Submodularity in Machine Learning

Saifuddin Syed

MLRG Summer 2016

Outline

1 What are submodular functions
- Motivation
- Submodularity and Concavity
- Examples

2 Properties of submodular functions
- Submodularity and Convexity
- Lovász Extension

3 Submodular minimization
- Symmetric Submodular Functions
- Example: Clustering
- Example: Image Denoising

4 Maximization
- Greedy algorithm
- Examples

5 References

Motivation

In combinatorial optimization we are interested solving problems of the form

$$\max\{f(S) : S \in \mathcal{F}\}$$
$$\min\{f(S) : S \in \mathcal{F}\}$$

Where f is some function and $\mathcal{F}$ is some discrete set of feasible solutions. To make the above problems tractable we can either

- Work with each problem individually or
- Try an capture the properties of f and $\mathcal{F}$ that make the above tractable.

Motivation

In the continuous case we have have that $f : \mathbb{R}^n \to \mathbb{R}$ can be

- minimized efficiently if f is **convex** and
- maximized efficiently if f is **concave**.

We want to find the analogy to discrete functions.

Motivation

In the continuous case we have have that $f : \mathbb{R}^n \rightarrow \mathbb{R}$ can be

- minimized efficiently if f is **convex** and
- maximized efficiently if f is **concave**.

We want to find the analogy to discrete functions.

Submodularity is plays the role of concavity/convexity in the discrete regime.

Why should you care about submodularity?

There are many problems in machine learning that can be reformulated in the context of submodular optimization. They have provided elegant solutions to many important problems including:

- Coverage of sensor networks
- Variable selection/regularization
- Clustering
- MAP decoding in graphical models

Notation

For the rest of this talk we will assume V is a set of size n and

$$F : 2^V \to \mathbb{R}$$

where 2^V is the set of all subsets of V. Furthermore, we will assume $F(\emptyset) = 0$

Notation

For the rest of this talk we will assume V is a set of size n and

$$F : 2^V \to \mathbb{R}$$

where 2^V is the set of all subsets of V. Furthermore, we will assume $F(\emptyset) = 0$

Given $S \in 2^V$, we define $F_S : V \to \mathbb{R}$ by

$$F_S(i) = F(S \cup \{i\}) - F(S).$$

$F_S(i)$ represents the **marginal value** of i with respect to S.

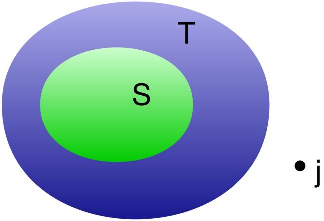
Submodularity

Definition

F is **submodular** if for all $S \subset T$ and $j \in V \backslash T$

$$F_S(j) \geq F_T(j).$$

F is **supermodular** if $-F$ is submodular.
F is **modular** (or **additive**) if it is both submodular and supermodular.



Intuitively the submodular condition says that "you have more to gain from something new, if you have less to begin with."

Intuitively the submodular condition says that "you have more to gain from something new, if you have less to begin with."

Note: Sometimes the less intuitive (but equivalent) definition of submodularity is used. F is submodular if for all $A, B \subset V$

$$F(A) + F(B) \geq F(A \cup B) + F(A \cap B).$$

More Notation

Note that $F : 2^V \to \mathbb{R}$ induces a function $\hat{F} : \{0,1\}^n \to \mathbb{R}$ by

$$\hat{F}(1_A) = F(A)$$

Where 1_A is the **indicator** function for A. I.e.,

$$1_A = (x_1^A, \ldots, x_n^A)$$

Where $x_i^A = 1$ if $i \in A$ and 0 otherwise.

We will use $\hat{F}$ and F interchangeably.

Submodularity and Concavity

In some sense submodular functions are the discrete analogue of concave functions.

Submodularity and Concavity

In some sense submodular functions are the discrete analogue of concave functions.

- $f : \mathbb{R} \to \mathbb{R}$ is concave is the derivative $f'(x)$ is non-increasing in x.

