sum_{i=1}^k \alpha_i F(S_i)$. However, there are many such possible collections of sets S_i to use here in the decomposition. We need to define a unique collection $\{S_i\}$ for each x with consistent behavior in terms of how we choose α_i.

One way to choose $\{S_i\}$ is to think of x as a bar diagram, with bars for each components corresponding to the component values. We can then think of "level sets" of x: for any $\theta \in [0,1]$, we define

$$S^\theta = \{i : x_i \geq \theta\}. \tag{5}$$

Since x has finitely many components, as we decrease θ from one down to zero, we get a sequence of sets $S_1 \subset S_2 \subset \cdots \subset S_k = \mathcal{V}$. This collection of sets is well-defined for each x, so we will use it as a decomposition for x.

We will now give two interpretations of how to decompose x (i.e. find the coefficients α_i) and thus define $f(x)$.

2.1.1 Extensions from expectations

For the moment, suppose $x \in [0,1]^n$. Let θ now be a random variable, uniform over $[0,1]$. Then, define our extension as

$$f(x) = \mathbb{E}_\theta[F(S^\theta)], \tag{6}$$

where $S^\theta = \{i : x_i \geq \theta\}$ as in 5. We can equivalently write this as

$$f(x) = \int_0^1 F(S^\theta)\, d\theta, \tag{7}$$

which is known as the Choquet integral. Note that, while we get a distribution of sets S^θ, the only sets we will ever see from this distribution are the level sets $S_1 \subset S_2 \subset \cdots \subset S_k = \mathcal{V}$ from earlier.

[1] Here, $\{0,1\}^n$ is a representation of the powerset of $\mathcal{V}$ with $|\mathcal{V}| = n$

2.1.2 Extensions from sorting x

We now give an equivalent definition of this extension which works for x not necessarily in $[0,1]^n$. Define a permutation $\pi(\cdot)$ which sorts x in descending order:

$$x_{\pi(1)} \geq x_{\pi(2)} \geq \cdots \geq x_{\pi(n)}. \tag{8}$$

Then, defining $S_i = \{\pi(1), \ldots, \pi(i)\}$ and $S_0 = \emptyset$, we can write

$$f(x) = \sum_{i=2}^{n} (x_{\pi(i-1)} - x_{\pi(i)}) F(S_i) + x_{\pi(n)} F(\mathcal{V}) \tag{9}$$

$$= \sum_{i=1}^{n} x_{\pi(i)} (F(S_i) - F(S_{i-1})). \tag{10}$$

The main takeaway from this version of the definition for the Lovász extension is that f is a linear function of x as long as the permutation stays the same, i.e. the extension $f(x)$ is piecewise linear.

2.1.3 Lovász extension examples

Example 2.1. Consider $F(S) = \min\{1, |S|\}$, which is zero for $S = \emptyset$ and one elsewhere. Then, we have

$$f(x) = \sum_i \alpha_i F(S_i) = \sum_i \alpha_i = \max_j x_j. \tag{11}$$

Example 2.2. Consider a graph with two nodes u, v and an edge between them with weight ν. Let $F(S)$ be the graph cut function, i.e.

$$F(S) = \begin{cases} \nu & \text{if } |S| = 1 \text{ (i.e. if we cut)} \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

Assume without loss of generality that $x_u > x_v$. Then,

$$f(x) = \nu(x_u - x_v) = \nu(\max_j x_j - \min_j x_j) = \nu |x_u - x_v|. \tag{13}$$

This is also called total variation, which is used heavily in signal processing.

2.2 Multilinear extension

Before further exploring the Lovász extension, we note that there are other ways to define an extension from an expectation of $F(S)$, depending on how we sample S. One specific other way is, given $x \in [0,1]^n$, sample each element i with probability x_i:

$$f_M(x) = \mathbb{E}_{S \sim x}[F(S)] \tag{14}$$

$$= \sum_{S \subseteq \mathcal{V}} F(S) \prod_{e \in S} x_e \prod_{e \notin S} (1 - x_e). \tag{15}$$

This function $f_M(x)$ is called the *multilinear extension* of $F(S)$. There are a couple of important comparisons to draw versus the Lovász extension:

1. Even when $F(S)$ is submodular, $f_M(x)$ is not convex in general. For the graph cut function (example 2.2), the associated multilinear extension is

$$f_M(x) = x_u + x_v - 2x_u x_v. \tag{16}$$

2. The Lovász extension is actually a lower bound on all possible extensions of $F(S)$ that one can get from taking expectations over $F(S)$, including the multilinear extension. Hence, $f_L(x) \leq f_M(x)$ for all x.

