# Submodularity for Distributed Sensing Problems

Zeyn Saigol

IR Lab, School of Computer Science

University of Birmingham

6th July 2010

# Outline

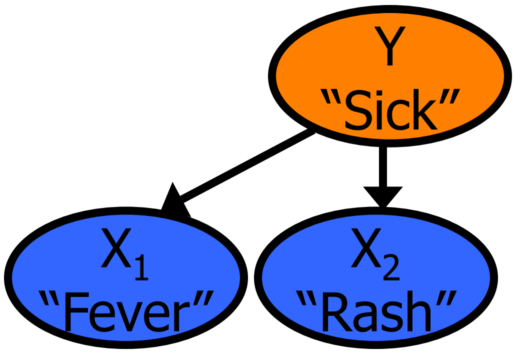
1. What is submodularity?
2. Non-myopic maximisation of information gain
3. Path planning
4. Robust maximisation and minimisation
5. Summary

* These slides are borrowed from Andreas Krause and Carlos Guestrin (see http://www.submodularity.org/)

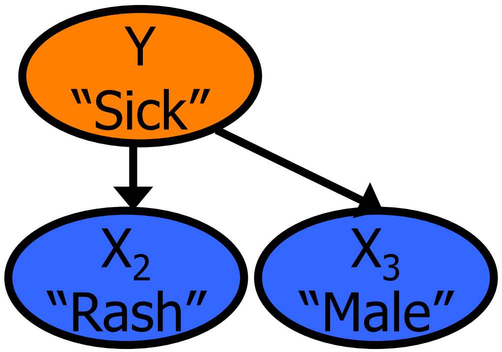
# Set functions

- Submodularity in AI has been popularised by Andreas Krause and Carlos Guestrin
- Finite set $V = \{1,2,\ldots,n\}$
- Function $F: 2^V \Rightarrow R$
- Example: $F(A) = IG(X_A; Y) = H(Y) - H(Y \mid X_A)$

$$= \sum_{y,x_A} P(x_A) [\log P(y \mid x_A) - \log P(y)]$$




$F(\{X_1,X_2\}) = 0.9$

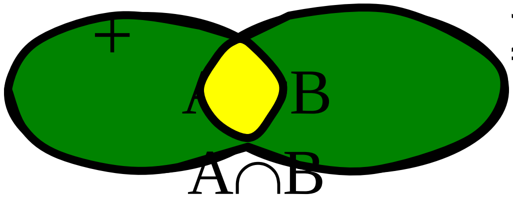



$F(\{X_2,X_3\}) = 0.5$

# Submodular set functions

- Set function F on V is called submodular if

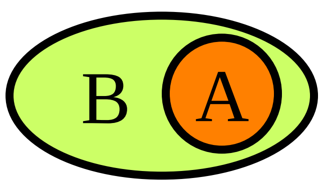
$$\text{For all } A,B \subseteq V: F(A)+F(B) \geq F(A\cup B)+F(A\cap B)$$




+ ≥ +

- Equivalent diminishing returns characterization:

**Submodularity:**



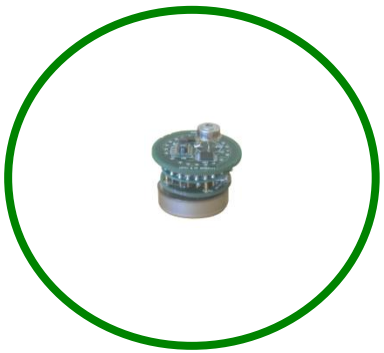

+ •S Large improvement

+ •S Small improvement

$$\text{For } A\subseteq B,\ s\notin B,\ F(A\cup\{s\}) - F(A) \geq F(B\cup\{s\}) - F(B)$$