Submodularity and Concavity

In some sense submodular functions are the discrete analogue of concave functions.

- $f : \mathbb{R} \to \mathbb{R}$ is concave is the derivative $f'(x)$ is non-increasing in x.
- $F : \{0,1\}^n \to \mathbb{R}$ is **submodular** if $\forall i$ the discrete derivative,

$$\partial_i f(x) = f(x + e_i) - f(x),$$

is non-increasing in x.

Submodularity and Concavity

In some sense submodular functions are the discrete analogue of concave functions.

- $f : \mathbb{R} \rightarrow \mathbb{R}$ is concave is the derivative $f'(x)$ is non-increasing in x.
- $F : \{0,1\}^n \rightarrow \mathbb{R}$ is **submodular** if $\forall i$ the discrete derivative,

  $$\partial_i f(x) = f(x + e_i) - f(x),$$

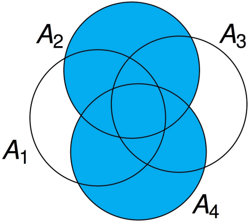
  is non-increasing in x.
- Furthermore if $g : \mathbb{R}_+ \rightarrow \mathbb{R}$ is concave, then $F(A) = g(|A|)$ is submodular.

Examples of submodular functions

- **Coverage function.** Suppose $(A_i)_{i \in V}$ are measurable sets . Then

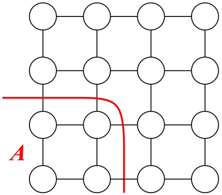
$$F(S) = |\cup_{i \in S} A_i|$$

is submodular.



Examples of submodular functions

- **Cut functions.** Given a (un)directed graph (V, E). Define $F(A)$ to be the total number of edges from A to $V \backslash A$ is submodular.



- More generally if $d : V \times V \to \mathbb{R}_+$ then

$$F(A) = \sum_{i \in A, j \in V \backslash A} d(i, j)$$

is submodular.

Examples of submodular functions

- **Entropy.** Given n random variables $(X_i)_{i \in V}$, define

$$F(A) = H(X_A)$$

to be the joint entropy. Then F is submodular.

Examples of submodular functions

- **Entropy.** Given n random variables $(X_i)_{i \in V}$, define

$$F(A) = H(X_A)$$

to be the joint entropy. Then F is submodular.

Indeed, suppose that $A \subset B$, $k \in V \backslash B$, then

$$\begin{aligned} F(A \cup \{k\}) - F(A) &= H(X_A, X_k) - H(X_A) \\ &= H(X_k | X_A) \\ &\geq H(X_k | X_B) \end{aligned}$$

Examples of submodular functions

- **Entropy.** Given n random variables $(X_i)_{i \in V}$, define

$$F(A) = H(X_A)$$

to be the joint entropy. Then F is submodular.

Indeed, suppose that $A \subset B$, $k \in V \backslash B$, then

$$\begin{aligned} F(A \cup \{k\}) - F(A) &= H(X_A, X_k) - H(X_A) \\ &= H(X_k | X_A) \\ &\geq H(X_k | X_B) \end{aligned}$$

- **Mutual information** also submodular.

$$I(A) = F(A) + F(V \backslash A) - F(V)$$

Outline

1 What are submodular functions
- Motivation
- Submodularity and Concavity
- Examples

2 Properties of submodular functions
- Submodularity and Convexity
- Lovász Extension

3 Submodular minimization
- Symmetric Submodular Functions
- Example: Clustering
- Example: Image Denoising

4 Maximization
- Greedy algorithm
- Examples

5 References

Properties of Submodular Functions

- **Positive linear combinations:** If F_i are submodular and $\alpha_i \geq 0$ then

$$\sum_i \alpha_i F_i$$

is submodular.

Properties of Submodular Functions

- **Positive linear combinations:** If F_i are submodular and $\alpha_i \geq 0$ then

$$\sum_i \alpha_i F_i$$

is submodular.

