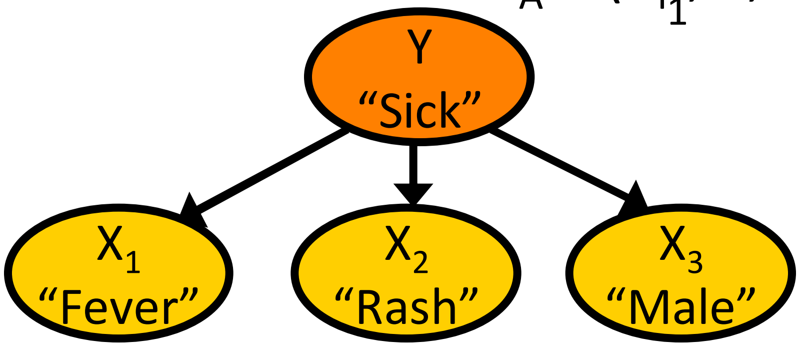
This and next slides taken from http://submodularity.org/submodularity-slides.pdf
Prepared by Andreas Krause & Carlos Guestrin

Feature selection

- Given random variables Y, $X_1, \ldots X_n$
- Want to predict Y from subset $X_A = (X_{i_1},\ldots,X_{i_k})$



Naïve Bayes Model

Want k most informative features:

$$A^* = \text{argmax } IG(X_A; Y) \text{ s.t. } |A| \leq k$$

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

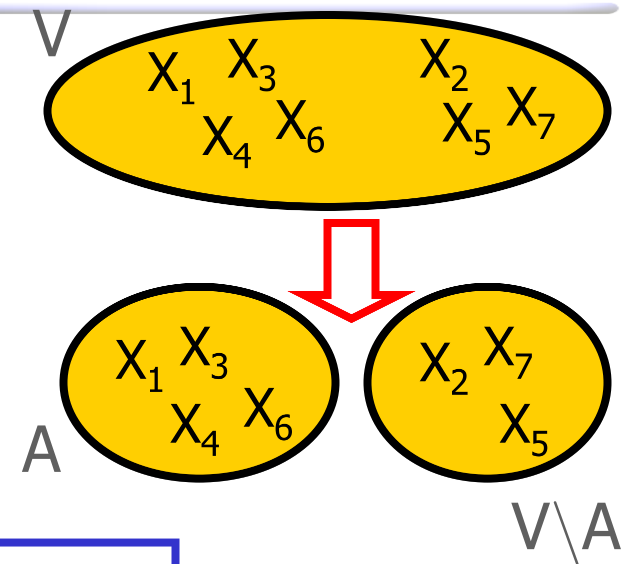
Uncertainty before knowing X_A

Uncertainty after knowing X_A

Problem inherently combinatorial!

Factoring distributions

- Given random variables $X_1,\ldots,X_n$
- Partition variables V into sets A and $V \setminus A$ as independent as possible



Formally: Want

$$A^* = \text{argmin}_A\, I(X_A; X_{V \setminus A}) \quad \text{s.t. } 0<|A|<n$$

where $I(X_A,X_B) = H(X_B) - H(X_B \mid X_A)$

Fundamental building block in structure learning [Narasimhan&Bilmes, UAI '04]

Problem inherently combinatorial!

Combinatorial problems in ML

Given a (finite) set V, function F: $2^V \rightarrow R$, want

$$A^* = \text{argmin } F(A) \quad \text{s.t. some constraints on } A$$

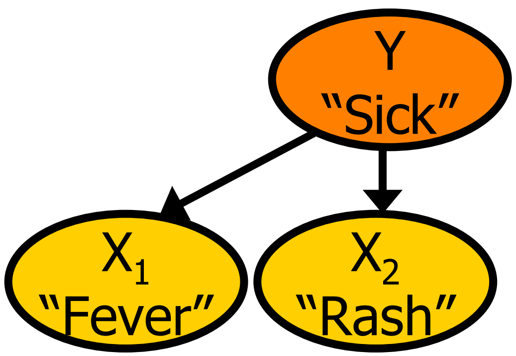
- This talk:
  Fully combinatorial algorithms (spanning tree, matching, …)
  Exploit problem structure to get guarantees about solution!