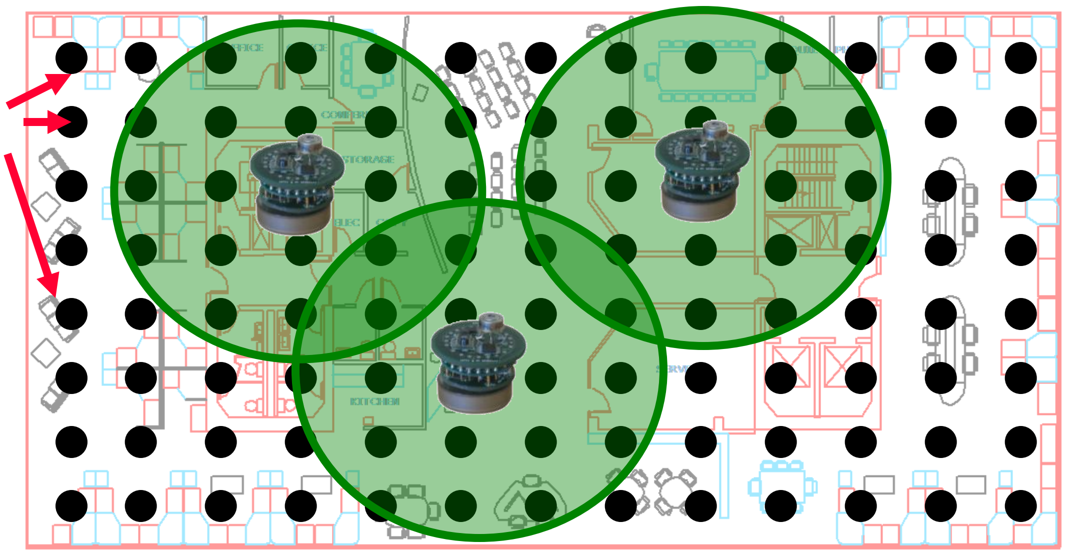
# Example problem: sensor coverage

Place sensors in building



Node predicts values of positions with some radius

Possible locations V



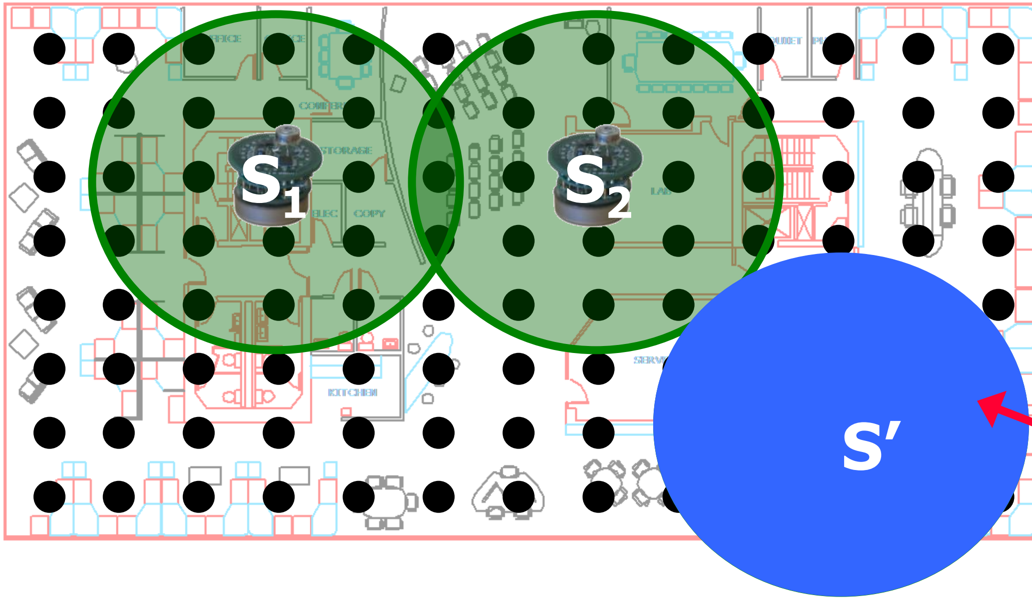
For $A \subseteq V$: $F(A)$ = “area covered by sensors placed at A”

Formally:

