Submodular Optimization with Submodular Cover and Submodular Knapsack Constraints (SCSC/ SCSK)

Rishabh Iyer Jeff Bilmes

University of Washington, Seattle

NIPS-2013

Outline

Set functions $f : 2^V \rightarrow \mathbb{R}$

$$V = \left\{ \text{  } \right\}$$

- V is a finite "ground" set of objects.
- A set function $f : 2^V \rightarrow \mathbb{R}$ produces a value for any subset $A \subseteq V$.

Set functions $f : 2^V \rightarrow \mathbb{R}$

$$A = \left\{ \text{  }, \text{  }, \text{  }, \text{  }, \right\}$$

- For example, $f(A) = 22$,

Submodular Set Functions

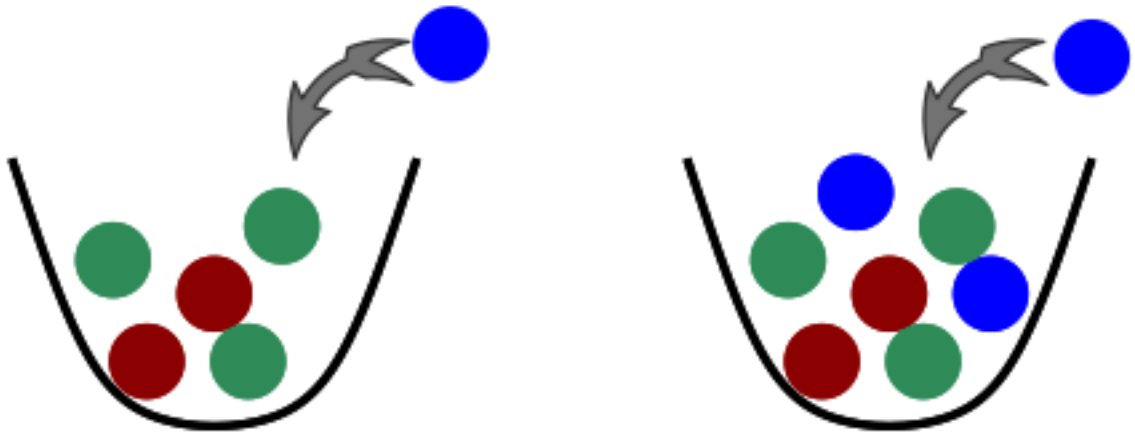
- Special class of set functions.

$$f(A \cup v) - f(A) \geq f(B \cup v) - f(B), \text{ if } A \subseteq B \tag{1}$$

Submodular Set Functions

- Special class of set functions.

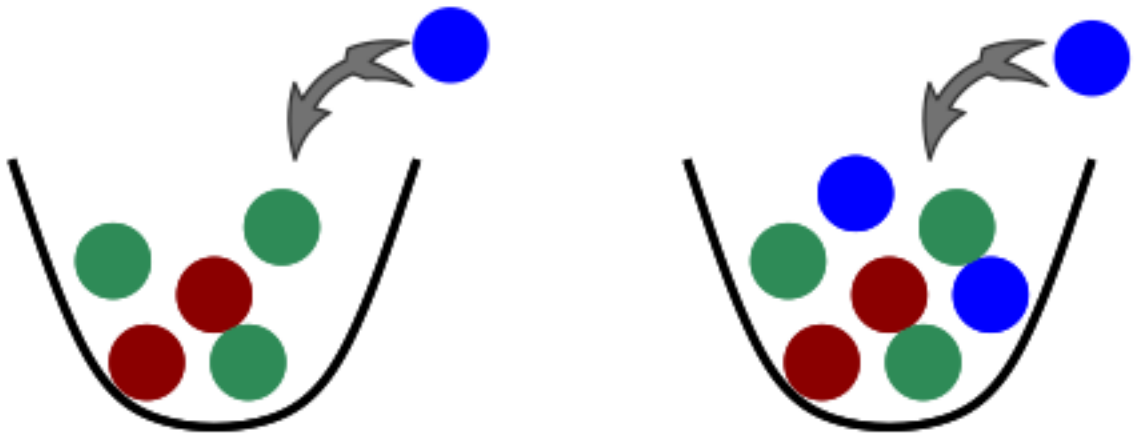
$$f(A \cup v) - f(A) \geq f(B \cup v) - f(B), \text{ if } A \subseteq B \tag{1}$$



Submodular Set Functions

- Special class of set functions.

$$f(A \cup v) - f(A) \geq f(B \cup v) - f(B), \text{ if } A \subseteq B \quad (1)$$

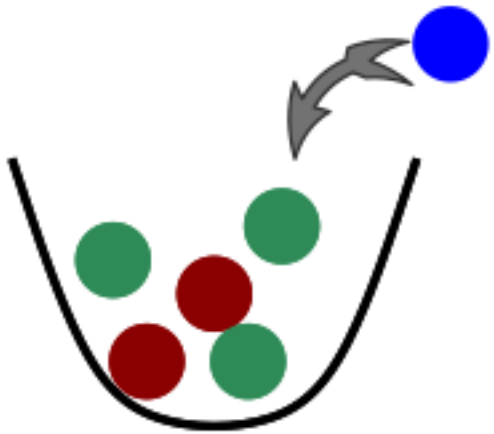


Gain $= 1$

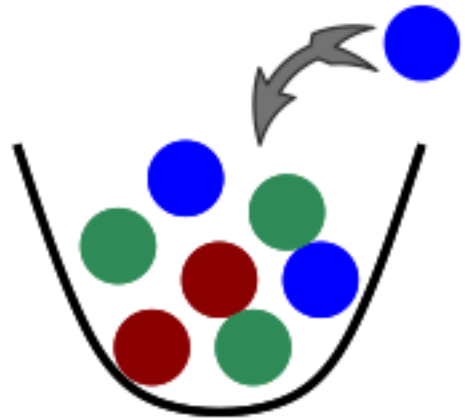
Submodular Set Functions

- Special class of set functions.

$$f(A \cup v) - f(A) \geq f(B \cup v) - f(B), \text{ if } A \subseteq B \quad (1)$$



Gain $= 1$

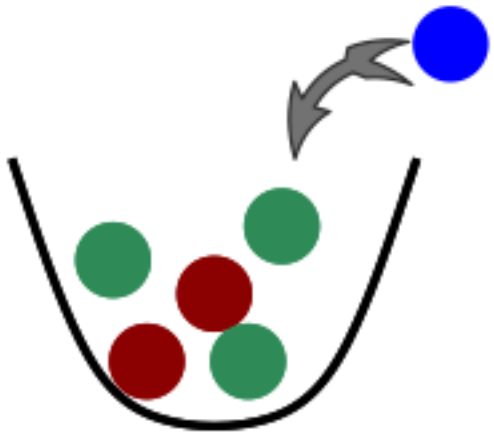


Gain $= 0$

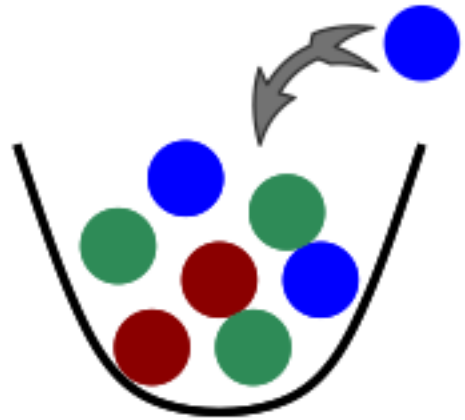
Submodular Set Functions

- Special class of set functions.

$$f(A \cup v) - f(A) \geq f(B \cup v) - f(B), \text{ if } A \subseteq B \tag{1}$$



Gain $= 1$



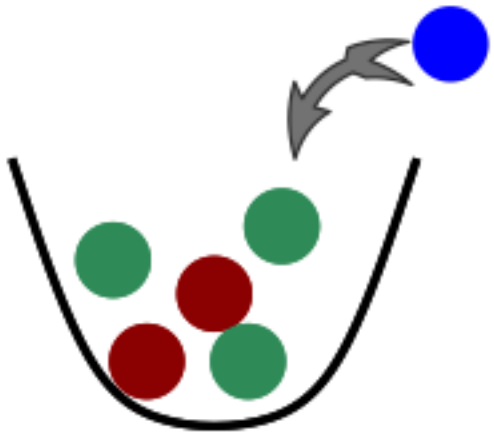
Gain $= 0$

- Monotonicity: $f(A) \leq f(B)$, if $A \subseteq B$.

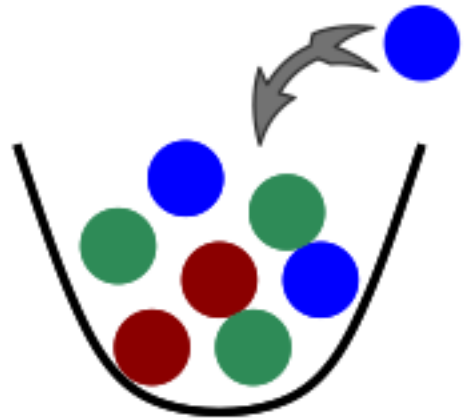
Submodular Set Functions

- Special class of set functions.

$$f(A \cup v) - f(A) \geq f(B \cup v) - f(B), \text{ if } A \subseteq B \quad (1)$$



Gain $= 1$



Gain $= 0$

- Monotonicity: $f(A) \leq f(B)$, if $A \subseteq B$.
- Modular function $f(X) = \sum_{i \in X} f(i)$ analogous to linear functions.

Two Sides of Submodularity

Two Sides of Submodularity

Submodular Minimization

- Solve $\min\{f(X)|X \subseteq V\}$.
- Polynomial-time.
- Relation to convexity.
- Models cooperation.

$f($  $)-f($  $) \geq f($  $)-f($  $)$

Two Sides of Submodularity

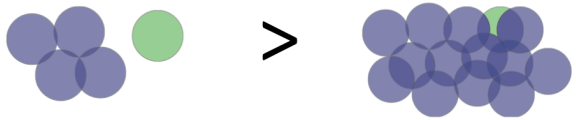
Submodular Minimization

- Solve $\min\{f(X)|X \subseteq V\}$.
- Polynomial-time.
- Relation to convexity.
- Models cooperation.

$f($  $)-f($  $) \geq f($  $)-f($  $)$

Submodular Maximization

- Solve $\max\{g(X)|X \subseteq V\}$.
- Constant-factor approximable.
- Relation to concavity.
- Models diversity and coverage.



Two Sides of Submodularity

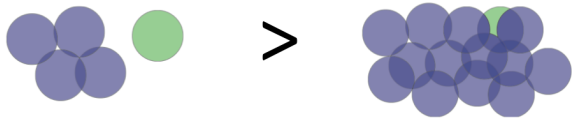
Submodular Minimization

- Solve $\min\{f(X)|X \subseteq V\}$.
- Polynomial-time.
- Relation to convexity.
- Models cooperation.

$f($  $)-f($  $) \geq f($  $)-f($  $)$

Submodular Maximization

- Solve $\max\{g(X)|X \subseteq V\}$.
- Constant-factor approximable.
- Relation to concavity.
- Models diversity and coverage.



- Sometimes we want to simultaneously maximize coverage/ diversity (g) while minimizing cooperative costs (f).

Two Sides of Submodularity

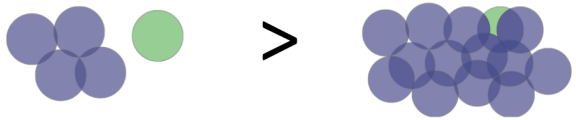
Submodular Minimization

- Solve $\min\{f(X)|X \subseteq V\}$.
- Polynomial-time.
- Relation to convexity.
- Models cooperation.

$f($  $)-f($  $) \geq f($  $)-f($  $)$

Submodular Maximization

- Solve $\max\{g(X)|X \subseteq V\}$.
- Constant-factor approximable.
- Relation to concavity.
- Models diversity and coverage.



- Sometimes we want to simultaneously maximize coverage/ diversity (g) while minimizing cooperative costs (f).
- Often these naturally occur as budget or cover constraints (for example, maximize diversity subject to a budget constraint on the submodular cost).

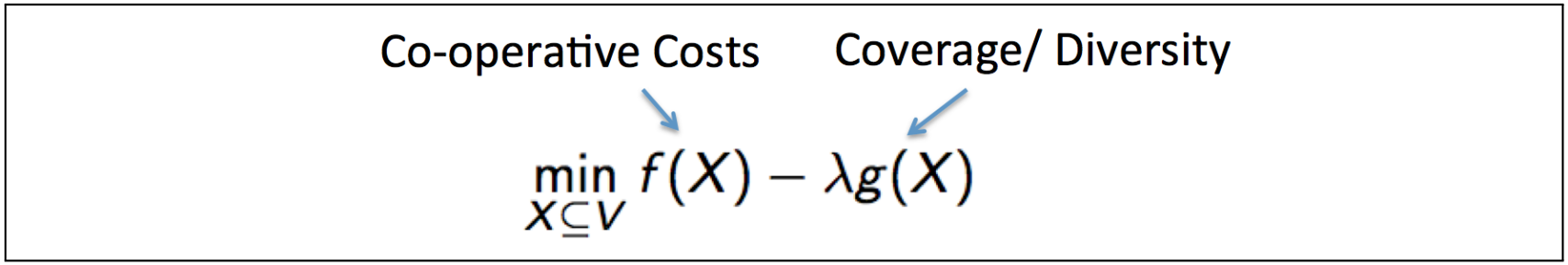
Submodular Optimization with Submodular Constraints

- Historically: DS optimization

$$\min_{X \subseteq V} f(X) - \lambda g(X)$$

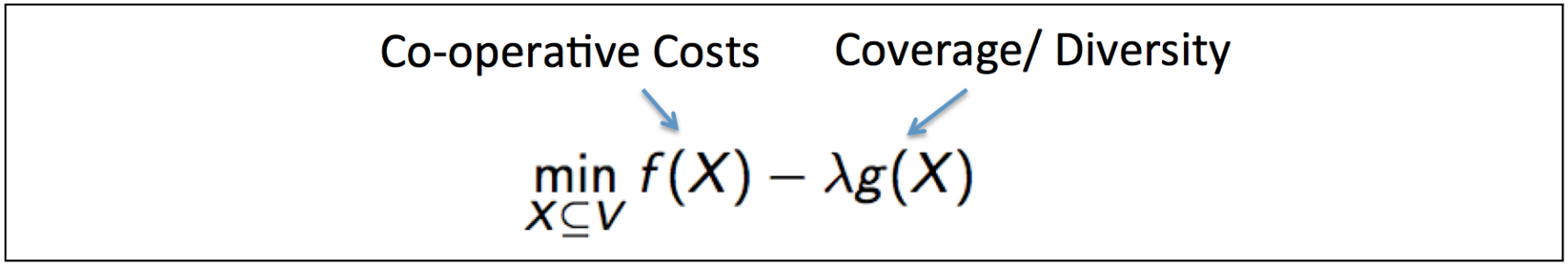
Submodular Optimization with Submodular Constraints