3 Properties of the Lovász extension

First, some basic properties of the Lovász extension $f(x)$:

1. For any $A \subseteq \mathcal{V}$, $f(1_A) = F(A)$.
2. The Lovász extension is *positively homogeneous*, meaning $f(\lambda x) = \lambda f(x)$ for all $\lambda \geq 0$.
3. Note that $1_{\mathcal{V}}$ is the all-ones vector. By looking at the Choquet integral, for $\lambda \geq 0$ we find that $f(x + \lambda 1_{\mathcal{V}}) = f(x) + \lambda F(\mathcal{V})$.
4. If $F(S)$ is *symmetric* ($F(S) = F(\mathcal{V} \setminus S)$), then $f(-x) = f(x)$.

Perhaps the most useful property of the Lovász extension is the following theorem:

Theorem 3.1. *The Lovász extension $f(x)$ is convex if and only if the original function $F(S)$ is submodular.*

Proof. First, we prove the $\implies$ direction:

Assume the Lovász extension is convex. Let $A, B \subseteq \mathcal{V}$, and note that $1_A + 1_B = 1_{A\cup B} + 1_{A\cap B}$. By convexity and then positive homogeneity (for $\lambda = 2$), we see that $f(1_A + 1_B) \leq f(1_A) + f(1_B)$. Then, by the definition of the Lovász extension,

$$\begin{aligned} F(A) + F(B) &= f(1_A) + f(1_B) \\ &\geq f(1_{A\cup B} + 1_{A\cap B}) \\ &\overset{(a)}{=} f(1_{A\cup B}) + F(A \cap B) \\ &= F(A \cup B) + F(A \cap B) \end{aligned}$$

where (a) is by looking at the sorting definition of $f(x)$. Specifically, note that the decomposition $\{S_i\}$ corresponds to $S_1 = A \cap B \subset A \cup B = S_2$. The inequality we have just proved is the definitely of a submodular function, so $F(S)$ is submodular.

Now, we prove the $\impliedby$ direction:

Assume that $F(S)$ is submodular. Then, we have (but will not prove) that

$$f(x) = \max_{y\in\mathcal{B}_F} y^T x, \tag{17}$$

where $\mathcal{B}_F$ is the so-called *base polytope*. As a maximum of convex functions over the convex set $\mathcal{B}_F$, it follows that $f(x)$ is convex. □

3.1 Submodular polyhedra and base polytopes

To help understand the meaning of (17), we will define the base polytope $\mathcal{B}_F$. First, we must define the *submodular polyhedron*:

Definition 3.1. The *submodular polyhedron* of a submodular function $F(S)$ is the polyhedron defined by

$$\mathcal{P}_F = \{y \in \mathbb{R}^n : y(A) \leq F(A) \text{ for all } A \subseteq \mathcal{V}\}.$$

Here we are writing $y(A)$ as shorthand for $\sum_{a\in A} y_a$. The set $\mathcal{P}_F$ is the set of all linear (modular) functions which are dominated by the function $F(S)$. The base polytope $\mathcal{B}_F$ is a subset of $\mathcal{P}_F$ of linear functions which are dominated by $F(S)$ but agree with $F(S)$ on the full ground set $\mathcal{V}$:

Definition 3.2. The *base polytope* of a submodular function $F(S)$ is the polyhedron defined by

$$\mathcal{B}_F = \{y \in \mathcal{P}_F : y(\mathcal{V}) = F(\mathcal{V})\}.$$

Geometrically, the base polytope is one face of the submodular polyhedron.

We will now look at some examples.

Example 3.1. Consider the function $F(A)$ for $A \subseteq \{a, b\}$, defined below:

A	$F(A)$
$\emptyset$	0
$\{a\}$	-1
$\{b\}$	2
$\{a, b\}$	0

We can write

$$\mathcal{P}_F = \{y \in \mathbb{R}^2 : y_a \leq -1,\ y_b \leq 2,\ y_a + y_b \leq 0\} \tag{18}$$

and

$$\mathcal{B}_F = \{y \in \mathbb{R}^2 : y_a \leq -1,\ y_b \leq 2,\ y_a + y_b = 0\}. \tag{19}$$

This is illustrated graphically in figure 3.1.