W finite set, collection of n subsets $S_i \subseteq W$

For $A \subseteq V=\{1,\ldots,n\}$ define $F(A) = |\bigcup_{i \in A} S_i|$

# Set coverage is submodular

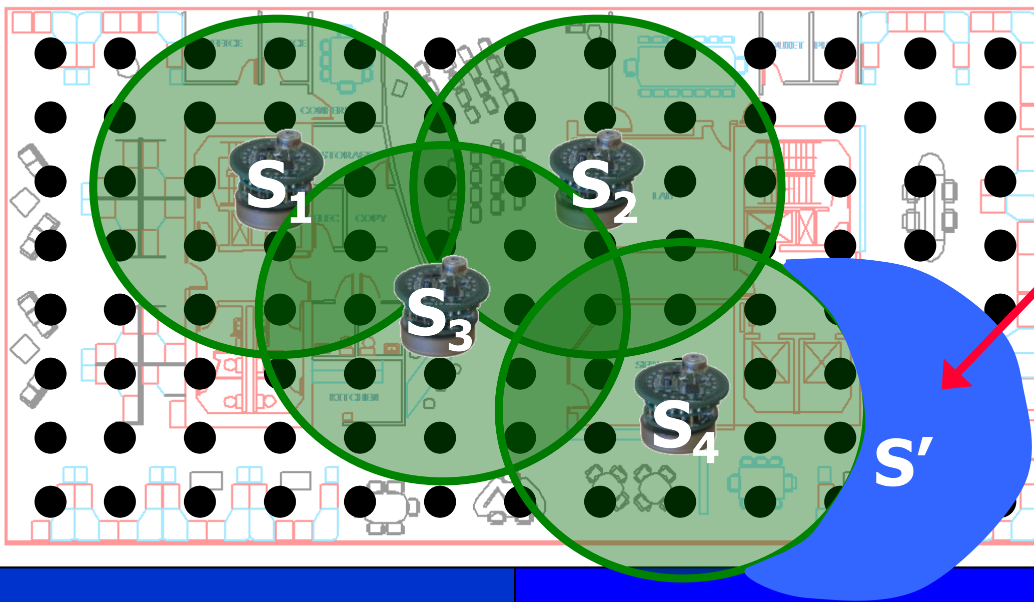



$A=\{S_1,S_2\}$

$F(A\cup\{S'\})-F(A)$

$\geq$

$F(B\cup\{S'\})-F(B)$




$B = \{S_1,S_2,S_3,S_4\}$

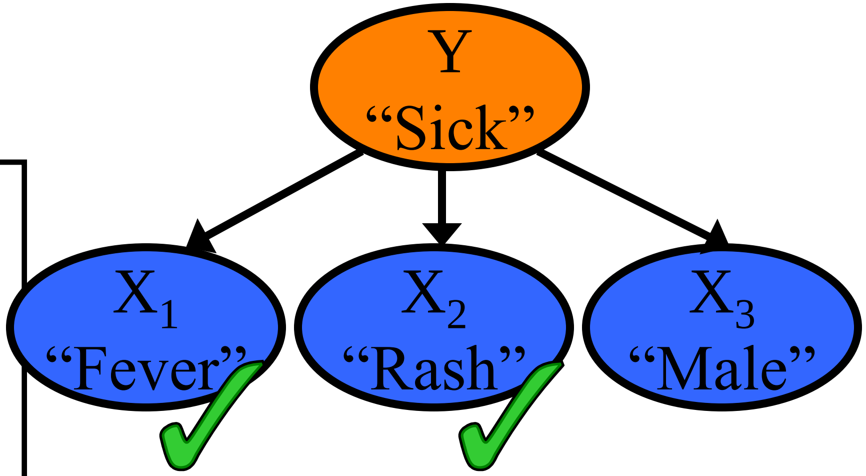
# Approximate maximization

- Given: finite set V, monotonic submodular function F(A)

  Want: $A^* \subseteq V$ such that

$$\mathcal{A}^* = \operatorname*{argmax}_{|\mathcal{A}| \leq k} F(\mathcal{A})$$

**NP-hard!**




Greedy algorithm: M

Start with $A_0 = \{\}$;

For i = 1 to k

$s_i := \operatorname{argmax}_s F(A_{i-1} \cup \{s\}) - F(A_{i-1})$

$A_i := A_{i-1} \cup \{s_i\}$

# Guarantee on greedy algorithm

**Theorem** [Nemhauser et al '78]

Given a monotonic submodular function F, $F(\emptyset)=0$, the greedy maximization algorithm returns $A_{greedy}$