- Historically: DS optimization



Submodular Optimization with Submodular Constraints

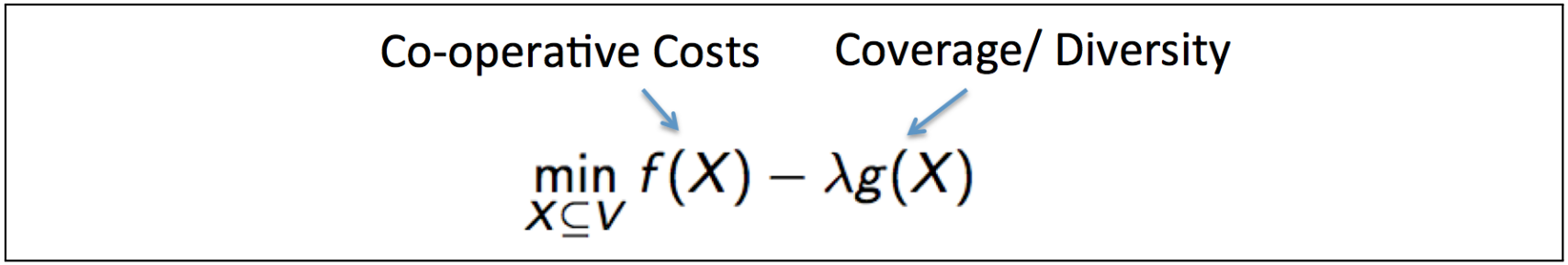
- Historically: DS optimization



- Unfortunately, NP hard to approximate (Iyer-Bilmes'12).

Submodular Optimization with Submodular Constraints

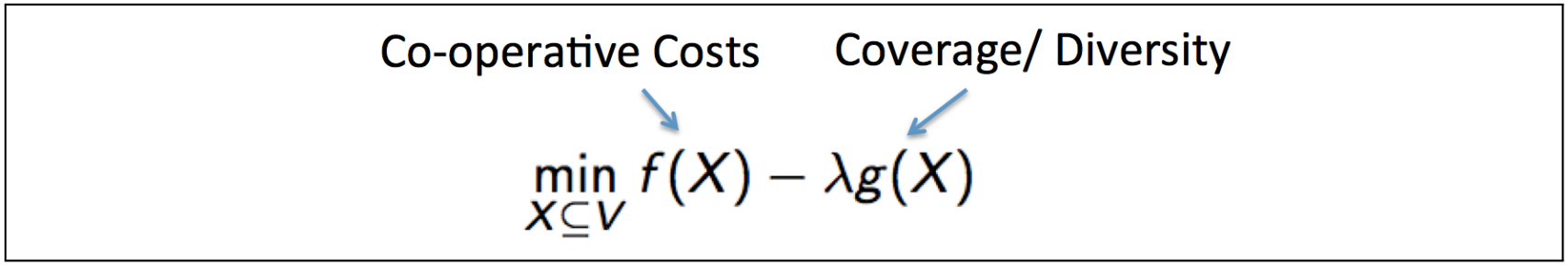
- Historically: DS optimization



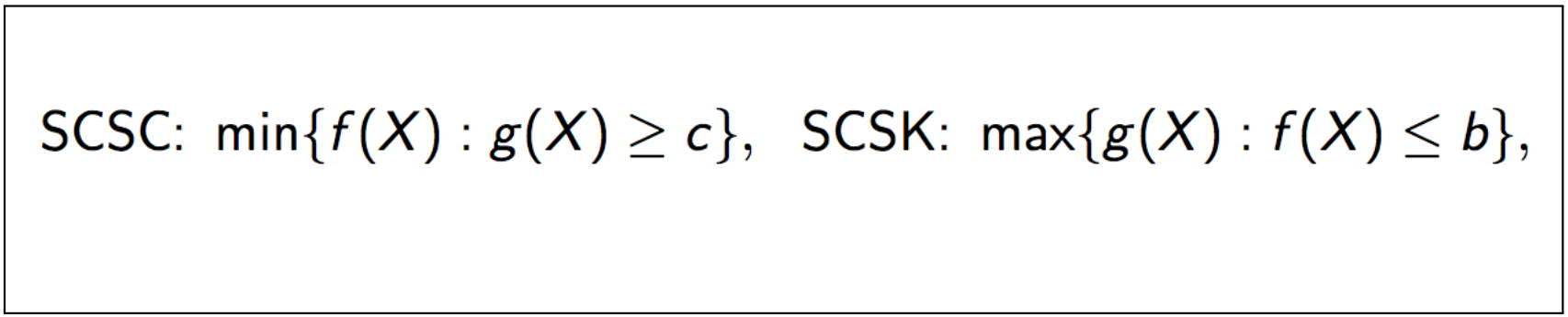
- Unfortunately, NP hard to approximate (Iyer-Bilmes'12).
- We introduce the following, which is often more natual anyway:

Submodular Optimization with Submodular Constraints

- Historically: DS optimization



- Unfortunately, NP hard to approximate (Iyer-Bilmes'12).
- We introduce the following, which is often more natual anyway:



Submodular Optimization with Submodular Constraints

- Historically: DS optimization

Co-operative Costs Coverage/ Diversity

$$\min_{X \subseteq V} f(X) - \lambda g(X)$$

- Unfortunately, NP hard to approximate (Iyer-Bilmes'12).
- We introduce the following, which is often more natual anyway:

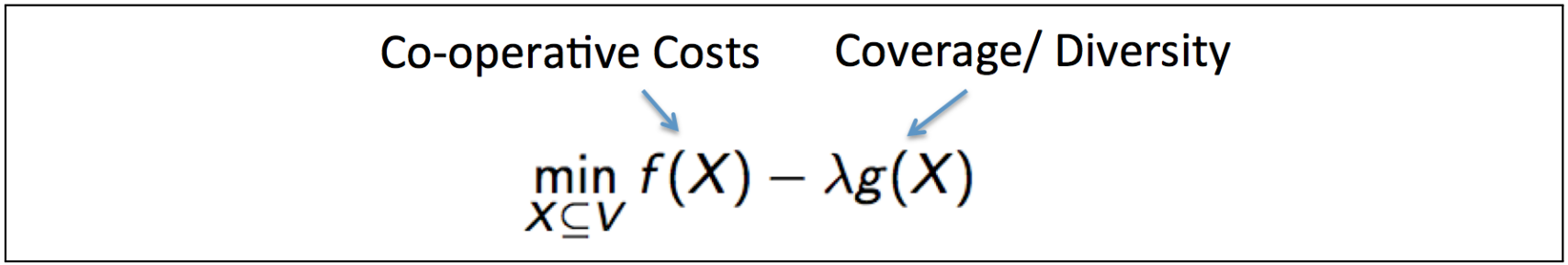
Coverage/ Diversity

SCSC: $\min\{f(X) : g(X) \geq c\}$, SCSK: $\max\{g(X) : f(X) \leq b\}$,

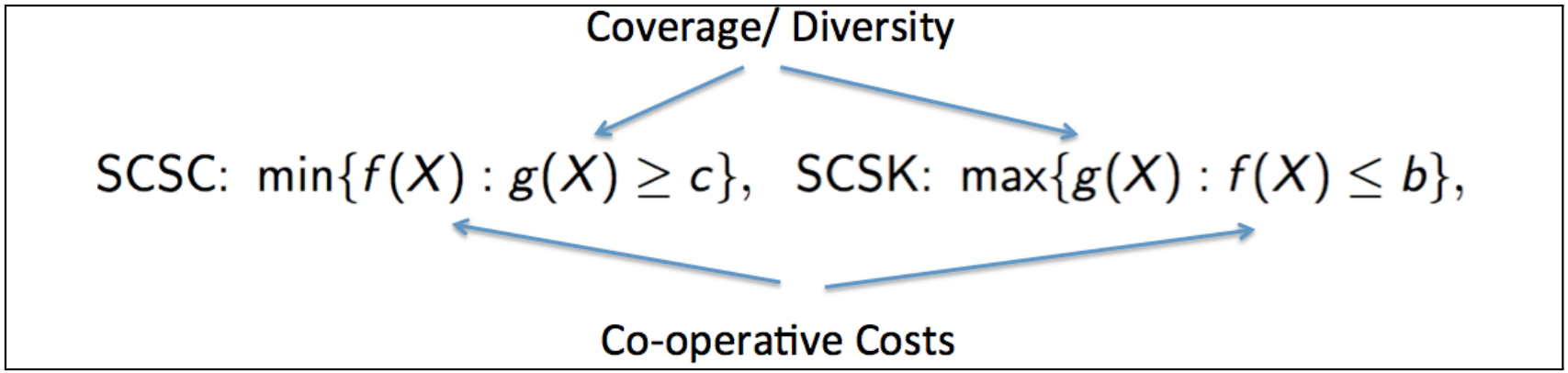
Co-operative Costs

Submodular Optimization with Submodular Constraints

- Historically: DS optimization



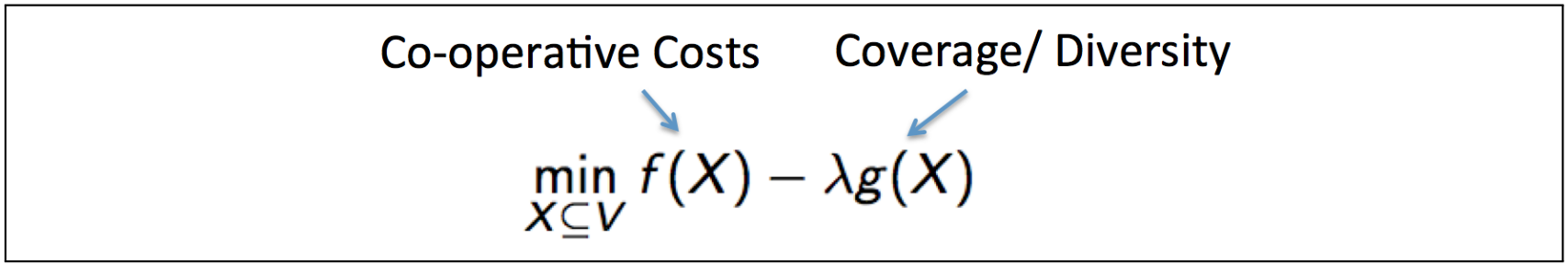
- Unfortunately, NP hard to approximate (Iyer-Bilmes'12).
- We introduce the following, which is often more natual anyway:



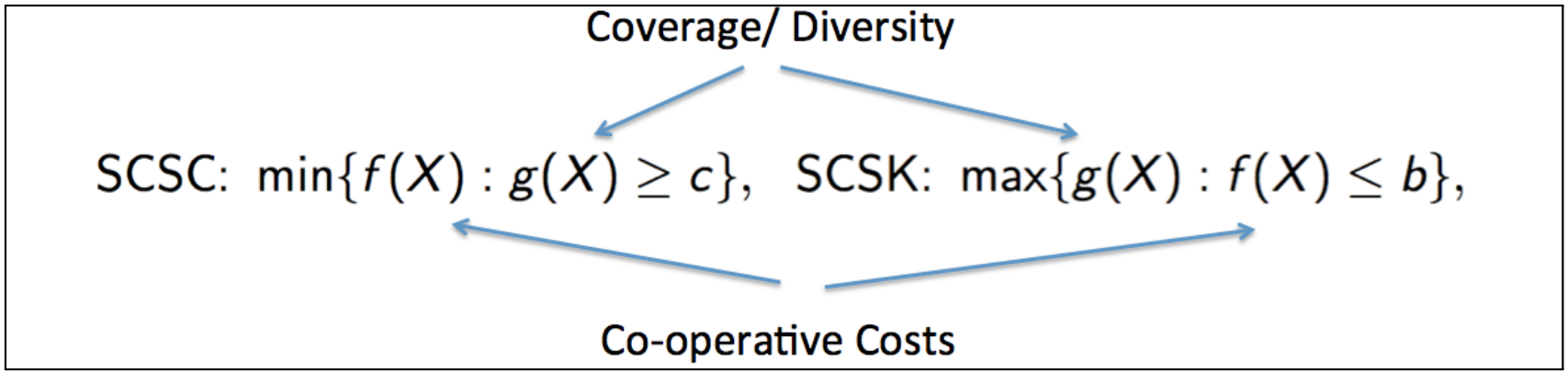
- While DS optimization is NP hard to approximate, SCSC and SCSK however, retain approximation guarantees!

Submodular Optimization with Submodular Constraints

- Historically: DS optimization

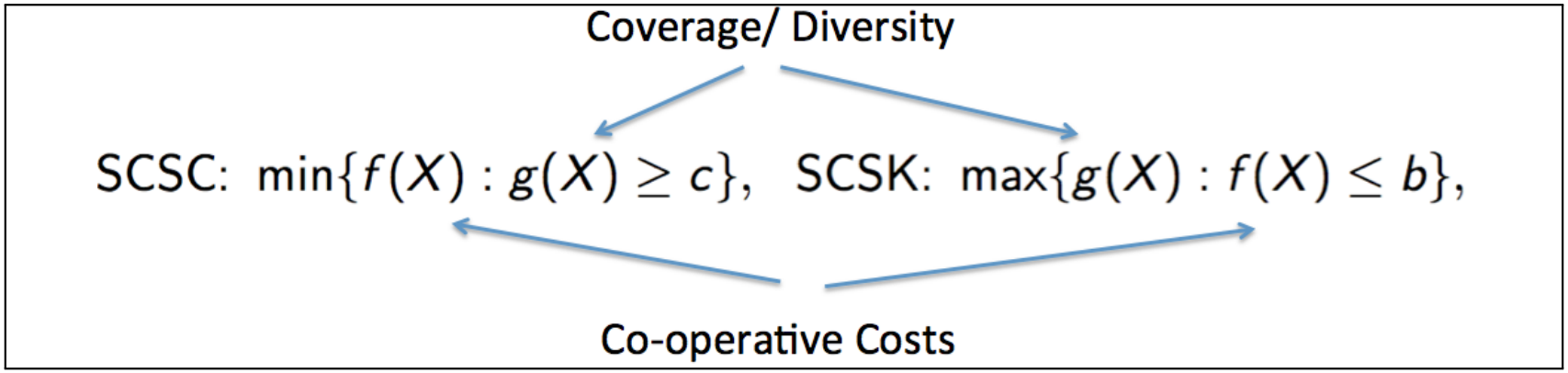


- Unfortunately, NP hard to approximate (Iyer-Bilmes'12).
- We introduce the following, which is often more natual anyway:



- While DS optimization is NP hard to approximate, SCSC and SCSK however, retain approximation guarantees!
- Throughout this talk, assume f and g are monotone.