Set functions

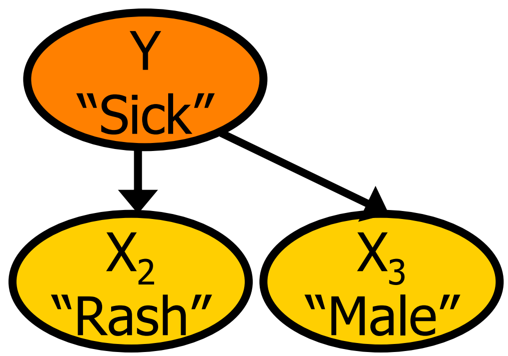
- Finite set $V = \{1,2,\ldots,n\}$
- Function $F: 2^V \rightarrow R$
- Will always assume $F(\emptyset) = 0$ (w.l.o.g.)
- Assume black-box that can evaluate F for any input A

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

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



$F(\{1, 2\}) = 0.9$

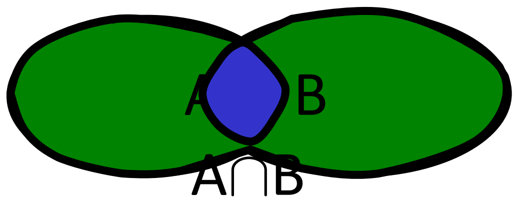


$F(\{2, 3\}) = 0.5$

Submodular set functions

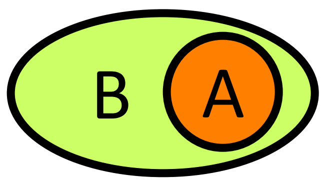
- Set function F on V is called submodular if

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



- Equivalent diminishing returns characterization:

Submodularity:



A + •S ← Large improvement

B + •S ← Small improvement

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

Submodularity and supermodularity

- Set function F on V is called submodular if

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

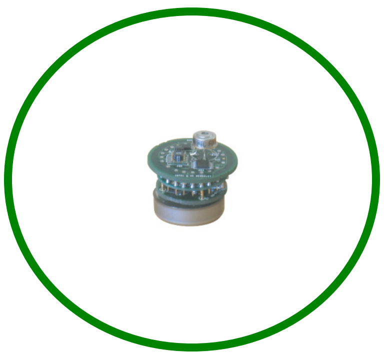
  $\Leftrightarrow$ 2) For all $A\subseteq B$, $s\notin B$, $F(A \cup \{s\}) - F(A) \geq F(B \cup \{s\}) - F(B)$

- F is called supermodular if –F is submodular
- F is called modular if F is both sub- and supermodular