$$F(A_{greedy}) \geq (1-1/e) \max_{|A| \leq k} F(A)$$

~63%

**Sidenote**: Greedy algorithm gives 1/2 approximation for maximization over any matroid C! [Fisher et al '78]

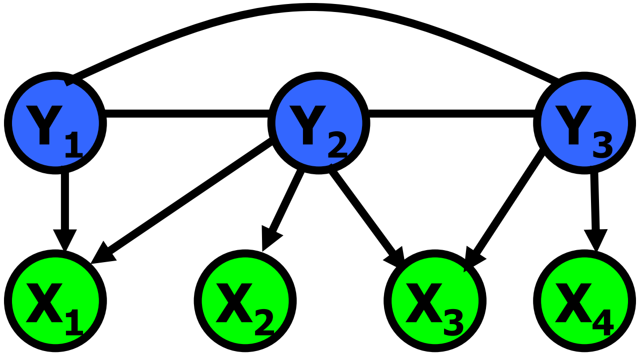
# Example: Submodularity of info-gain

$Y_1, \ldots, Y_m, X_1, \ldots, X_n$ discrete RVs

$F(A) = IG(Y; X_A) = H(Y) - H(Y \mid X_A)$

- F(A) is always monotonic
- However, NOT always submodular

**Theorem** [Krause & Guestrin UAI' 05]
If $X_i$ are all conditionally independent given Y, then F(A) is submodular!




Hence, greedy algorithm works!

In fact, NO practical algorithm can do better than (1-1/e) approximation!

# Information gain with *cost*

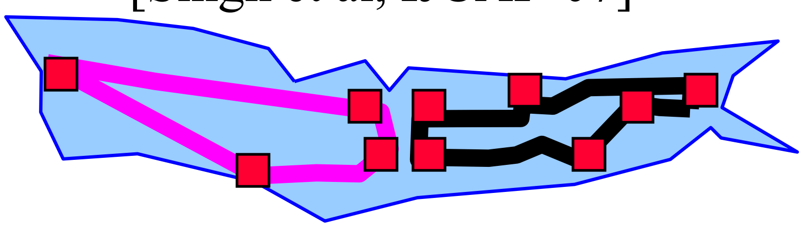
- Instead of each sensor having the same measurement cost, variable cost C(X) for each node
- Aim: max F(A) s.t. C(A) ≤ B

  where $C(A)=\sum_{X\in A} C(X)$
- In this case, construct every possible 3-element subset of V, and run greedy algorithm on each
- Greedy algorithm selects additional nodes X by maximising $\frac{F(A\cup X)-F(A)}{C(X)}$
- Finally choose best set A
- Maintains (1 − 1/e) approximation guarantee

$\max_{\mathbf{A}} F(\mathbf{A})$ or $\max_{\mathbf{A}} \min_i F_i(\mathbf{A})$ subject to

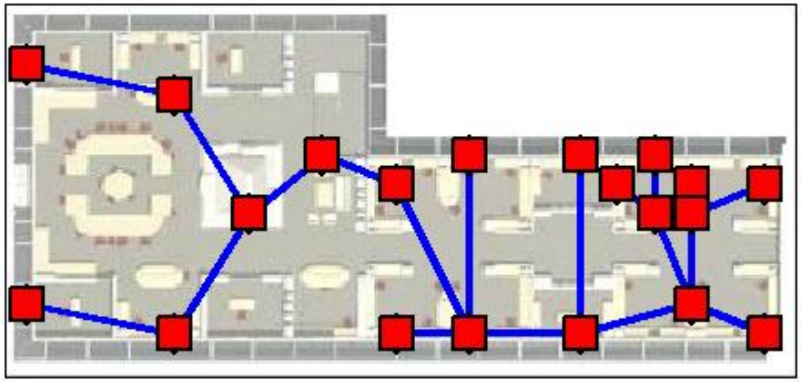
- So far: $|\mathbf{A}| \leq k$
- In practice, more complex constraints:

Locations need to be connected by paths
[Chekuri & Pal, FOCS '05]
[Singh et al, IJCAI '07]