Our Main Contributions

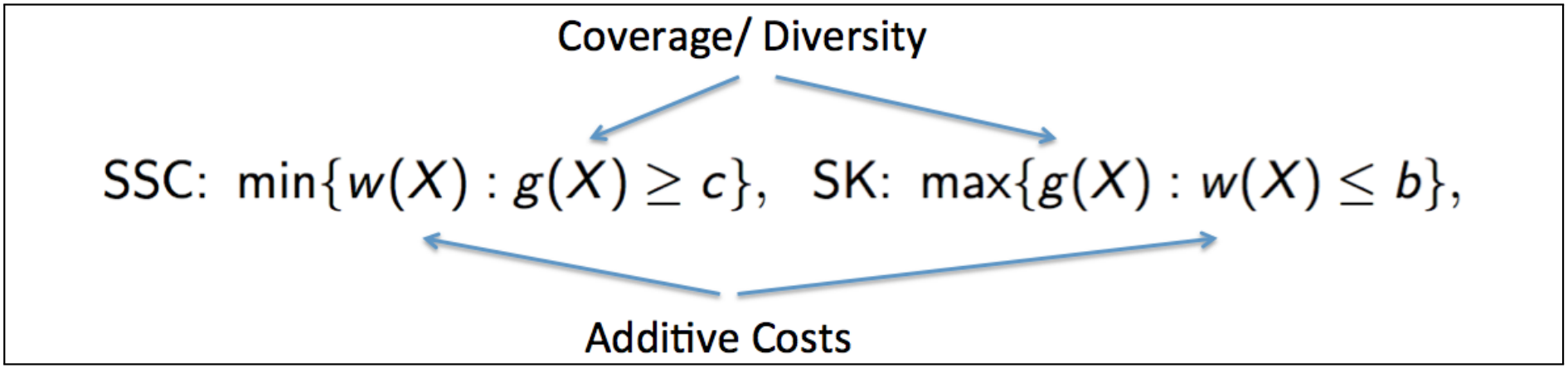


- Show how SCSC/SCSK subsume a number of important optimization problems.
- Provide a unifying algorithmic framework for these.
- Provide a complete characterization of the hardness of these problems.
- Emphasize the scalability and practicality of some of our algorithms!

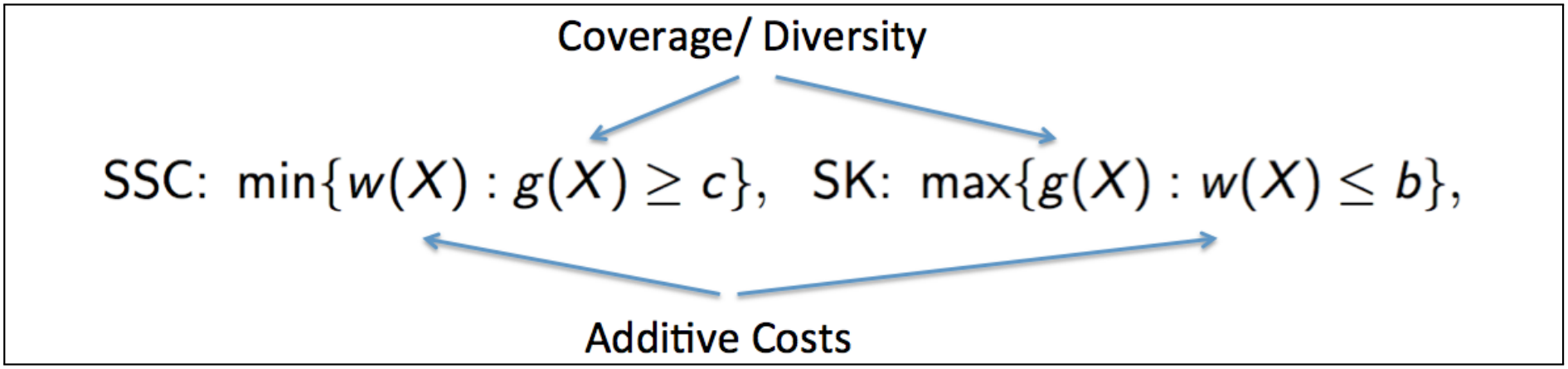
I - Submodular Set Cover and Submodular Knapsack

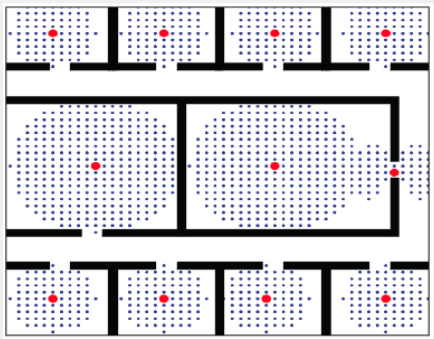
$$\text{SSC: } \min\{w(X) : g(X) \geq c\}, \quad \text{SK: } \max\{g(X) : w(X) \leq b\},$$

I - Submodular Set Cover and Submodular Knapsack



I - Submodular Set Cover and Submodular Knapsack





Sensor Placement
(Krause et al'08)

```
1 all_right how are_you doing
2 how are_you with yours
3 hi nadine my name is lorraine how are_you
4 good how are_you
5 hello hi how are_you
6 good thanks how are_you
7 uh how are_you
8 i'm good how are_you
9 fine how are_you
```

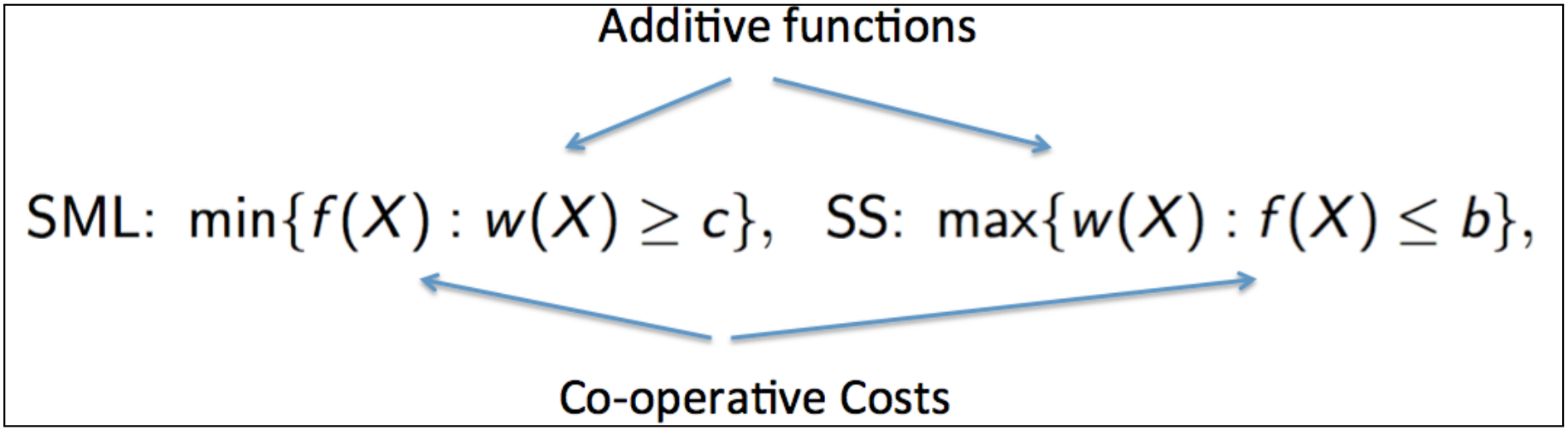
Data Subset Selection
(Wei et al'13)

Document Summarization
(Lin-Bilmes'11)

II - Submodular Cost with Modular Constraints

$$\text{SML: } \min\{f(X) : w(X) \geq c\}, \quad \text{SS: } \max\{w(X) : f(X) \leq b\},$$

II - Submodular Cost with Modular Constraints

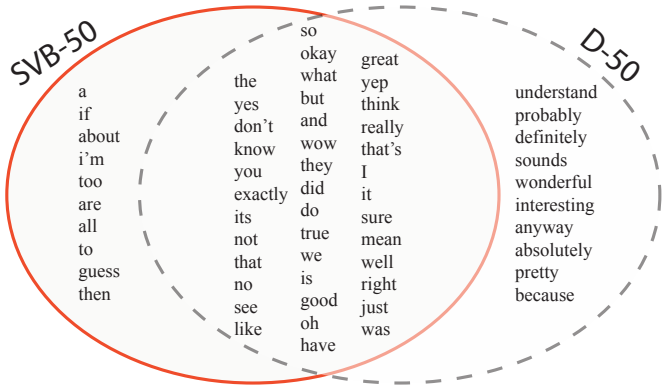


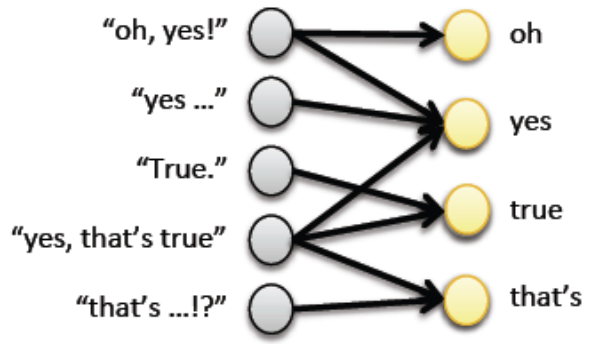
II - Submodular Cost with Modular Constraints

Additive functions

$$\text{SML: } \min\{f(X) : w(X) \geq c\}, \quad \text{SS: } \max\{w(X) : f(X) \leq b\},$$

Co-operative Costs



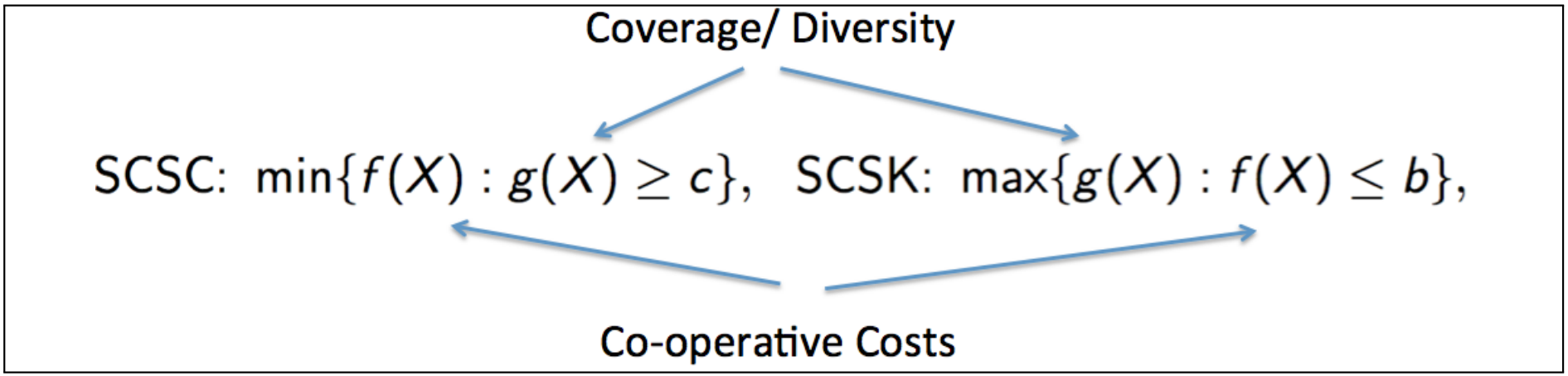


Limited vocabulary speech corpus selection (Lin-Bilmes'11)

III - Most General Case: SCSC and SCSK

$$\text{SCSC: } \min\{f(X) : g(X) \geq c\}, \quad \text{SCSK: } \max\{g(X) : f(X) \leq b\},$$

III - Most General Case: SCSC and SCSK

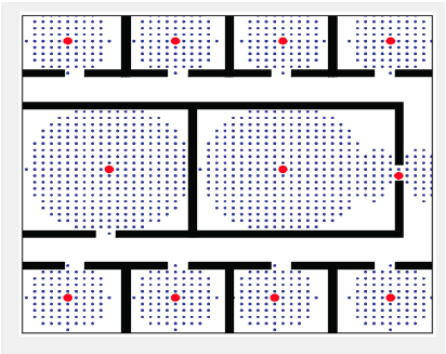


III - Most General Case: SCSC and SCSK

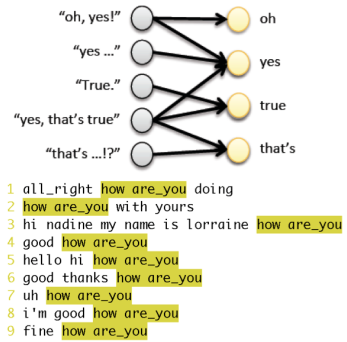
Coverage/ Diversity

$$\text{SCSC: } \min\{f(X) : g(X) \geq c\}, \quad \text{SCSK: } \max\{g(X) : f(X) \leq b\},$$

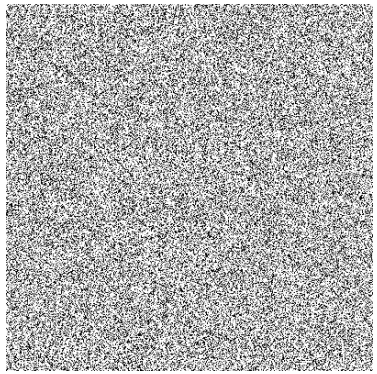
Co-operative Costs



Sensor Placement with Submodular Costs (I-Bilmes'12)