- **Restriction/marginalization:** If $B \subset V$ and F is submodular, then

$$A \rightarrow F(A \cap B)$$

is submodular on V and B.

Properties of Submodular Functions

- **Positive linear combinations:** If F_i are submodular and $\alpha_i \geq 0$ then

$$\sum_i \alpha_i F_i$$

is submodular.

- **Restriction/marginalization:** If $B \subset V$ and F is submodular, then

$$A \to F(A \cap B)$$

is submodular on V and B.

- **Contraction/conditioning:** If $B \subset V$ and F is submodular, then

$$A \to F(A \cup B) - F(B)$$

Is submodular on V and $V \backslash B$

Properties of Submodular Functions

Remark: If F, G are submodular then

$$\max\{F, G\},$$
$$\min\{F, G\}$$

need **NOT** be submodular.

Submodularity and Convexity

Although submodular functions are defined like concave functions, their behaviour is very similar to convex functions. Before we explore this relation, we will need more notation.

Submodularity and Convexity

Although submodular functions are defined like concave functions, their behaviour is very similar to convex functions. Before we explore this relation, we will need more notation.

Given $x \in \mathbb{R}^n_+$, $A \subset V$ define

$$x(A) = \sum_{i \in A} x_i = x^T 1_A$$

Where $1_A \in \mathbb{R}^n$ is the indicator of A.

Lovász Extension

Given $F : \{0,1\}^n \to \mathbb{R}$ we will define the **Lovász extension** $f : \mathbb{R}^n \to \mathbb{R}$ as follows. For $w \in \mathbb{R}^n$, order $w_{j_1} \geq \cdots \geq w_{j_n}$ and then

$$
\begin{aligned}
f(w) &= w_{j_1} F(\{j_1\}) + \sum_{k=2}^{n} w_{j_k} [F(\{j_1, \ldots, j_k\}) - F(\{j_1, \ldots, j_{k-1}\})] \\
&= w_{j_1} F(\{j_1\}) + \sum_{k=2}^{n} w_{j_k} F_{V_{k-1}}(j_k)
\end{aligned}
$$

Where $V_k = \{j_1, \ldots, j_k\}$.

Intuitively you are summing the marginal gains of F, weighted by the components of w.

Lovász Extension

The following are equivalent definitions of the Lovász Extension.

$$
\begin{aligned}
f(w) &= w_{j_1} F(\{j_1\}) + \sum_{k=2}^{n} w_{j_k} F_{V_{k-1}}(j_k) && (1)\\
&= \sum_{k=1}^{n-1} (w_{j_k} - w_{j_{k+1}}) F(V_k) + w_{j_n} F(V) && (2)\\
&= \int_{w_{j_n}}^{\infty} F(w \geq z) dz + w_{j_n} F(V) && (3)\\
&= \sup_{x \in P(F)} w^T x && (4)
\end{aligned}
$$

Where $P(F) = \{x \in \mathbb{R}^n : \forall A \subset V, x(A) \leq F(A)\}$, is the **submodular Polyhedra**.

Properties of Lovász Extension

- f is indeed an **extension** of F. For $A \subset V$,

$$f(1_A) = F(A).$$

Properties of Lovász Extension

- f is indeed an **extension** of F. For $A \subset V$,

$$f(1_A) = F(A).$$

- f is peicewise affine
- f is convex iff F is submodular

Properties of Lovász Extension

- f is indeed an **extension** of F. For $A \subset V$,

$$f(1_A) = F(A).$$

- f is peicewise affine
- f is convex iff F is submodular
- If f is restricted to $[0,1]^n$, then f attains it's minimum at the corner! I.e.

$$\min_{w \in [0,1]^n} f(w) = \min_{x \in \{0,1\}} F(x)$$

Outline

1 What are submodular functions
- Motivation
- Submodularity and Concavity
- Examples

2 Properties of submodular functions
- Submodularity and Convexity
- Lovász Extension

3 Submodular minimization
- Symmetric Submodular Functions
- Example: Clustering
- Example: Image Denoising

4 Maximization
- Greedy algorithm
- Examples

5 References

Minimization of Submodular functions

Suppose we now want to find the minimizing set of a submodular function. Ie, we want to find

$$A^* = \operatorname{argmin}\{F(A) : A \subset V\}$$

By the Lovász extention it is equivalent to finding

$$\operatorname{argmin}\{f(w) : w \in [0,1]^n\},$$

where f is the Lovász function of F.