  for modular ("additive") F, $F(A) = \sum_{i\in A} w(i)$

Example: Set cover

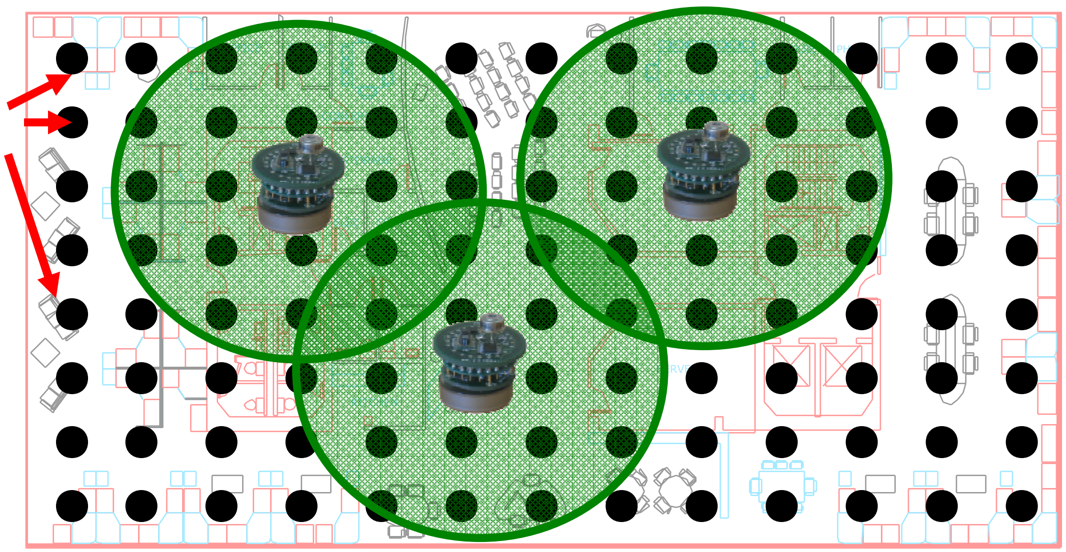
Place sensors in building



Node predicts values of positions with some radius

Want to cover floorplan with discs

Possible locations V



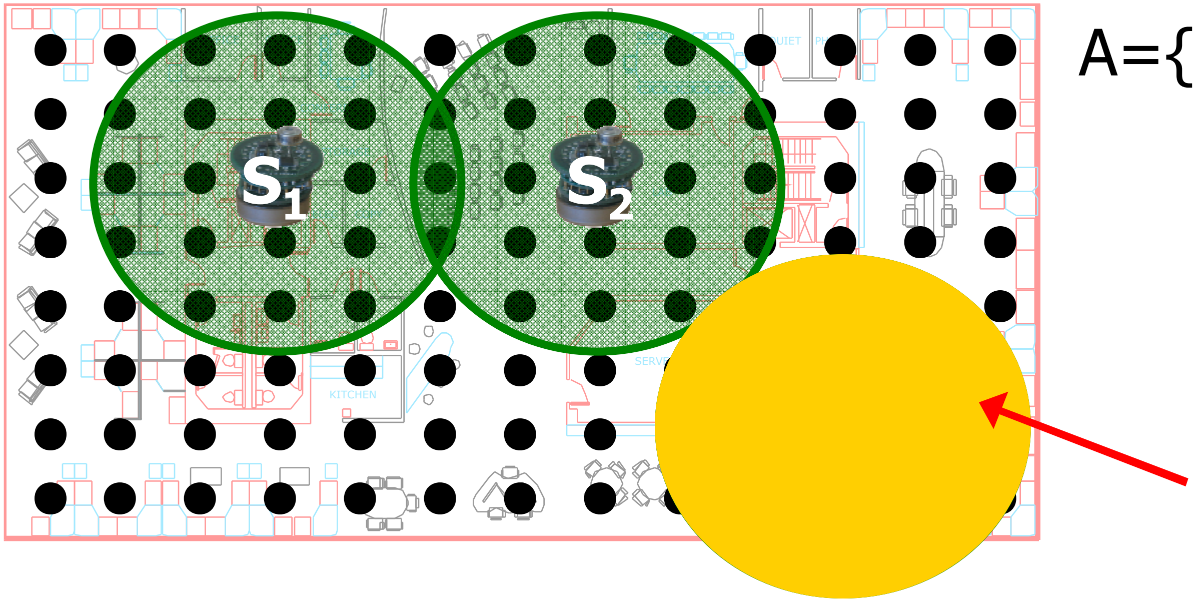
For $A \subseteq V$: F(A) = "area covered by sensors placed at A"