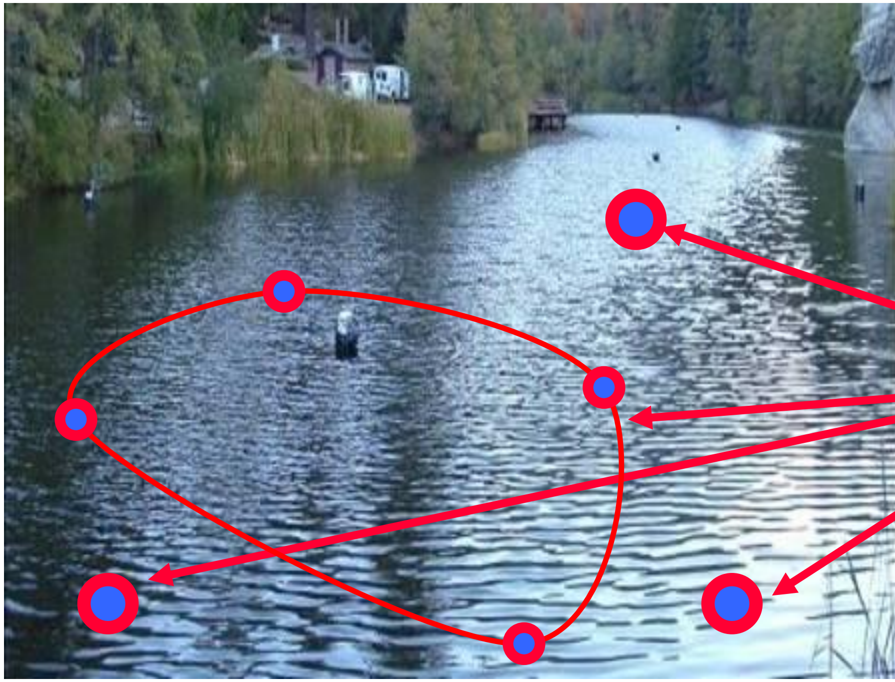
Lake monitoring

Sensors need to communicate (form a routing tree)
[Krause et al., IPSN 06]



Building monitoring

# Informative path planning

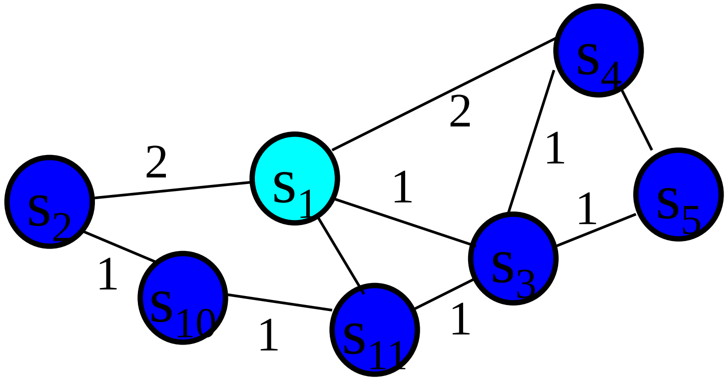


So far:

$\max F(A)$ s.t. $|A| \leq k$

Most informative locations might be far apart!

Robot needs to travel between selected locations




Locations V nodes in a graph

C(A) = cost of cheapest path connecting nodes A

$$\max F(A) \text{ s.t. } C(A) \leq B$$

# The **pSPIEL** Algorithm [K, Guestrin, Gupta, Kleinberg IPSN '06]

- ***pSPIEL***: Efficient **nonmyopic** algorithm

  (**p**added **S**ensor **P**lacements at **I**nformative and cost-**E**ffective **L**ocations)

  - Select **starting** and **ending** location $\mathbf{s_1}$ and $\mathbf{s}_B$
  - **Decompose** sensing region into small, well-separated clusters
  - Solve cardinality constrained problem **per cluster** (greedy)
  - **Combine** solutions using orienteering algorithm
  - Smooth resulting path

# Limitations of pSPEIL

- Requires locality property – far apart observations are independent
- Adaptive algorithm [Singh, Krause, Kaiser IJCAI‘09]
  - Just re-plans on every timestep
  - Often this is near-optimal; if it isn’t, have to add an *adaptivity gap* term to the objective function U to encourage exploration

# Submodular optimisation spectrum

- Maximization: A* = argmax F(A)
  - Sensor placement
  - Informative path planning
  - Active learning
  - …
- Optimise for worst case: $\mathcal{A}^* = \underset{|\mathcal{A}| \le k}{\operatorname{argmax}} \min_i F_i(\mathcal{A})$
- Minimization: A* = argmin F(A)
  - Structure learning ($A^* = \operatorname{argmin} I(X_A; X_{V \setminus A})$)
  - Clustering
  - MAP inference in Markov Random Fields

# Summary

- Submodularity is useful!