Minimization of Submodular functions

Suppose we now want to find the minimizing set of a submodular function. Ie, we want to find

$$A^* = \operatorname{argmin}\{F(A) : A \subset V\}$$

By the Lovász extention it is equivalent to finding

$$\operatorname{argmin}\{f(w) : w \in [0,1]^n\},$$

where f is the Lovász function of F.

Theorem

f can be minimized using the Ellipsoid method in $O(n^8 \log^2 n)$.

Symmetric Submodular Functions

We can knock down that $O(n^8)$ time down if we impose some extra structure onto F.

Symmetric Submodular Functions

We can knock down that $O(n^8)$ time down if we impose some extra structure onto F.

We say that F is **symmetric** if $F(A) = F(V \backslash A)$. Examples include:

Symmetric Submodular Functions

We can knock down that $O(n^8)$ time down if we impose some extra structure onto F.

We say that F is **symmetric** if $F(A) = F(V \backslash A)$. Examples include:

- **Mutual Information**. Given random variables $(X_i)_{i \in V}$ then

$$F(A) = I(X_A; X_{V \backslash A}) = I(X_{V \backslash A}; X_A) = F(V \backslash A)$$

Symmetric Submodular Functions

We can knock down that $O(n^8)$ time down if we impose some extra structure onto F.

We say that F is **symmetric** if $F(A) = F(V \backslash A)$. Examples include:

- **Mutual Information**. Given random variables $(X_i)_{i \in V}$ then

$$F(A) = I(X_A; X_{V \backslash A}) = I(X_{V \backslash A}; X_A) = F(V \backslash A)$$

- **Cut functions**. Given a weighted graph (V, E), with weights $\{d(e)\}_{e \in E}$

$$F(A) = \sum_{i \in A, j \in V \backslash A} d(i, j) = F(V \backslash A).$$

Symmetric Submodular functions

Note that for symmetric sub modular functions

$$\begin{aligned} 2F(A) &= F(A) + F(V \backslash A) \\ &\geq F(A \cap (V \backslash A)) + F(A \cup (V \backslash A)) \\ &= F(\emptyset) + f(V) \\ &= 2F(\emptyset) \\ &= 0 \end{aligned}$$

Symmetric Submodular functions

Note that for symmetric sub modular functions

$$\begin{aligned} 2F(A) &= F(A) + F(V \backslash A) \\ &\geq F(A \cap (V \backslash A)) + F(A \cup (V \backslash A)) \\ &= F(\emptyset) + f(V) \\ &= 2F(\emptyset) \\ &= 0 \end{aligned}$$

So $F(A)$ is trivially minimized at V. We are interested in

$$\mathrm{argmin}\{F(A) : A \subset V, 0 < |A| < n\}$$

Theorem (Queyranne 98)

If F is a symmetric submodular function, then there is a fully combinatorial, algorithm for solving

$$\operatorname{argmin}\{F(A) : A \subset V, 0 < |A| < n\}$$

with run time $O(n^3)$.

The algorithm is very easy to implement but requires some new machinery that we don't have time for.

See slides 47-53 of
"`http://submodularity.org/submodularity-slides.pdf`"

Example: Clustering

Suppose we want to partition V into k clusters $A_1, \ldots, A_k$ such that

$$F(A_1, \ldots, A_k) = \sum_{i=1}^{k} E(A_i)$$

Where E is some submodular function such as Entropy, or a cut functions.