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 cover is submodular

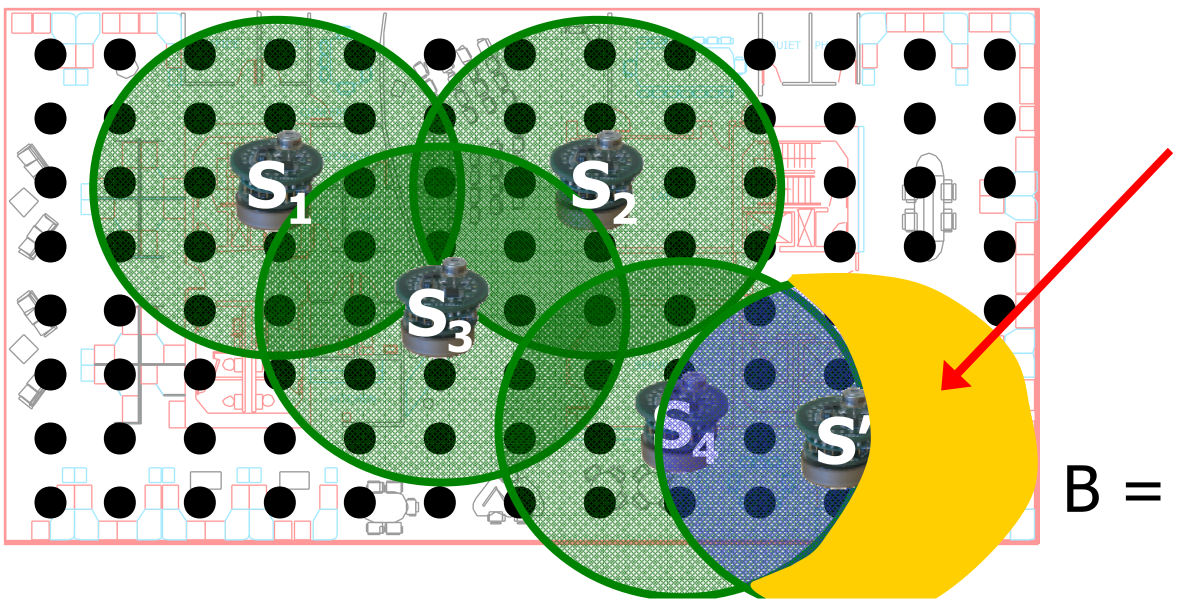


A={ 1, 2 }

F(A∪{S'})-F(A)

≥

F(B∪{S'})-F(B)



B = { 1, 2, 3, 4 }

Example: Mutual information

- Given random variables $X_1,\ldots,X_n$
- $F(A) = I(X_A; X_{V\setminus A}) = H(X_{V\setminus A}) - H(X_{V\setminus A} \mid X_A)$

Lemma: Mutual information F(A) is submodular

$$F(A \cup \{s\}) - F(A) = \underbrace{H(X_s \mid X_A)}_{\text{Nonincreasing in A}} - \underbrace{H(X_s \mid X_{V\setminus(A\cup\{s\})})}_{\text{Nondecreasing in A}}$$