Limited vocabulary and accoustically diverse speech corpus selection (Lin-Bilmes'11, Wei et al'13)



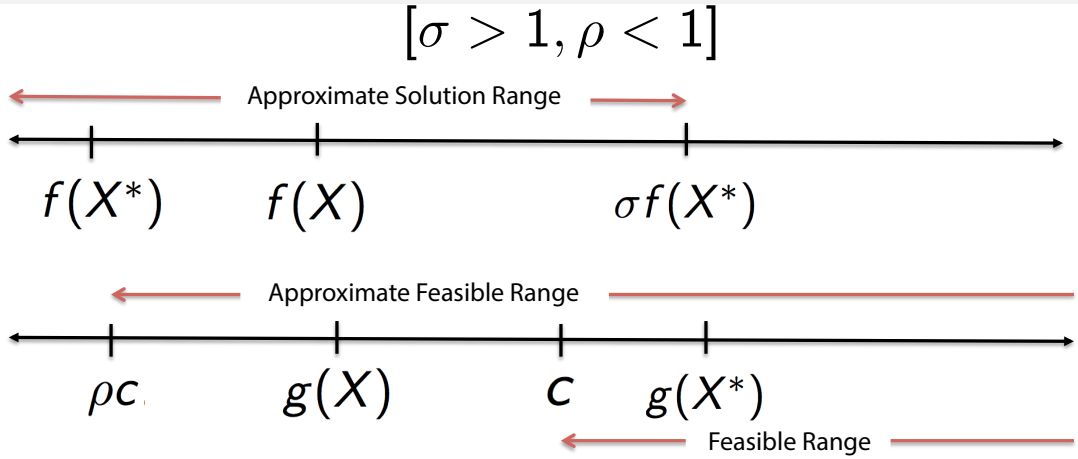
Privacy preserving communication (I-Bilmes'13)

Connections between SCSC and SCSK

- **Bi-criterion factors:**

Connections between SCSC and SCSK

- **Bi-criterion factors:**
- $\min\{f(X) : g(X) \geq c\}$: $[\sigma, \rho]$ approximation for SCSC is a set $X : f(X) \leq \sigma f(X^*)$ and $g(X) \geq \rho c$.

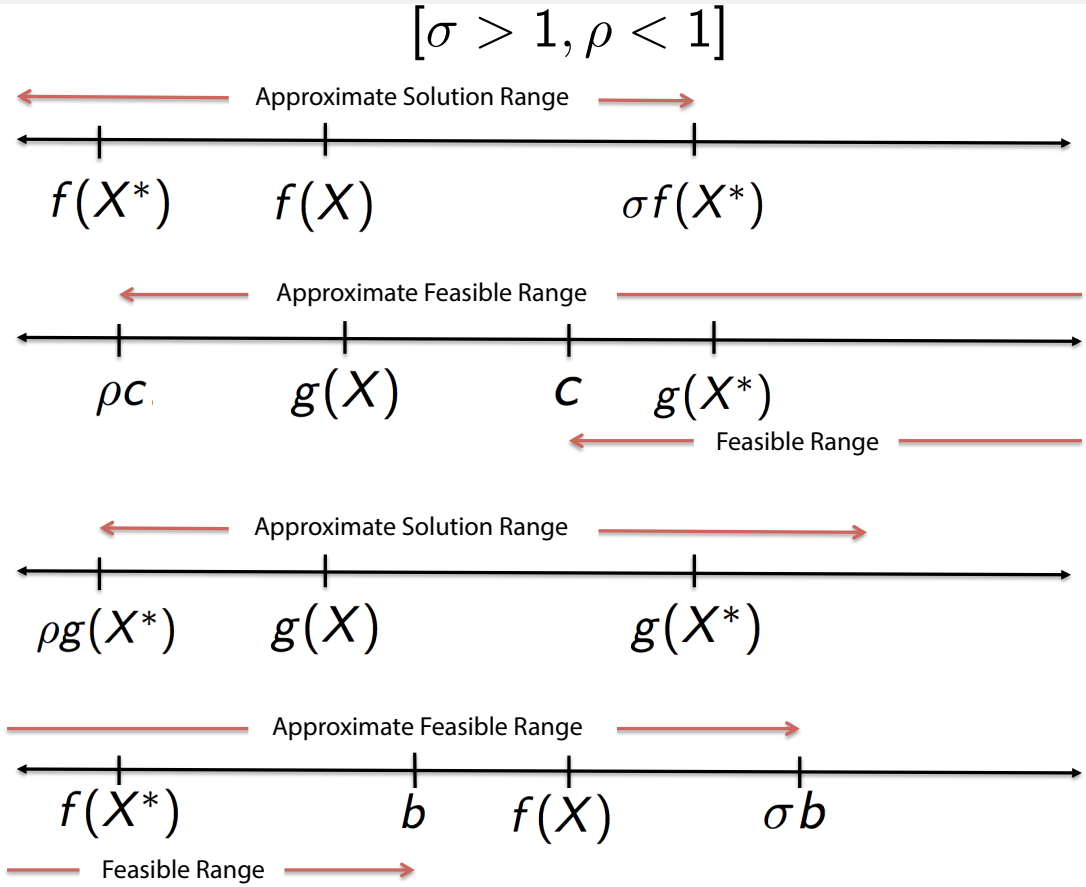


Connections between SCSC and SCSK

- **Bi-criterion factors:**

- $\min\{f(X) : g(X) \geq c\}$: $[\sigma, \rho]$ approximation for SCSC is a set $X : f(X) \leq \sigma f(X^*)$ and $g(X) \geq \rho c$.

- $\max\{g(X) : f(X) \leq b\}$: $[\rho, \sigma]$ approximation for SCSK is a set $X : g(X) \geq \rho g(X^*)$ and $f(X) \leq \sigma b$.

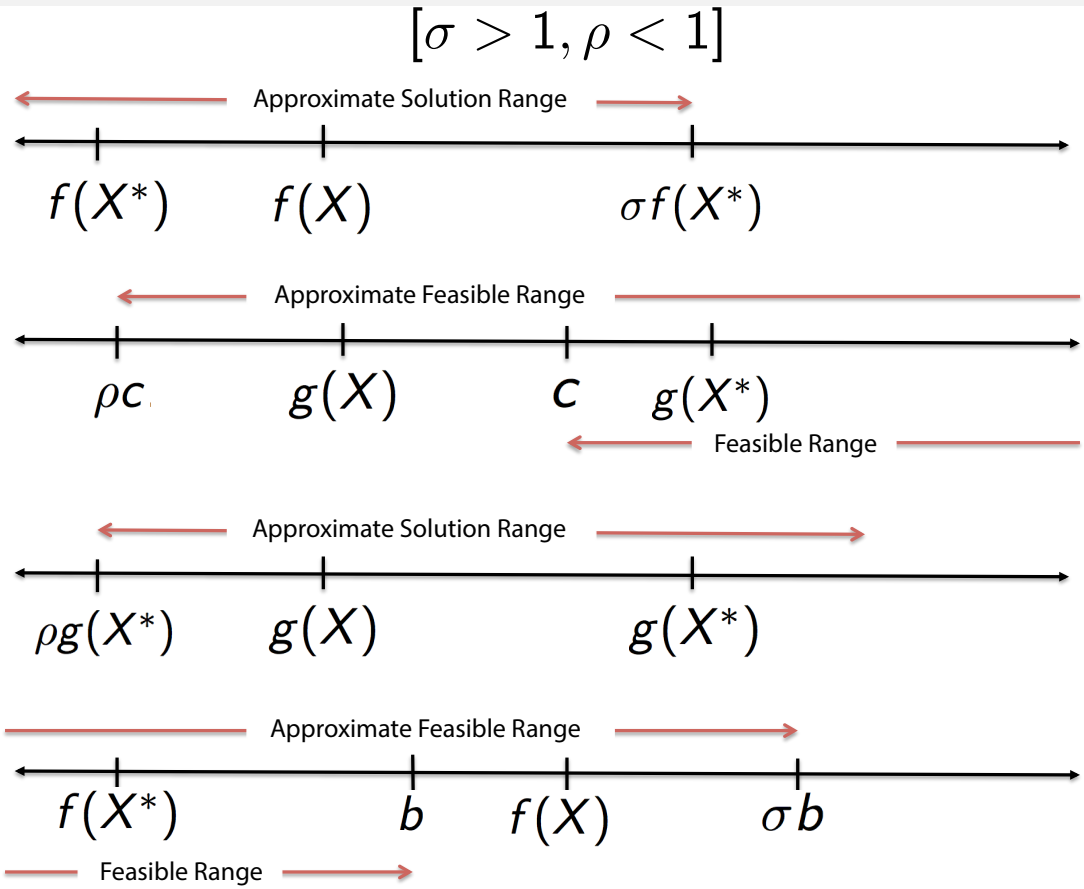


Connections between SCSC and SCSK

- **Bi-criterion factors:**

- $\min\{f(X) : g(X) \geq c\}$: $[\sigma, \rho]$ approximation for SCSC is a set $X : f(X) \leq \sigma f(X^*)$ and $g(X) \geq \rho c$.

- $\max\{g(X) : f(X) \leq b\}$: $[\rho, \sigma]$ approximation for SCSK is a set $X : g(X) \geq \rho g(X^*)$ and $f(X) \leq \sigma b$.

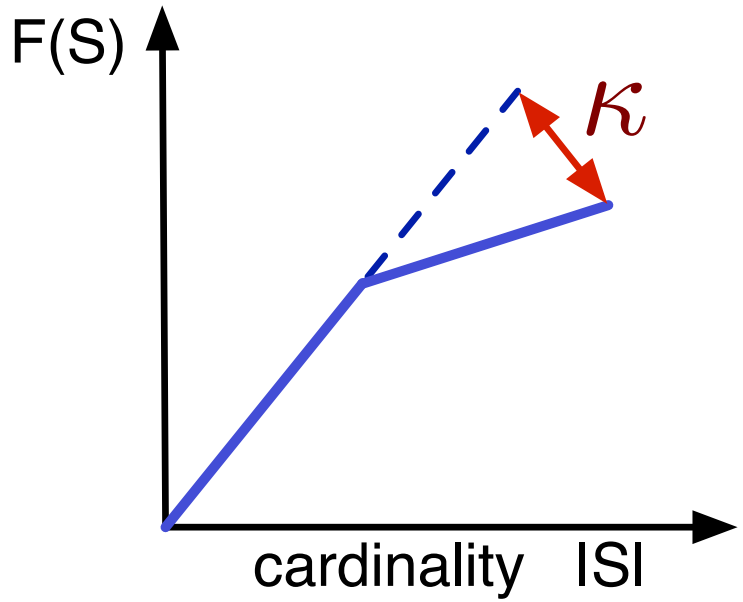


- **Theorem:** Given a $[\sigma, \rho]$ bi-criterion approx. algorithm for SCSC, we can obtain a $[(1+\epsilon)\rho, \sigma]$ bi-criterion approx. algorithm for SCSK, by running the algorithm for SCSC, $O(\log \frac{1}{\epsilon})$ times.
- The other direction also holds!

Curvature of a Submodular Function

- Curvature:

$$\kappa_f = 1 - \min_{j \in V} \frac{f(j|V \backslash j)}{f(j)} \quad \text{and} \quad \kappa_g = 1 - \min_{j \in V} \frac{g(j|V \backslash j)}{g(j)} \tag{2}$$



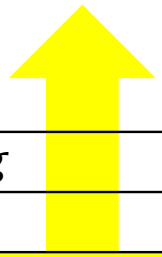
- Curvature is a fundamental "complexity" parameter of a submodular function.

Hardness (Lower bounds) of the problems

	Modular g	Submodular g	
	$(\kappa_g = 0)$	$(0 < \kappa_g < 1)$	$(\kappa_g = 1)$
Modular f $(\kappa_f = 0)$			
Submod f $(0 < \kappa_f < 1)$			
Submod f $(\kappa_f = 1)$			

Hardness (Lower bounds) of the problems

Knapsack



	Modular g	Submodular g	
	$(\kappa_g = 0)$	$(0 < \kappa_g < 1)$	$(\kappa_g = 1)$
Modular f $(\kappa_f = 0)$	FPTAS		
Submod f $(0 < \kappa_f < 1)$			
Submod f $(\kappa_f = 1)$			

Hardness (Lower bounds) of the problems

Knapsack

SSC/SK

	Modular g	Submodular g	
	$(\kappa_g = 0)$	$(0 < \kappa_g < 1)$	$(\kappa_g = 1)$
Modular f $(\kappa_f = 0)$	FPTAS	$\frac{1}{\kappa_g}(1 - e^{-\kappa_g})$	$1 - 1/e$
Submod f $(0 < \kappa_f < 1)$			
Submod f $(\kappa_f = 1)$			

Hardness (Lower bounds) of the problems

Knapsack

SSC/SK

	Modular g	Submodular g	
	$(\kappa_g = 0)$	$(0 < \kappa_g < 1)$	$(\kappa_g = 1)$
Modular f $(\kappa_f = 0)$	FPTAS	$\frac{1}{\kappa_g}(1 - e^{-\kappa_g})$	$1 - 1/e$
Submod f $(0 < \kappa_f < 1)$	$\Omega(\frac{\sqrt{n}}{1+(\sqrt{n}-1)(1-\kappa_f)})$		
Submod f $(\kappa_f = 1)$	$\Omega(\sqrt{n})$		

SML/SS

Hardness (Lower bounds) of the problems

Knapsack

SSC/SK

	Modular g	Submodular g	
	$(\kappa_g = 0)$	$(0 < \kappa_g < 1)$	$(\kappa_g = 1)$
Modular f $(\kappa_f = 0)$	FPTAS	$\frac{1}{\kappa_g}(1 - e^{-\kappa_g})$	$1 - 1/e$
Submod f $(0 < \kappa_f < 1)$	$\Omega(\frac{\sqrt{n}}{1+(\sqrt{n}-1)(1-\kappa_f)})$	$\Omega(\frac{\sqrt{n}}{1+(\sqrt{n}-1)(1-\kappa_f)})$	$\Omega(\frac{\sqrt{n}}{1+(\sqrt{n}-1)(1-\kappa_f)})$
Submod f $(\kappa_f = 1)$	$\Omega(\sqrt{n})$	$\Omega(\sqrt{n})$	$\Omega(\sqrt{n})$

SML/SS

SCSC/SCSK

Hardness (Lower bounds) of the problems

Knapsack

SSC/SK

	Modular g	Submodular g	
	$(\kappa_g = 0)$	$(0 < \kappa_g < 1)$	$(\kappa_g = 1)$
Modular f $(\kappa_f = 0)$	FPTAS	$\frac{1}{\kappa_g}(1 - e^{-\kappa_g})$	$1 - 1/e$
Submod f $(0 < \kappa_f < 1)$	$\Omega(\frac{\sqrt{n}}{1+(\sqrt{n}-1)(1-\kappa_f)})$	$\Omega(\frac{\sqrt{n}}{1+(\sqrt{n}-1)(1-\kappa_f)})$	$\Omega(\frac{\sqrt{n}}{1+(\sqrt{n}-1)(1-\kappa_f)})$
Submod f $(\kappa_f = 1)$	$\Omega(\sqrt{n})$	$\Omega(\sqrt{n})$	$\Omega(\sqrt{n})$

SML/SS

SCSC/SCSK

- Hardness depends (mainly) on κ_f and not (so much) on that of κ_g.