Example: Clustering

Suppose we want to partition V into k clusters $A_1, \ldots, A_k$ such that

$$F(A_1, \ldots, A_k) = \sum_{i=1}^{k} E(A_i)$$

Where E is some submodular function such as Entropy, or a cut functions.

In the special case of $k = 2$, then

$$F(A) = E(A) + E(V \backslash A)$$

is symmetric and submodular and thus we can apply Queyranne's algorithm

Example: Clustering

When $k > 2$ we can apply a greedy slitting algorithm.

1. Initially let the partition $P_1 = \{V\}$.
2. For $i = 1 \ldots k-1$.
   - For each $C_j \in P_i$;
   - Get a partition P_i^j from splitting C_j in 2 using Queyranne's algorithm.
   - $P_{i+1} = \mathrm{argmin} F(P_i^j)$

Example: Clustering

When $k > 2$ we can apply a greedy slitting algorithm.

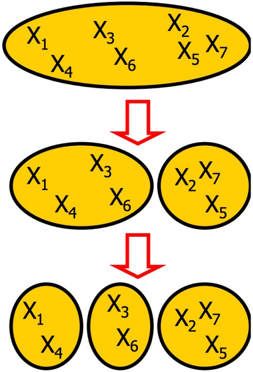
1. Initially let the partition $P_1 = \{V\}$.
2. For $i = 1 \ldots k - 1$.
   - For each $C_j \in P_i$;
   - Get a partition P_i^j from splitting C_j in 2 using Queyranne's algorithm.
   - $P_{i+1} = \mathrm{argmin} F(P_i^j)$

Theorem

If P is the partition of size k from the greedy splitting algorithm, then

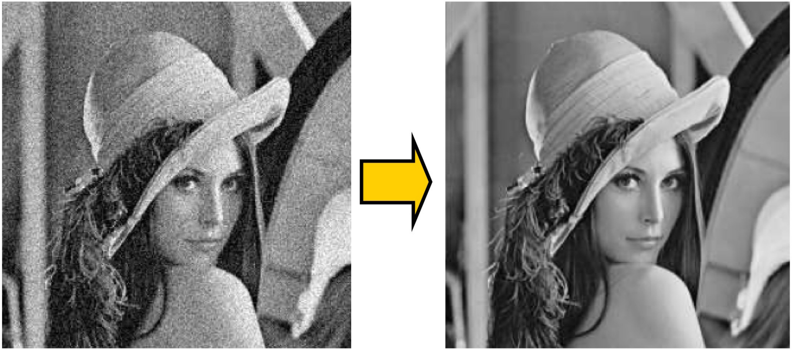
$$F(P) \leq \left(2 - \frac{2}{k}\right) F(P_{opt})$$

Example: Clustering



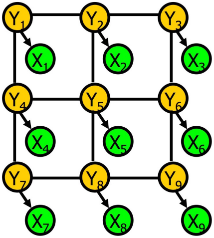
Example: Image Denoising

Suppose we have a noisy image and we want to find the true underlying image?



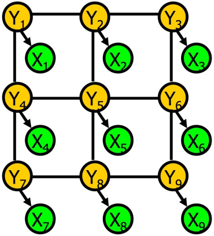
Example: Image Denoising

Suppose we have a Pairwise Markov Random Field. Suppose Y_i are the true pixels and X_i are the "noisey" ones.



Example: Image Denoising

Suppose we have a Pairwise Markov Random Field. Suppose Y_i are the true pixels and X_i are the "noisey" ones.