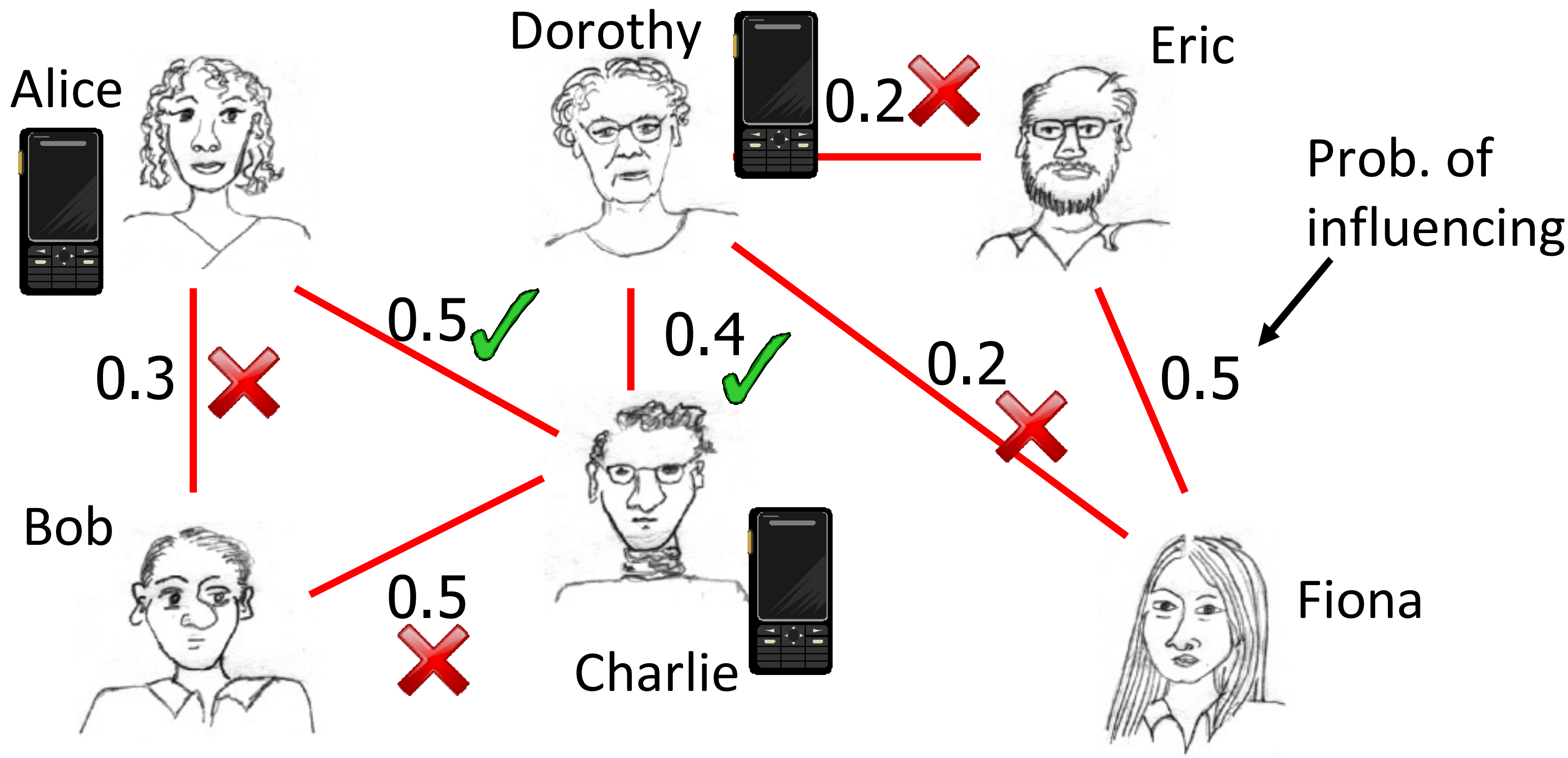
Nonincreasing in A: Nondecreasing in A

$A \subseteq B \Rightarrow H(X_s|X_A) \geq H(X_s|X_B)$

$\delta_s(A) = F(A\cup\{s\})-F(A)$ monotonically nonincreasing

$\Leftrightarrow$ F submodular ☺

Example: Influence in social networks
[Kempe, Kleinberg, Tardos KDD '03]