Example 3.2. Consider the cut function from example 2.2. This function has

$$\mathcal{P}_F = \{y \in \mathbb{R}^2 : y_u, y_v \leq \nu,\ y_u + y_v \leq 0\} \tag{20}$$

and

$$\mathcal{B}_F = \{y \in \mathbb{R}^2 : y_u, y_v \leq \nu,\ y_u + y_v = 0\} \tag{21}$$

This is illustrated graphically in figure 3.1.

Example 3.3. Consider a modular function defined by $F(S) = \sum_{i \in S} w_i$, where $S \subseteq \mathcal{V} = \{1, 2\}$. Then the submodular polyhedron is

$$\mathcal{P}_F = \{y \in \mathbb{R}^2 : y_1 \leq w_1,\ y_2 \leq w_2,\ y_1 + y_2 \leq w_1 + w_2\}. \tag{22}$$

The last constraint is actually implied by the first two, so

$$\mathcal{P}_F = \{y \in \mathbb{R}^2 : y_1 \leq w_1,\ y_2 \leq w_2\}. \tag{23}$$

The base polytope is the subset of $\mathcal{P}_F$ where the last constraint is tight, but the only way to make this constraint tight is to set $y_1 = w_1$ and $y_2 = w_2$, so that $\mathcal{B}_f = \{(w_1, w_2)\}$. This is illustrated graphically in figure 3.1.