So we have the graphical model,

$$P(X_1, \ldots, X_n, Y_1, \ldots, Y_n) = \prod_{i,j} \psi_{i,j}(Y_i, Y_j) \prod_i \phi_i(X_i, Y_i)$$

Example: Image Denoising

To find the MAP estimate we want,

$$\begin{aligned} &\mathrm{argmax}_Y P(Y|X) \\ =&\mathrm{argmax}_Y P(X, Y) \\ =&\mathrm{argmin}_Y \sum_{i,j} E_i(Y_i, Y_j) + \sum_i E_i(Y_i) \end{aligned}$$

Where

$$\begin{aligned} E_{i,j}(Y_i, Y_j) &= -\log \psi_{i,j}(Y_i, Y_j) \\ E_i(Y_i) &= -\log \phi_i(X_i, Y_i) \end{aligned}$$

In genral When is the MAP inference efficiently solvable (in high tree width graphical models)? In general it is NP-hard.

Example: Image Denoising

Suppose y_i are binary, then we have

Theorem (Kolmogorov, Kabih,'04)

MAP inference problem is solvable by **graph cuts**
iff for all i, j,

$$E_{i,j}(0,0) + E_{i,j}(1,1) \leq E_{i,j}(0,1) + E_{i,j}(1,0)$$

iff each $E_{i,j}$ *is submodular.*

See
"`http://www.cs.cornell.edu/~rdz/papers/kz-pami04.pdf`"
if you are interested in seeing the details.

Outline

1 What are submodular functions
- Motivation
- Submodularity and Concavity
- Examples

2 Properties of submodular functions
- Submodularity and Convexity
- Lovász Extension

3 Submodular minimization
- Symmetric Submodular Functions
- Example: Clustering
- Example: Image Denoising

4 Maximization
- Greedy algorithm
- Examples

5 References

Submodular maximization

Again, even though submodular functions are defined to emulate concave functions, in practice they behave like convex ones.

Submodular maximization

Again, even though submodular functions are defined to emulate concave functions, in practice they behave like convex ones.

Convex functions:

- Minimizing $\Rightarrow$ polynomial time
- Maximizing $\Rightarrow$ NP-hard

Submodular maximization

Again, even though submodular functions are defined to emulate concave functions, in practice they behave like convex ones.

Convex functions:

- Minimizing $\Rightarrow$ polynomial time
- Maximizing $\Rightarrow$ *NP*-hard

Submodular functions:

- Minimizing $\Rightarrow$ polynomial time
- Maximizing $\Rightarrow$ *NP*-hard

Submodular maximization

Again, even though submodular functions are defined to emulate concave functions, in practice they behave like convex ones.

Convex functions:

- Minimizing $\Rightarrow$ polynomial time
- Maximizing $\Rightarrow$ NP-hard

Submodular functions:

- Minimizing $\Rightarrow$ polynomial time
- Maximizing $\Rightarrow$ NP-hard

BUT all hope is not lost, as we can sometimes efficiently get approximate guarantees!

Monotonic Functions

We say that F is **monotonic** if $A \subset B$ then

$$F(A) \leq F(B)$$

Monotonic Functions

We say that F is **monotonic** if $A \subset B$ then

$$F(A) \leq F(B)$$

Some examples include:

- **Coverage function**. If $(A_i)_{i \in V}$ are measureable sets, then $A \subset B \subset V$,

$$F(A) = |\cup_{i \in A} A_i| \leq |\cup_{i \in B} A_i| = F(B)$$

Monotonic Functions

We say that F is **monotonic** if $A \subset B$ then

$$F(A) \leq F(B)$$

Some examples include:

- **Coverage function**. If $(A_i)_{i \in V}$ are measureable sets, then $A \subset B \subset V$,

$$F(A) = |\cup_{i \in A} A_i| \leq |\cup_{i \in B} A_i| = F(B)$$

- **Entropy**. If $(X_i)_{i \in V}$ are random variables then if $B = A \cup C \subset V$,

$$F(B) = H(X_A, X_C) = H(X_A) + H(X_C | X_A) \geq H(X_A) = F(A)$$

- Similarly **Information Gain** is an other example.

Greedy Algorithm

For monotonic functions we clearly have F is maximized at V. So we are interested in the constraint problem:

$$\mathrm{argmax}_{|A|\leq k} F(A).$$

Greedy Algorithm

For monotonic functions we clearly have F is maximized at V. So we are interested in the constraint problem:

$$\mathrm{argmax}_{|A|\leq k} F(A).$$

We will apply the greedy approach.