Algorithmic framework

Algorithm 1 General algorithmic framework to address both Problems 1 and 2

Algorithmic framework

Algorithm 1 General algorithmic framework to address both Problems 1 and 2

1: **for** $t = 1, 2, \cdots, T$ **do**

4: **end for**

Algorithmic framework

Algorithm 1 General algorithmic framework to address both Problems 1 and 2

1: **for** $t = 1, 2, \cdots, T$ **do**

2: Choose surrogate functions $\hat{f}_t$ and $\hat{g}_t$ for f and g respectively.

4: **end for**

Algorithmic framework

Algorithm 1 General algorithmic framework to address both Problems 1 and 2

1: **for** $t = 1, 2, \cdots, T$ **do**
2: Choose surrogate functions $\hat{f}_t$ and $\hat{g}_t$ for f and g respectively.
3: Obtain X^t as the optimizer of SCSC/SCSK with $\hat{f}_t$ and $\hat{g}_t$ instead of f and g.
4: **end for**

Algorithmic framework

Algorithm 1 General algorithmic framework to address both Problems 1 and 2

1: **for** $t = 1, 2, \cdots, T$ **do**
2: Choose surrogate functions $\hat{f}_t$ and $\hat{g}_t$ for f and g respectively.
3: Obtain X^t as the optimizer of SCSC/SCSK with $\hat{f}_t$ and $\hat{g}_t$ instead of f and g.
4: **end for**

- Surrogate functions: modular upper/ lower bounds or Ellipsoidal Approximations.

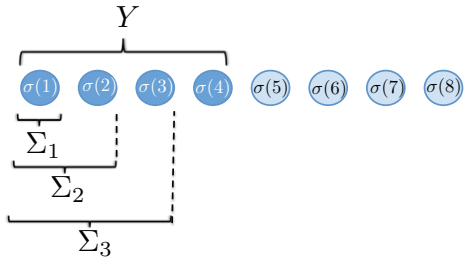
Surrogate functions

- **Modular Lower Bounds:** Induced via orderings of elements:

Surrogate functions

- **Modular Lower Bounds:** Induced via orderings of elements:

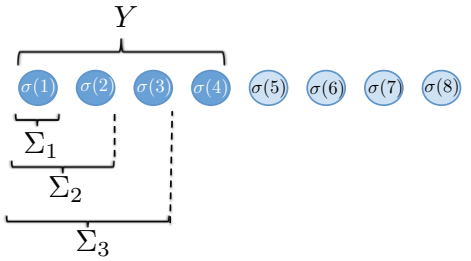
$$f(X) \leq h^{\sigma}_{Y}(X), \text{ where } h^{\sigma}_{Y}(\sigma(i)) = f(\Sigma_i) - f(\Sigma_{i-1})$$



Surrogate functions

- **Modular Lower Bounds:** Induced via orderings of elements:

$$f(X) \leq h^{\sigma}_{Y}(X), \text{ where } h^{\sigma}_{Y}(\sigma(i)) = f(\Sigma_i) - f(\Sigma_{i-1})$$

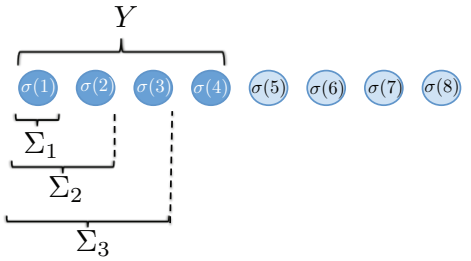


- **Modular upper bounds:**

Surrogate functions

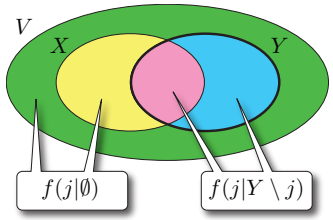
- **Modular Lower Bounds:** Induced via orderings of elements:

$$f(X) \leq h^{\sigma}_{Y}(X), \text{ where } h^{\sigma}_{Y}(\sigma(i)) = f(\Sigma_i) - f(\Sigma_{i-1})$$



- **Modular upper bounds:**
  Upper bound-I

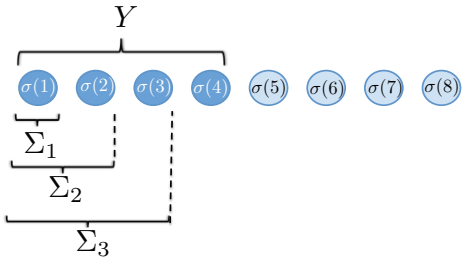
$$f(X) \leq m_{Y,1}(X) = f(Y) - \sum_{j \in Y \setminus X} f(j|Y \backslash j) + \sum_{j \in X \setminus Y} f(j|\emptyset)$$



Surrogate functions

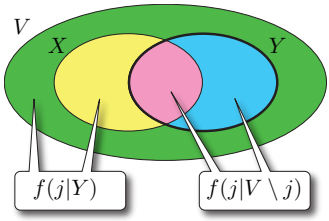
- **Modular Lower Bounds:** Induced via orderings of elements:

$$f(X) \leq h^{\sigma}_{Y}(X), \text{ where } h^{\sigma}_{Y}(\sigma(i)) = f(\Sigma_i) - f(\Sigma_{i-1})$$



- **Modular upper bounds:**
  Upper bound-II

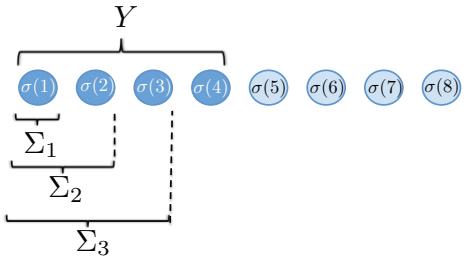
$$f(X) \leq m_{Y,2}(X) = f(Y) - \sum_{j \in Y \setminus X} f(j|V \setminus j) + \sum_{j \in X \setminus Y} f(j|Y)$$



Surrogate functions

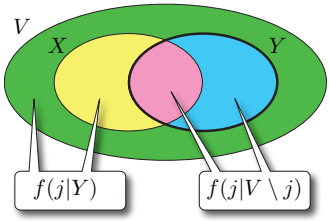
- **Modular Lower Bounds:** Induced via orderings of elements:

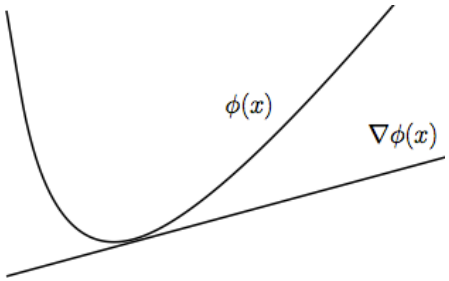
$$f(X) \leq h^{\sigma}_{Y}(X), \text{ where } h^{\sigma}_{Y}(\sigma(i)) = f(\Sigma_i) - f(\Sigma_{i-1})$$

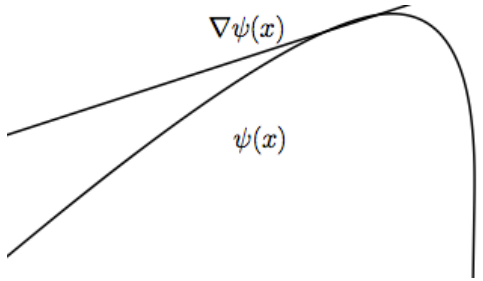


- **Modular upper bounds:**
  Upper bound-II

$$f(X) \leq m_{Y,2}(X) = f(Y) - \sum_{j \in Y \setminus X} f(j|V \setminus j) + \sum_{j \in X \setminus Y} f(j|Y)$$



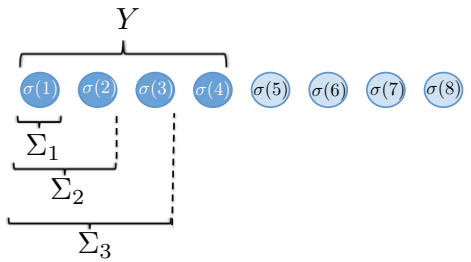




Surrogate functions

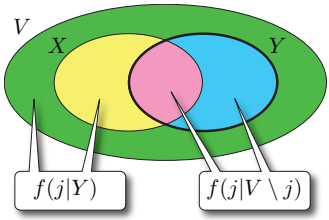
- **Modular Lower Bounds:** Induced via orderings of elements:

$$f(X) \leq h^{\sigma}_{Y}(X), \text{ where } h^{\sigma}_{Y}(\sigma(i)) = f(\Sigma_i) - f(\Sigma_{i-1})$$

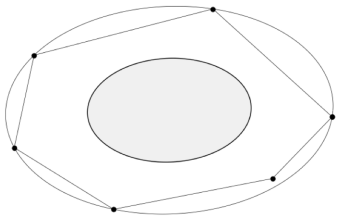


- **Modular upper bounds:**
  Upper bound-II

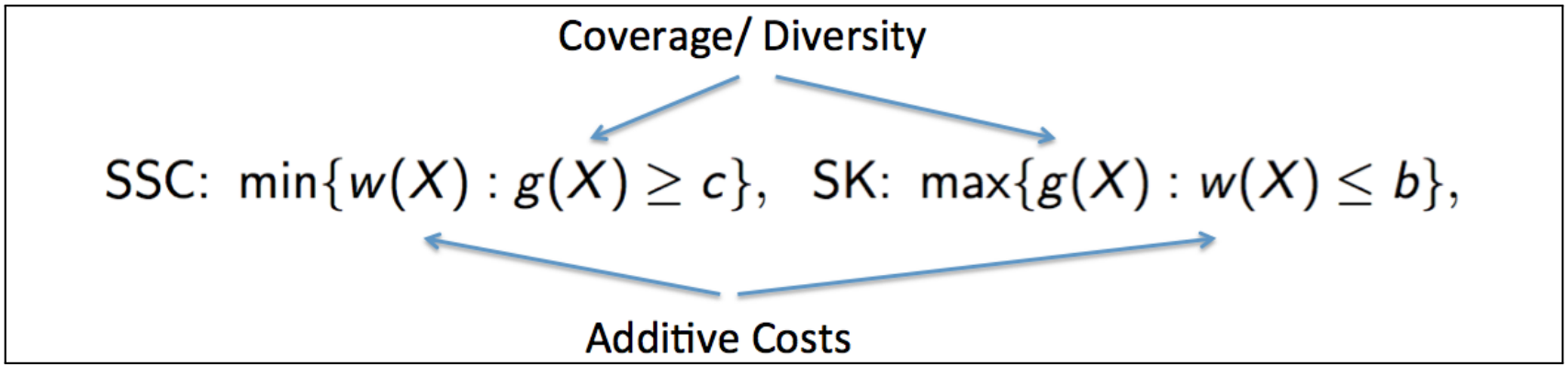
$$f(X) \leq m_{Y,2}(X) = f(Y) - \sum_{j \in Y \setminus X} f(j|V \backslash j) + \sum_{j \in X \setminus Y} f(j|Y)$$



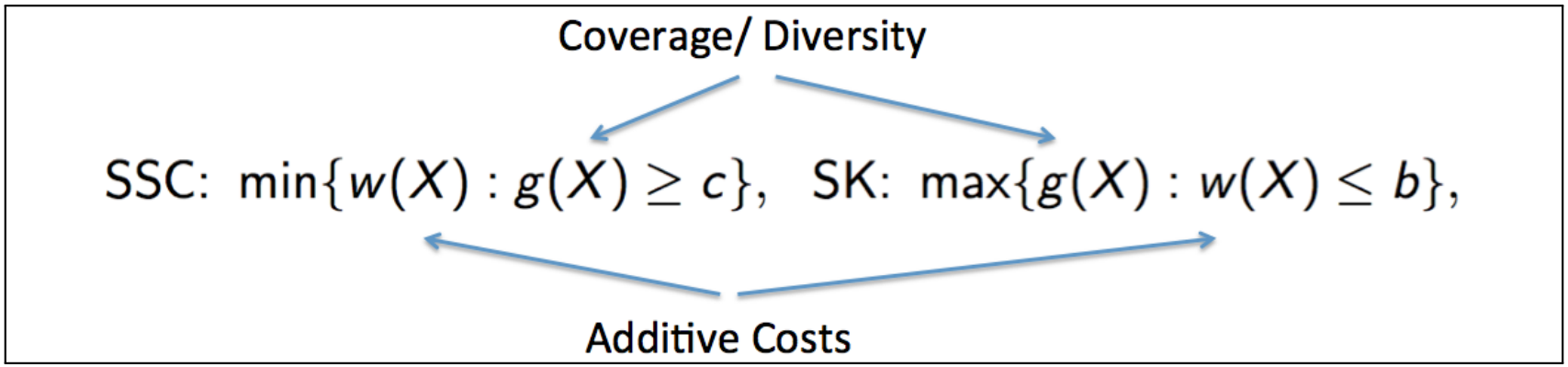
- **Approximations:** Ellipsoidal Approximation gives the *tightest* approximation to a submodular function.