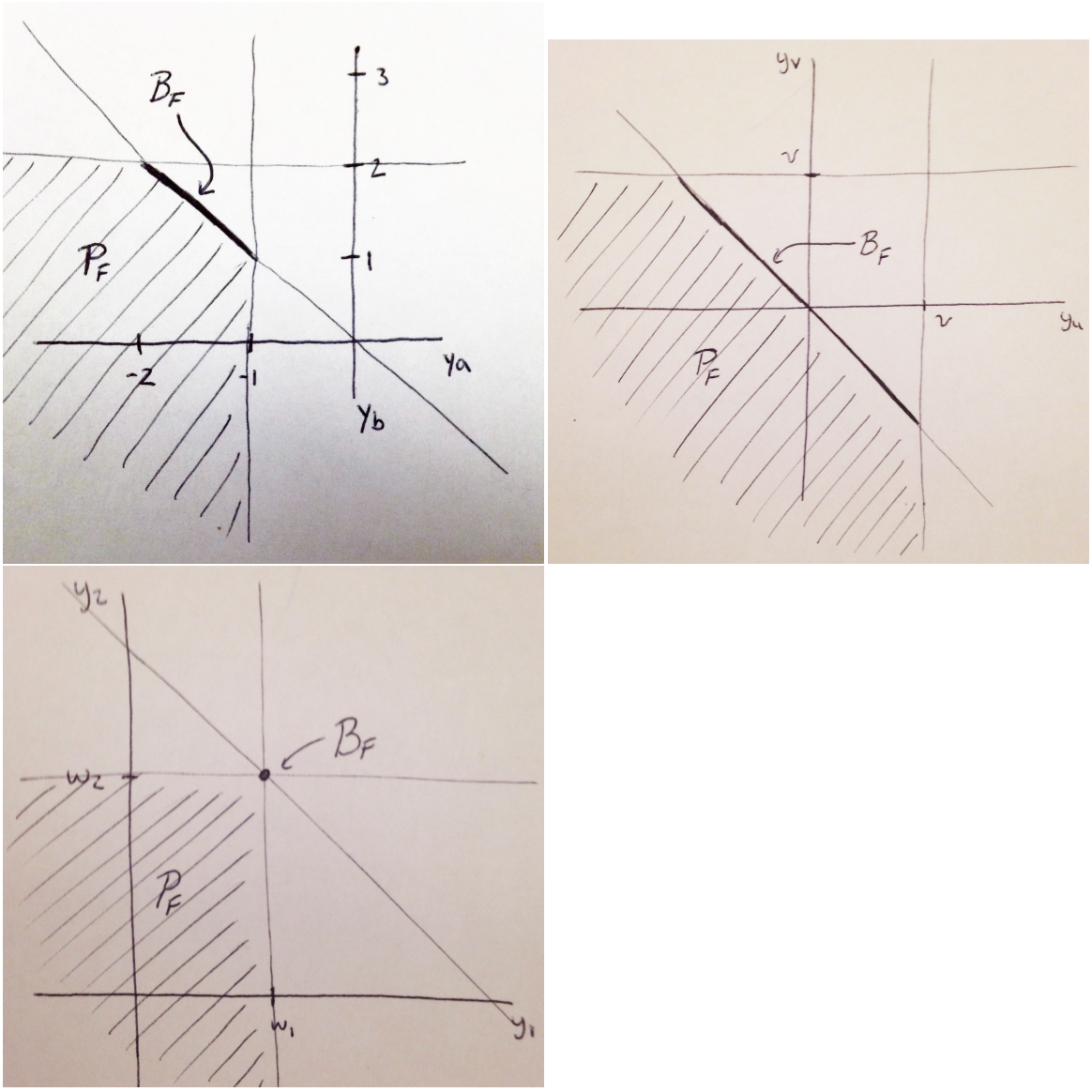

Figure 1: In clockwise order starting from the top left, the submodular polyhedron $\mathcal{P}_F$ and the base polytope $\mathcal{B}_F$ for examples 3.1, 3.2, and 3.3

3.2 Computing the Lovász extension (optimizing over the base polytope)

We turn our attention to actually computing $f(x) = \max_{y \in \mathcal{B}_F} y^T x$. As we saw in the previous section, $\mathcal{B}_F$ is defined by linear constraints, which is good news. The bad news is that there is a linear constraint for each subset $S \subseteq \mathcal{V}$, so we have *exponentially many* linear constraints (in $n = |\mathcal{V}|$).

Despite this shortfall, we can actually solve this maximization problem in $O(n \log n)$ time with the following greedy algorithm:

1. Sort the elements of x, i.e. find a permutation $\pi(\cdot)$ so that
$$x_{\pi(1)} \geq x_{\pi(2)} \geq \cdots \geq x_{\pi(n)}.$$
2. Initialize the sets $S_0 = \emptyset$ and $S_i = \{\pi(1), \pi(2), \ldots, \pi(i)\}$ for $1 \leq i \leq n$.
3. For $1 \leq i \leq n$, set $y_{\pi(i)} = F(S_i) - F(S_{i-1})$.

Intuitively, we look at the constraint $y_{\pi(1)} \leq F(\{\pi(1)\}) = F(S_1)$ and make it tight. Then, we look at the constraint $y_{\pi(1)} + y_{\pi(2)} \leq F(\{\pi(1), \pi(2)\}) = F(S_2)$, and make it tight by setting $y_{\pi(2)} = F(S_2) - y_{\pi(1)} = F(S_2) - F(S_1)$. We continue until all components of y are assigned.

3.2.1 Duality background

Before we can prove that the above procedure actually results in optimal y, we need to review duality theory. Suppose we have a problem of the form

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{s.t.} \quad & Ax \leq b. \end{aligned} \tag{24}$$

We form what is called the *Lagrangian*:

$$\mathcal{L}(x, \lambda) = f(x) + \sum_{i=1}^{m} \lambda_i (a_i^T x - b_i) \tag{25}$$

$$= f(x) + \lambda^T (Ax - b), \tag{26}$$

where $\lambda \geq 0$. Note that if we take the supremum over all $\lambda \geq 0$, if any constraint inequality is not satisfied, we can take the corresponding $\lambda_i \to \infty$, and the Lagrangian is unbounded above. Otherwise, all constraints are satisfied, so one optimal solution is to set all $\lambda_i = 0$, in which case $\mathcal{L}(x, \lambda)$ is just $f(x)$.

We can also form the *Lagrange dual function* $g(\lambda) = \inf_x \mathcal{L}(x, \lambda)$. As the infimum of affine functions of λ, $g(\lambda)$ is concave. Furthermore, if the optimal value of the primal problem

24 is $p^\star$, we have that $g(\lambda) \leq p^\star$. This is because for any feasible $\tilde{x}$, we have $\mathcal{L}(\tilde{x}, \lambda) \leq f(\tilde{x})$; the result follows by taking the infimum over $\tilde{x}$.