1. Initialize $A_0 = \emptyset$
2. For $i = 1$ to k:
   - $x_i = \mathrm{argmax}_x F_{A_{i-1}}(x) = \mathrm{argmax}_x F(A_{i-1} \cup \{x\}) - F(A_{i-1})$
   - $A_i = A_{i-1} \cup \{x_i\}$

Greedy Algorithm

For monotonic functions we clearly have F is maximized at V. So we are interested in the constraint problem:

$$\mathrm{argmax}_{|A|\leq k} F(A).$$

We will apply the greedy approach.

1. Initialize $A_0 = \emptyset$
2. For $i = 1$ to k:
   - $x_i = \mathrm{argmax}_x F_{A_{i-1}}(x) = \mathrm{argmax}_x F(A_{i-1} \cup \{x\}) - F(A_{i-1})$
   - $A_i = A_{i-1} \cup \{x_i\}$

Theorem (Nemhauser et al 78)

Given a monotonic submodular function F, then

$$F(A_{greedy}) \geq \left(1 - \frac{1}{e}\right) \max_{|A|\leq k} F(A) \approx 0.63 \max_{|A|\leq k} F(A)$$

Example: Variance Reduction

Suppose we have the linear model

$$Y = \sum_{i=1}^{n} \alpha_i X_i$$

- Each X_i represents a measurement by some sensor i with joint distribution $P(X_1, \ldots, X_n)$.
- Let V denote the set of possible sensors.
- Sensors are expensive so we want to pick the best k sensors that minimized the variance in the prediction Y.

Example: Variance Reduction

Suppose we have the linear model

$$Y = \sum_{i=1}^{n} \alpha_i X_i$$

- Each X_i represents a measurement by some sensor i with joint distribution $P(X_1, \ldots, X_n)$.
- Let V denote the set of possible sensors.
- Sensors are expensive so we want to pick the best k sensors that minimized the variance in the prediction Y.

We want to find $|A| \leq k$ such that $Var(Y|X_A)$ is minimized. Equivalently we want to find A such that the variance reduction is maximized ie.

$$F(A) = Var(Y) - Var(Y|X_A)$$

Example: Variance Reduction

$$\operatorname{argmax}_{|A|\leq k} F(A) = \operatorname{argmax}_{|A|\leq k} Var(Y) - Var(Y|X_A)$$

In general this problem is *NP*-hard but It should be noted that F is always monotonic.

Example: Variance Reduction

$$\mathrm{argmax}_{|A|\leq k} F(A) = \mathrm{argmax}_{|A|\leq k} Var(Y) - Var(Y|X_A)$$

In general this problem is *NP*-hard but It should be noted that F is always monotonic.

Theorem (Das & Kempe, 08)

If $X_1, \ldots, X_n$ are jointly Gaussian, then F is submodular.

Thus we can apply the greedy algorithm!

Outline

1. What are submodular functions
   - Motivation
   - Submodularity and Concavity
   - Examples
2. Properties of submodular functions
   - Submodularity and Convexity
   - Lovász Extension
3. Submodular minimization
   - Symmetric Submodular Functions
   - Example: Clustering
   - Example: Image Denoising
4. Maximization
   - Greedy algorithm
   - Examples
5. References

References

These are some of the sources I used to prepare for this talk and I think are good to check out in case you are further interested in submodularity or want more of a rigourous treatment.

Some slides worth reading:

- http://www.di.ens.fr/~fbach/submodular_fbach_mlss2012.pdf
- http://submodularity.org/submodularity-slides.pdf
- http://theory.stanford.edu/~jvondrak/data/submod-tutorial-1.pdf

The following notes from Francis Bach were very helpful especially if you are interested in the theory as opposed to a big picture overview.

- http://arxiv.org/pdf/1010.4207.pdf