Submodular Set Cover (SSC) and Submodular Knapsack (SK)

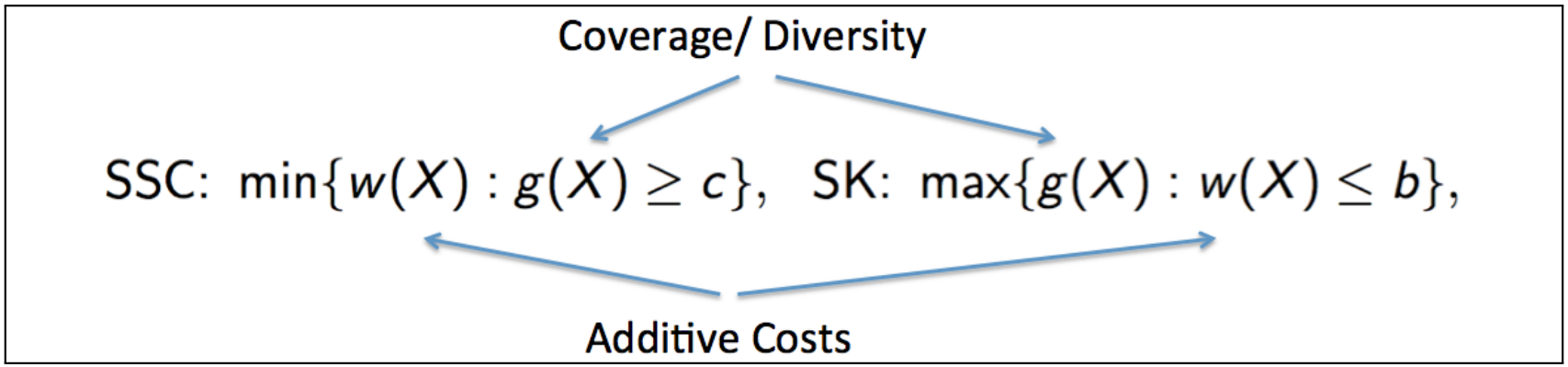


Submodular Set Cover (SSC) and Submodular Knapsack (SK)



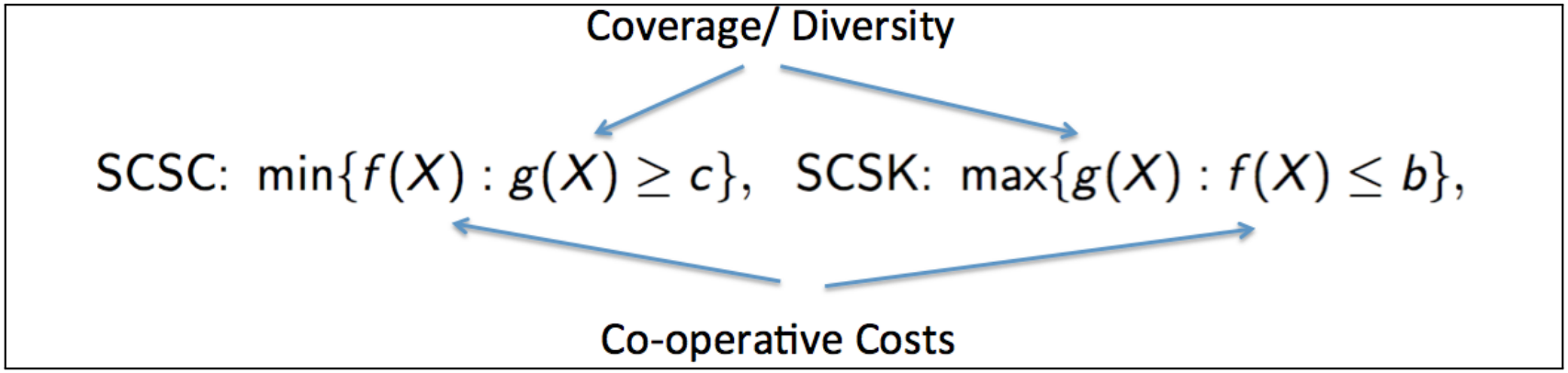
- Lemma: The greedy algorithm for SSC (Wolsey, 82) and SK (Nemhauser, 78) is special case of Algorithm 1 with g replaced by its modular lower bound.

Submodular Set Cover (SSC) and Submodular Knapsack (SK)



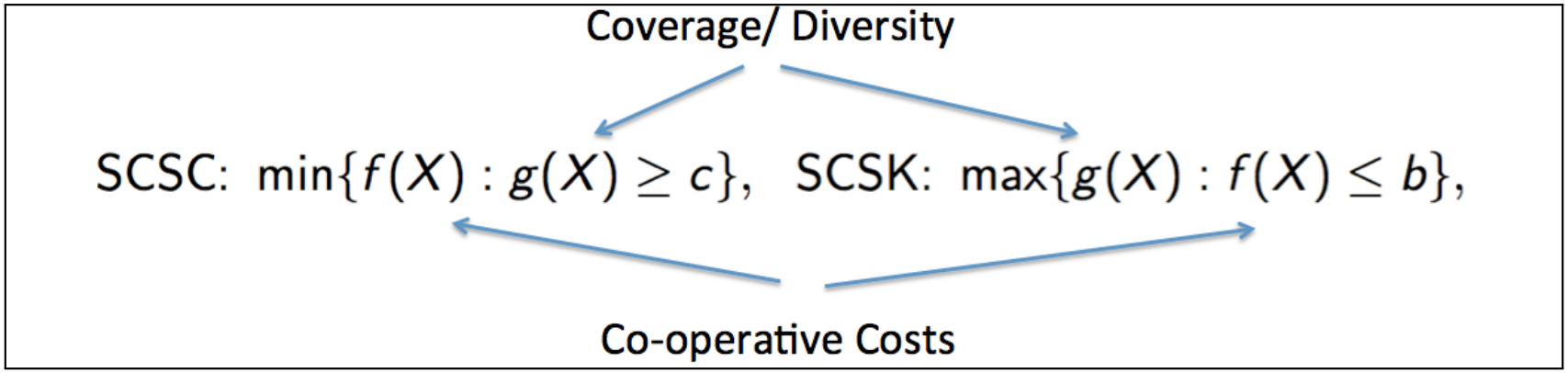
- Lemma: The greedy algorithm for SSC (Wolsey, 82) and SK (Nemhauser, 78) is special case of Algorithm 1 with g replaced by its modular lower bound.
- Approximation guarantees are constant factor – $1 - 1/e$ respectively.

Iterative Submodular Set Cover (ISSC)/Submodular Knapsack (ISK)



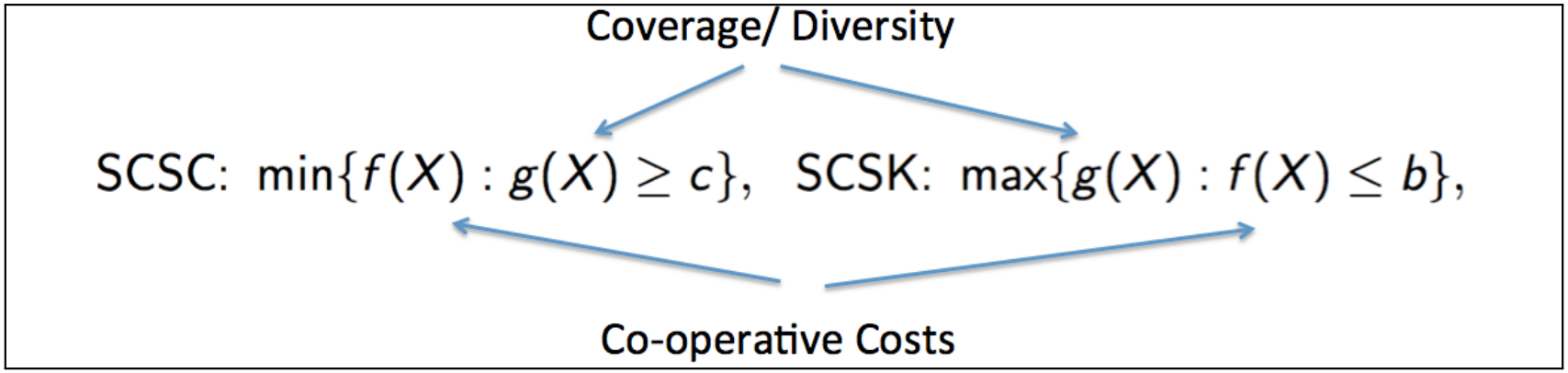
- Choose surrogate functions $\hat{f}_t$ as modular upper bounds.

Iterative Submodular Set Cover (ISSC)/Submodular Knapsack (ISK)



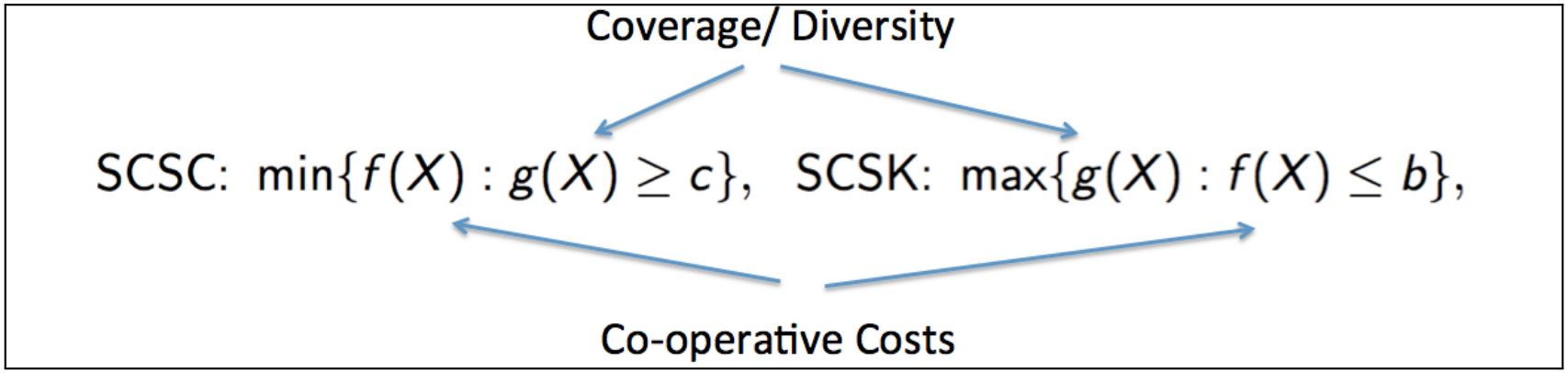
- Choose surrogate functions $\hat{f}_t$ as modular upper bounds.
- Fast iterative algorithms for SCSC and SCSK – Iteratively solve SSC or SK.

Iterative Submodular Set Cover (ISSC)/Submodular Knapsack (ISK)



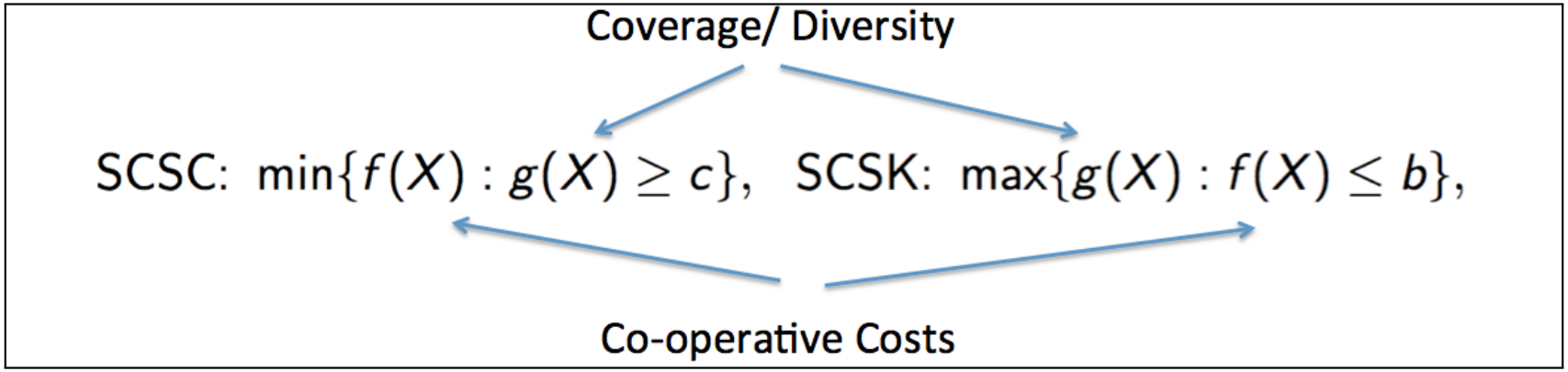
- Choose surrogate functions $\hat{f}_t$ as modular upper bounds.
- Fast iterative algorithms for SCSC and SCSK – Iteratively solve SSC or SK.
- Theorem: ISSC and ISK obtain (bi-criterion) approximation factors $\frac{\sigma}{\rho} = O(\frac{n}{1+(n-1)(1-\kappa_f)})$.

Iterative Submodular Set Cover (ISSC)/Submodular Knapsack (ISK)



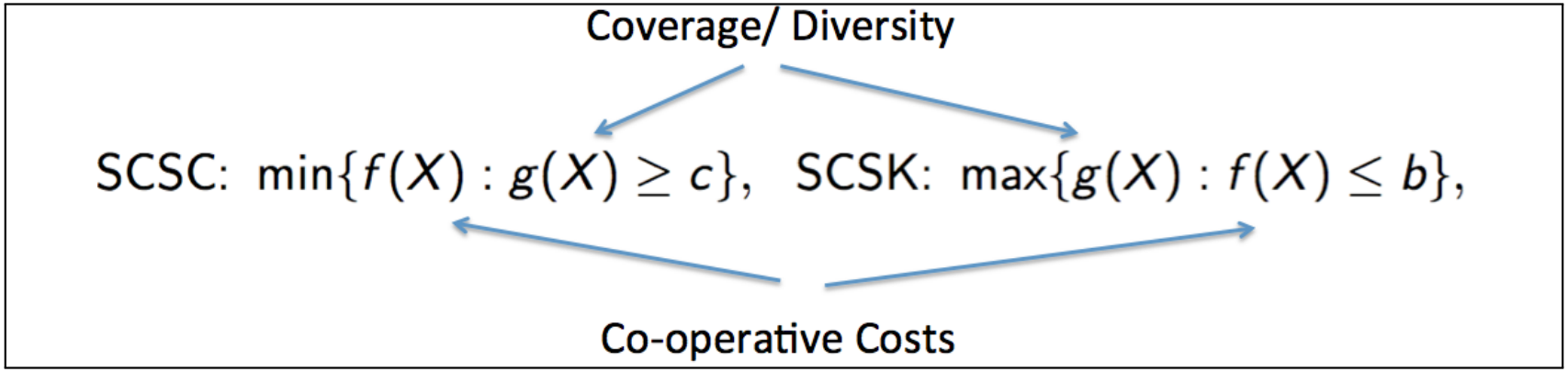
- Choose surrogate functions $\hat{f}_t$ as modular upper bounds.
- Fast iterative algorithms for SCSC and SCSK – Iteratively solve SSC or SK.
- Theorem: ISSC and ISK obtain (bi-criterion) approximation factors $\frac{\sigma}{\rho} = O(\frac{n}{1+(n-1)(1-\kappa_f)})$.
- These algorithms also extend to SML/SS.