By using this fact and maximizing $g(\lambda)$ over all $\lambda \geq 0$, we have what is called *weak duality*:

$$d^\star = \max_{\lambda \geq 0} g(\lambda) \leq p^\star. \tag{27}$$

Under certain conditions, we get what is called *strong duality*, where $d^\star = p^\star$, or in other words, the primal problem and the dual problem $\max_{\lambda \geq 0} g(\lambda)$ have the same optimal value.

3.2.2 Proof of correctness of the greedy algorithm

It happens that strong duality holds for the problem $\max_{y \in \mathcal{B}_F} y^T x$. With this in mind, our strategy will be to construct a y and a λ so that $y^T x = g(\lambda)$, which can only happen if y and λ are optimal for the primal and dual problems, respectively. First, we must compute the Lagrangian and then the dual function $g(\lambda)$ for this problem: [2]

$$\mathcal{L}(y, \lambda) = -x^T y + \sum_{S \subseteq \mathcal{V}} \lambda_S (y(S) - F(S)) \tag{28}$$

$$= \sum_{a \in \mathcal{V}} y_a \left(-x_a + \sum_{S : a \in S} \lambda_S \right) - \sum_{S \subseteq \mathcal{V}} \lambda_S F(S) \tag{29}$$

If the coefficient of any y_a is nonzero, we can take y_a to $\pm\infty$ and the problem is unbounded. Hence, we can write

$$g(\lambda) = \inf_y \mathcal{L}(y, \lambda) = \begin{cases} \sum_{S \subseteq \mathcal{V}} \lambda_S F(S) & \text{if } x_a = \sum_{S : a \in S} \lambda_S \text{ for all } a \\ -\infty & \text{otherwise.} \end{cases} \tag{30}$$

Note that

$$x_a = \sum_{S : a \in S} \lambda_S \; \forall a \in \mathcal{V} \Leftrightarrow x = \sum_{S \subseteq \mathcal{V}} \lambda_S 1_S, \tag{31}$$

and therefore the dual problem is

$$\begin{aligned} \text{maximize} \quad & \textstyle\sum_{S \subseteq \mathcal{V}} \lambda_S F(S) \\ \text{s.t.} \quad & x = \textstyle\sum_{S \subseteq \mathcal{V}} \lambda_S 1_S \\ & \lambda \geq 0. \end{aligned} \tag{32}$$

[2] Instead of dealing with the equality constraint $y(\mathcal{V}) = F(\mathcal{V})$ directly, we will assume $x \geq 0$, so that there is always incentive to increase y; therefore we can replace this equality constraint with $y(\mathcal{V}) \leq F(\mathcal{V})$.

One way to get a feasible solution to this problem is take each λ_S from the level sets (sorting) definition of the Lovász extension, from equation 9. That is, set $\lambda_{\mathcal{V}} = x_{\pi(n)}$ and $\lambda_{S_i} = x_{\pi(i-1)} - x_{\pi(i)}$ for other S_i. This produces

$$g(\lambda) = \sum_{i=2}^{n} (x_{\pi(i-1)} - x_{\pi(i)}) F(S_i) + x_{\pi(n)} F(\mathcal{V}). \tag{33}$$

For the primal problem, we can take $y_{\pi(i)} = F(S_i) - F(S_{i-1})$, so that

$$x^T y = \sum_{i=1}^{n} x_{\pi(i)} (F(S_i) - F(S_{i-1}). \tag{34}$$

This is just equation 10, which we already saw was equal to equation 9. Hence, $g(y) = x^T y$, so y and λ are primal- and dual-optimal, respectively. The greedy algorithm results in the same vector y, so the greedy algorithm is optimal.

3.2.3 Implications

Now that we can compute $f(x) = \max_{y \in \mathcal{B}_F} x^T y$, there are a few important implications:

1. We can compute a vertex of $\mathcal{B}_F$. In general, unless x is perpendicular to $\mathcal{B}_F$ or some edge thereof, the optimal y will be the vertex of $\mathcal{B}_F$ most "aligned" with x.
2. We can compute subgradients of the Lovász extension. This is because $f(x)$ is written as a maximum of affine functions $x^T y$, so all we need to do is find a maximizing y, and then $y \in \partial f(x)$.