Ellipsoidal Approximation Submodular Set Cover (EASSC)/ Submodular Knapsack (EASK)



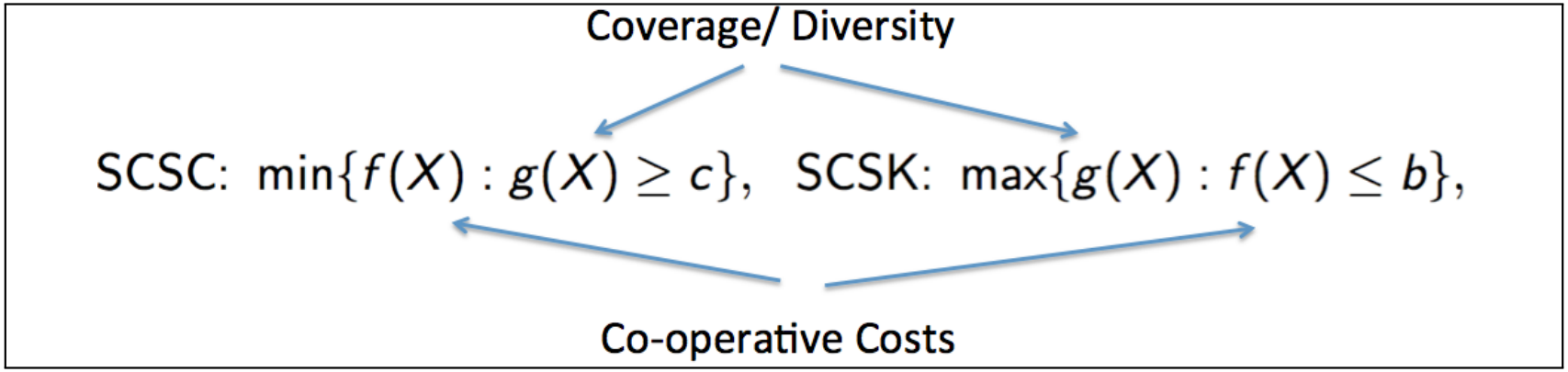
- Choose surrogate functions $\hat{f}_t$ as Ellipsoidal Approximation, in both SCSC and SCSK.

Ellipsoidal Approximation Submodular Set Cover (EASSC)/ Submodular Knapsack (EASK)



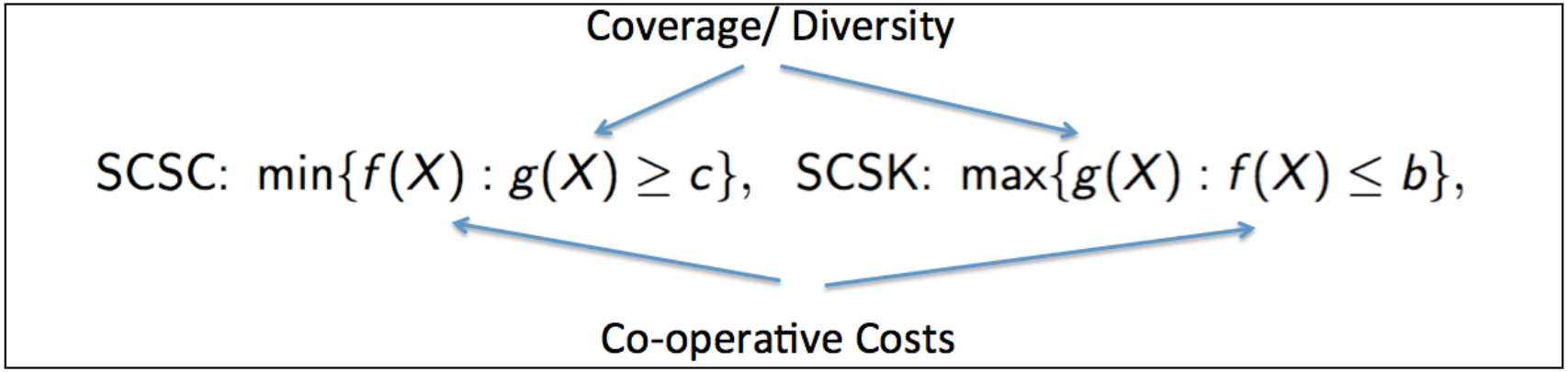
- Choose surrogate functions $\hat{f}_t$ as Ellipsoidal Approximation, in both SCSC and SCSK.
- Theorem: EASSC and EASK obtain (bi-criterion) approximation factors of $\frac{\sigma}{\rho} = O(\frac{\sqrt{n}\log n}{1+(\sqrt{n}\log n - 1)(1-\kappa_f)})$.

Ellipsoidal Approximation Submodular Set Cover (EASSC)/ Submodular Knapsack (EASK)



- Choose surrogate functions $\hat{f}_t$ as Ellipsoidal Approximation, in both SCSC and SCSK.
- Theorem: EASSC and EASK obtain (bi-criterion) approximation factors of $\frac{\sigma}{\rho} = O(\frac{\sqrt{n}\log n}{1+(\sqrt{n}\log n - 1)(1-\kappa_f)})$.
- These algorithms also extend to SML/SS.

Ellipsoidal Approximation Submodular Set Cover (EASSC)/ Submodular Knapsack (EASK)



- Choose surrogate functions $\hat{f}_t$ as Ellipsoidal Approximation, in both SCSC and SCSK.
- Theorem: EASSC and EASK obtain (bi-criterion) approximation factors of $\frac{\sigma}{\rho} = O(\frac{\sqrt{n}\log n}{1+(\sqrt{n}\log n - 1)(1-\kappa_f)})$.
- These algorithms also extend to SML/SS.
- This algorithm matches the hardness of this problem upto log factors.

Limited Vocabulary data subset selection with Accoustic diversity

- **Accoustic Diversity:**

```
1 all_right how are_you doing
2 how are_you with yours
3 hi nadine my name is lorraine how are_you
4 good how are_you
5 hello hi how are_you
6 good thanks how are_you
7 uh how are_you
8 i'm good how are_you
9 fine how are_you
```

Limited Vocabulary data subset selection with Accoustic diversity

- **Accoustic Diversity:**
  - Similarity matrix s_{ij} between utterances i and j (string kernel)

```
1 all_right how are_you doing
2 how are_you with yours
3 hi nadine my name is lorraine how are_you
4 good how are_you
5 hello hi how are_you
6 good thanks how are_you
7 uh how are_you
8 i'm good how are_you
9 fine how are_you
```

Limited Vocabulary data subset selection with Accoustic diversity

- **Accoustic Diversity:**
  - Similarity matrix s_{ij} between utterances i and j (string kernel)
  - Submodular functions:

```
1 all_right how are_you doing
2 how are_you with yours
3 hi nadine my name is lorraine how are_you
4 good how are_you
5 hello hi how are_you
6 good thanks how are_you
7 uh how are_you
8 i'm good how are_you
9 fine how are_you
```

Limited Vocabulary data subset selection with Accoustic diversity

- **Accoustic Diversity:**
  - Similarity matrix s_{ij} between utterances i and j (string kernel)
  - Submodular functions:
    1. Facility Location function: $g(X) = \sum_{i \in V} \max_{j \in X} s_{ij}$

```
1 all_right how are_you doing
2 how are_you with yours
3 hi nadine my name is lorraine how are_you
4 good how are_you
5 hello hi how are_you
6 good thanks how are_you
7 uh how are_you
8 i'm good how are_you
9 fine how are_you
```

Limited Vocabulary data subset selection with Accoustic diversity

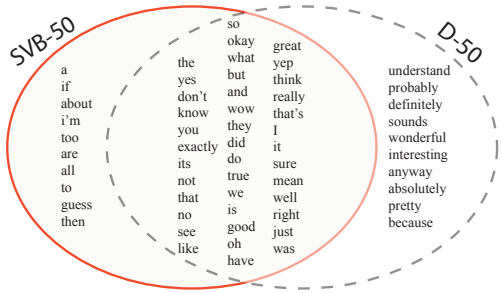
- **Accoustic Diversity:**
  - Similarity matrix s_{ij} between utterances i and j (string kernel)
  - Submodular functions:
    1. Facility Location function: $g(X) = \sum_{i \in V} \max_{j \in X} s_{ij}$
    2. Saturated coverage function $g(X) = \sum_{i \in V} \min\{\sum_{j \in X} s_{ij}, \beta \sum_{j \in V} s_{ij}\}$.

```
1 all_right how are_you doing
2 how are_you with yours
3 hi nadine my name is lorraine how are_you
4 good how are_you
5 hello hi how are_you
6 good thanks how are_you
7 uh how are_you
8 i'm good how are_you
9 fine how are_you
```

Limited Vocabulary data subset selection with Accoustic diversity

- **Accoustic Diversity:**
  - Similarity matrix s_{ij} between utterances i and j (string kernel)
  - Submodular functions:
    1. Facility Location function: $g(X) = \sum_{i \in V} \max_{j \in X} s_{ij}$
    2. Saturated coverage function $g(X) = \sum_{i \in V} \min\{\sum_{j \in X} s_{ij}, \beta \sum_{j \in V} s_{ij}\}$.
- **Limited Vocabulary:**

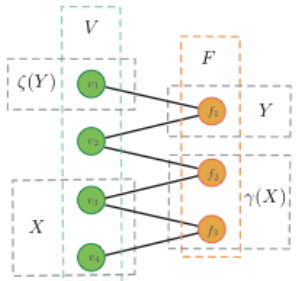
```
1 all_right how are_you doing
2 how are_you with yours
3 hi nadine my name is lorraine how are_you
4 good how are_you
5 hello hi how are_you
6 good thanks how are_you
7 uh how are_you
8 i'm good how are_you
9 fine how are_you
```



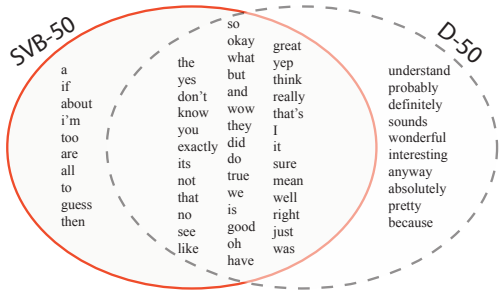
Limited Vocabulary data subset selection with Accoustic diversity

- **Accoustic Diversity:**
  - Similarity matrix s_{ij} between utterances i and j (string kernel)
  - Submodular functions:
    1. Facility Location function: $g(X) = \sum_{i \in V} \max_{j \in X} s_{ij}$
    2. Saturated coverage function $g(X) = \sum_{i \in V} \min\{\sum_{j \in X} s_{ij}, \beta \sum_{j \in V} s_{ij}\}$.
- **Limited Vocabulary:**

```
1 all_right how are_you doing
2 how are_you with yours
3 hi nadine my name is lorraine how are_you
4 good how are_you
5 hello hi how are_you
6 good thanks how are_you
7 uh how are_you
8 i'm good how are_you
9 fine how are_you
```



(a) Bipartite graph

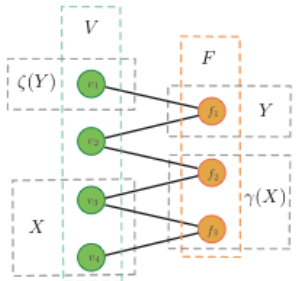


Limited Vocabulary data subset selection with Accoustic diversity

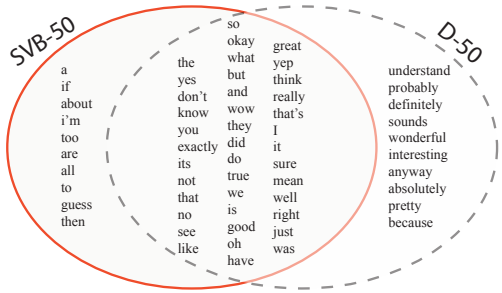
- **Accoustic Diversity:**
  - Similarity matrix s_{ij} between utterances i and j (string kernel)
  - Submodular functions:
    1. Facility Location function: $g(X) = \sum_{i \in V} \max_{j \in X} s_{ij}$
    2. Saturated coverage function $g(X) = \sum_{i \in V} \min\{\sum_{j \in X} s_{ij}, \beta \sum_{j \in V} s_{ij}\}$.

```
1 all_right how are_you doing
2 how are_you with yours
3 hi nadine my name is lorraine how are_you
4 good how are_you
5 hello hi how are_you
6 good thanks how are_you
7 uh how are_you
8 i'm good how are_you
9 fine how are_you
```

- **Limited Vocabulary:**



(a) Bipartite graph



Bipartite Neighborhood function: $|\gamma(X)|$.

Results

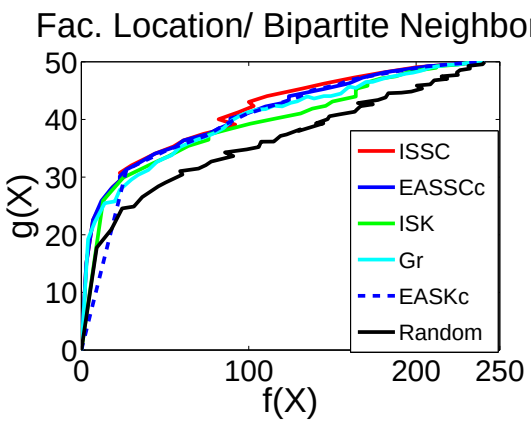
- Compare our different algorithms on the TIMIT speech corpus.

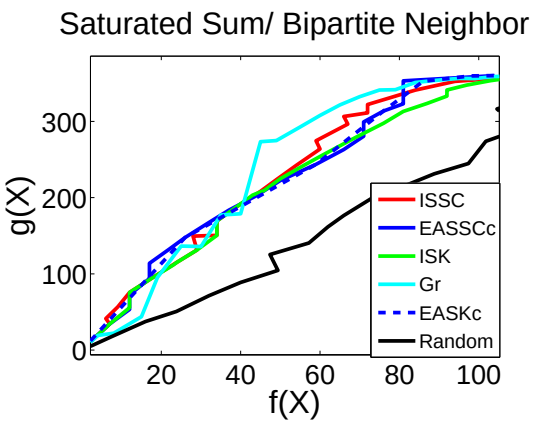
Results

- Compare our different algorithms on the TIMIT speech corpus.
- Baseline is choosing random subsets.

Results

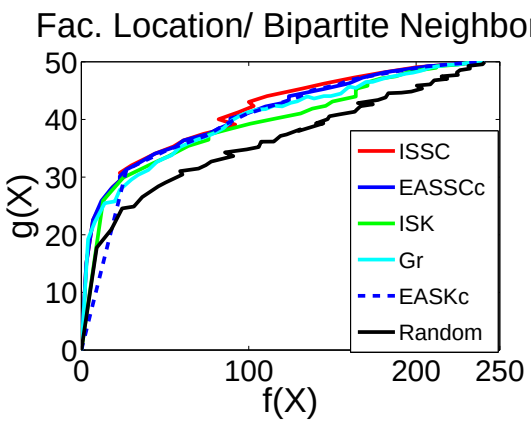
- Compare our different algorithms on the TIMIT speech corpus.
- Baseline is choosing random subsets.
- Observations:

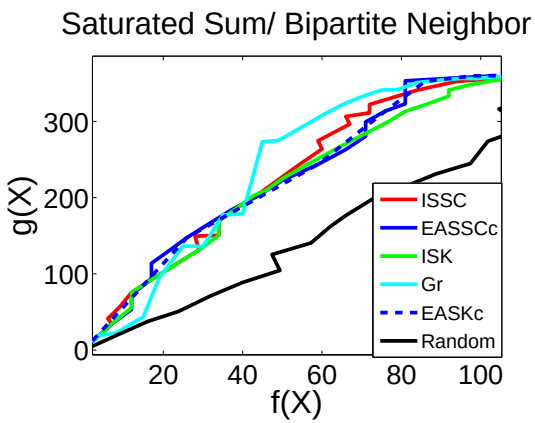




Results

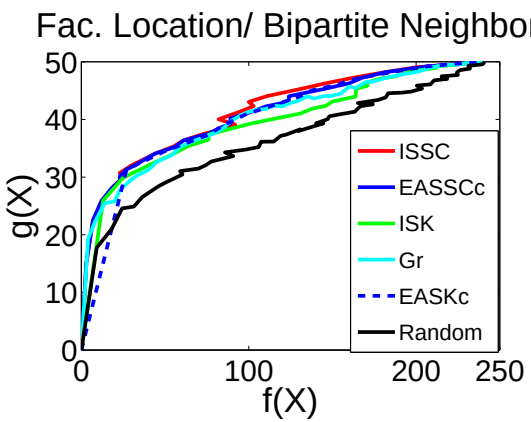
- Compare our different algorithms on the TIMIT speech corpus.
- Baseline is choosing random subsets.
- Observations:
  1. All the algorithms perform much better than random subset selection.

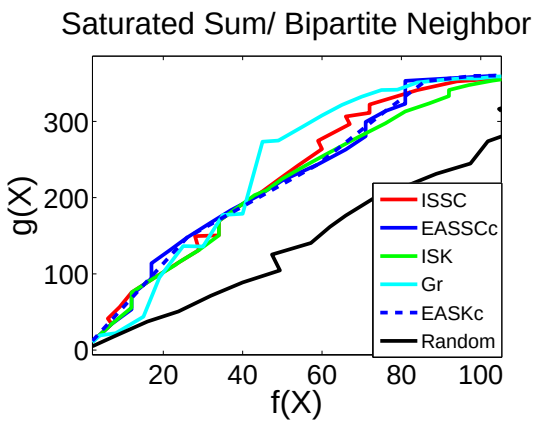




Results

- Compare our different algorithms on the TIMIT speech corpus.
- Baseline is choosing random subsets.
- Observations:
  1. All the algorithms perform much better than random subset selection.
  2. The iterative and much faster algorithms, perform comparably to the slower and tight Ellipsoidal Approximation based algorithms.





Conclusions/ Future Work

- We proposed some very efficient (scalable) algorithms and two tight algorithms for submodular optimization under submodular constraints.
- In the paper: Extensions to handle multiple constraints, and non-monotone submodular functions.
- Future Work: Investigate our new algorithms on different real world applications.

Thank You!

Connections between SCSC and SCSK

- SCSC and SCSK are closely related, and can be transformed into one another!

Connections between SCSC and SCSK

- SCSC and SCSK are closely related, and can be transformed into one another!
- **Bi-criterion factor:** $[\sigma, \rho]$ approximation for (1) $\implies$ a set $X : f(X) \leq \sigma f(X^*)$ and $g(X) \geq \rho c$ $[\sigma > 1, \rho < 1]$.

Connections between SCSC and SCSK

- SCSC and SCSK are closely related, and can be transformed into one another!
- **Bi-criterion factor:** $[\sigma, \rho]$ approximation for (1) $\implies$ a set $X : f(X) \leq \sigma f(X^*)$ and $g(X) \geq \rho c$. A $[\rho, \sigma]$ approximation for (2) $\implies$ a set $X : g(X) \geq \rho g(X^*)$ and $f(X) \leq \sigma b$ $[\sigma > 1, \rho < 1]$.

Connections between SCSC and SCSK

- SCSC and SCSK are closely related, and can be transformed into one another!
- **Bi-criterion factor:** $[\sigma, \rho]$ approximation for (1) $\implies$ a set $X : f(X) \leq \sigma f(X^*)$ and $g(X) \geq \rho c$. A $[\rho, \sigma]$ approximation for (2) $\implies$ a set $X : g(X) \geq \rho g(X^*)$ and $f(X) \leq \sigma b$ $[\sigma > 1, \rho < 1]$.

Algorithm 2 Algorithm for SCSC using an algorithm for SCSK

Connections between SCSC and SCSK

- SCSC and SCSK are closely related, and can be transformed into one another!
- **Bi-criterion factor:** $[\sigma, \rho]$ approximation for (1) $\implies$ a set $X : f(X) \leq \sigma f(X^*)$ and $g(X) \geq \rho c$. A $[\rho, \sigma]$ approximation for (2) $\implies$ a set $X : g(X) \geq \rho g(X^*)$ and $f(X) \leq \sigma b$ $[\sigma > 1, \rho < 1]$.

Algorithm 2 Algorithm for SCSC using an algorithm for SCSK

1: **Input:** An SCSC instance, c, $[\rho, \sigma]$ algorithm for SCSK, $\epsilon > 0$.

Connections between SCSC and SCSK

- SCSC and SCSK are closely related, and can be transformed into one another!
- **Bi-criterion factor:** $[\sigma, \rho]$ approximation for (1) $\implies$ a set $X : f(X) \leq \sigma f(X^*)$ and $g(X) \geq \rho c$. A $[\rho, \sigma]$ approximation for (2) $\implies$ a set $X : g(X) \geq \rho g(X^*)$ and $f(X) \leq \sigma b$ $[\sigma > 1, \rho < 1]$.

Algorithm 2 Algorithm for SCSC using an algorithm for SCSK

1: **Input:** An SCSC instance, c, $[\rho, \sigma]$ algorithm for SCSK, $\epsilon > 0$.
2: **Output:** $[(1 + \epsilon)\sigma, \rho]$ approx. for SCSC.

Connections between SCSC and SCSK

- SCSC and SCSK are closely related, and can be transformed into one another!
- **Bi-criterion factor:** $[\sigma, \rho]$ approximation for (1) $\implies$ a set $X : f(X) \leq \sigma f(X^*)$ and $g(X) \geq \rho c$. A $[\rho, \sigma]$ approximation for (2) $\implies$ a set $X : g(X) \geq \rho g(X^*)$ and $f(X) \leq \sigma b$ $[\sigma > 1, \rho < 1]$.

Algorithm 2 Algorithm for SCSC using an algorithm for SCSK

1: **Input:** An SCSC instance, c, $[\rho, \sigma]$ algorithm for SCSK, $\epsilon > 0$.
2: **Output:** $[(1+\epsilon)\sigma, \rho]$ approx. for SCSC.
3: $b \leftarrow \text{argmin}_j f(j), \hat{X}_b \leftarrow \emptyset$.

Connections between SCSC and SCSK

- SCSC and SCSK are closely related, and can be transformed into one another!
- **Bi-criterion factor:** $[\sigma, \rho]$ approximation for (1) $\implies$ a set $X : f(X) \leq \sigma f(X^*)$ and $g(X) \geq \rho c$. A $[\rho, \sigma]$ approximation for (2) $\implies$ a set $X : g(X) \geq \rho g(X^*)$ and $f(X) \leq \sigma b$ $[\sigma > 1, \rho < 1]$.

Algorithm 2 Algorithm for SCSC using an algorithm for SCSK

1: **Input:** An SCSC instance, c, $[\rho, \sigma]$ algorithm for SCSK, $\epsilon > 0$.
2: **Output:** $[(1+\epsilon)\sigma, \rho]$ approx. for SCSC.
3: $b \leftarrow \text{argmin}_j f(j), \hat{X}_b \leftarrow \emptyset$.
4: **while** $g(\hat{X}_b) < \rho c$ **do**

7: **end while**

Connections between SCSC and SCSK

- SCSC and SCSK are closely related, and can be transformed into one another!
- **Bi-criterion factor:** $[\sigma, \rho]$ approximation for (1) $\implies$ a set $X : f(X) \leq \sigma f(X^*)$ and $g(X) \geq \rho c$. A $[\rho, \sigma]$ approximation for (2) $\implies$ a set $X : g(X) \geq \rho g(X^*)$ and $f(X) \leq \sigma b$ $[\sigma > 1, \rho < 1]$.

Algorithm 2 Algorithm for SCSC using an algorithm for SCSK

1: **Input:** An SCSC instance, c, $[\rho, \sigma]$ algorithm for SCSK, $\epsilon > 0$.
2: **Output:** $[(1+\epsilon)\sigma, \rho]$ approx. for SCSC.
3: $b \leftarrow \text{argmin}_j f(j), \hat{X}_b \leftarrow \emptyset$.
4: **while** $g(\hat{X}_b) < \rho c$ **do**
5: $\quad b \leftarrow (1+\epsilon)b$
6: $\quad \hat{X}_b \leftarrow [\rho, \sigma]$ approx. for SCSK using b.
7: **end while**

Connections between SCSC and SCSK

- SCSC and SCSK are closely related, and can be transformed into one another!
- **Bi-criterion factor:** $[\sigma, \rho]$ approximation for (1) $\implies$ a set $X : f(X) \leq \sigma f(X^*)$ and $g(X) \geq \rho c$. A $[\rho, \sigma]$ approximation for (2) $\implies$ a set $X : g(X) \geq \rho g(X^*)$ and $f(X) \leq \sigma b$ $[\sigma > 1, \rho < 1]$.

Algorithm 2 Algorithm for SCSC using an algorithm for SCSK

1: **Input:** An SCSC instance, c, $[\rho, \sigma]$ algorithm for SCSK, $\epsilon > 0$.
2: **Output:** $[(1+\epsilon)\sigma, \rho]$ approx. for SCSC.
3: $b \leftarrow \text{argmin}_j f(j), \hat{X}_b \leftarrow \emptyset$.
4: **while** $g(\hat{X}_b) < \rho c$ **do**
5: $\quad b \leftarrow (1+\epsilon)b$
6: $\quad \hat{X}_b \leftarrow [\rho, \sigma]$ approx. for SCSK using b.
7: **end while**
8: Return $\hat{X}_b$.

Hardness Theorem

- Theorem: For any $\kappa > 0$, there exists submodular function f with curvature $\kappa_f = \kappa$ such that no polynomial time algorithm for SCSC and SCSK $\frac{\sigma}{\rho} = \frac{n^{1/2-\epsilon}}{1+(n^{1/2-\epsilon}-1)(1-\kappa)}$ for any $\epsilon > 0$.

Hardness Theorem

- Theorem: For any $\kappa > 0$, there exists submodular function f with curvature $\kappa_f = \kappa$ such that no polynomial time algorithm for SCSC and SCSK $\frac{\sigma}{\rho} = \frac{n^{1/2-\epsilon}}{1+(n^{1/2-\epsilon}-1)(1-\kappa)}$ for any $\epsilon > 0$.
- Hardness depends on the curvature of the submodular function f and not on that of g.

Hardness Theorem

- Theorem: For any $\kappa > 0$, there exists submodular function f with curvature $\kappa_f = \kappa$ such that no polynomial time algorithm for SCSC and SCSK $\frac{\sigma}{\rho} = \frac{n^{1/2-\epsilon}}{1+(n^{1/2-\epsilon}-1)(1-\kappa)}$ for any $\epsilon > 0$.
- Hardness depends on the curvature of the submodular function f and not on that of g.

	Modular g	Submodular g	
	$(\kappa_g = 0)$	$(0 < \kappa_g < 1)$	$(\kappa_g = 1)$
Modular f $(\kappa_f = 0)$	FPTAS	$\frac{1}{\kappa_g}(1 - e^{-\kappa_g})$	$1 - 1/e$
Submod f $(0 < \kappa_f < 1)$	$\Omega(\frac{\sqrt{n}}{1+(\sqrt{n}-1)(1-\kappa_f)})$	$\Omega(\frac{\sqrt{n}}{1+(\sqrt{n}-1)(1-\kappa_f)})$	$\Omega(\frac{\sqrt{n}}{1+(\sqrt{n}-1)(1-\kappa_f)})$
Submod f $(\kappa_f = 1)$	$\Omega(\sqrt{n})$	$\Omega(\sqrt{n})$	$\Omega(\sqrt{n})$

Table : Summary of Hardness results for SCSC/ SCSK