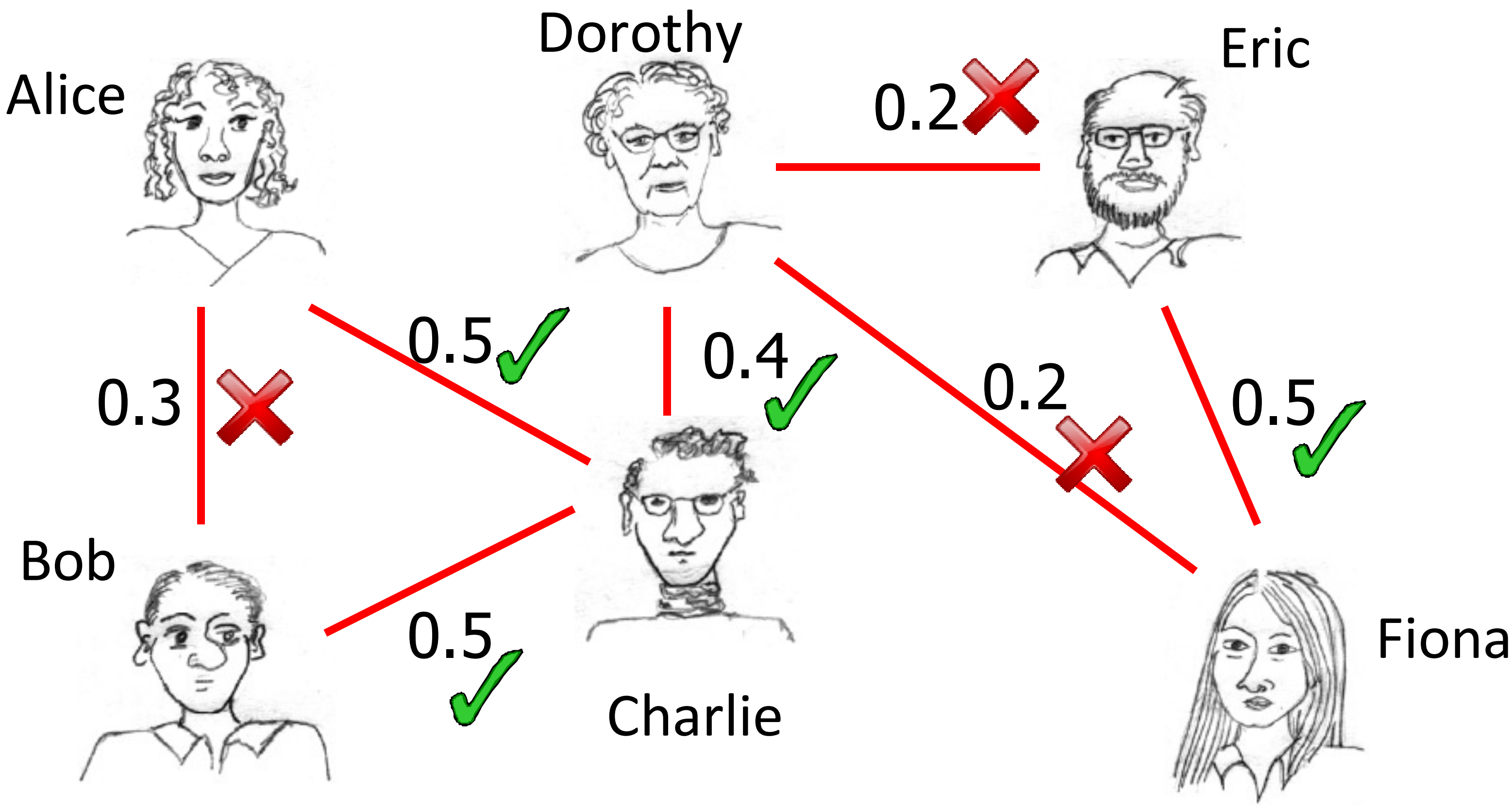


Who should get free cell phones?

V = {Alice,Bob,Charlie,Dorothy,Eric,Fiona}

F(A) = Expected number of people influenced when targeting A

Influence in social networks is submodular [Kempe, Kleinberg, Tardos KDD '03]



Key idea: Flip coins **c** in advance ➔ "live" edges

$F_c(A)$ = People influenced under outcome **c** (set cover!)

$F(A) = \sum_c P(c) F_c(A)$ is submodular as well!

Closedness properties

$F_1,\ldots,F_m$ submodular functions on V and $\lambda_1,\ldots,\lambda_m > 0$

Then: $F(A) = \sum_i \lambda_i F_i(A)$ is submodular!

Submodularity closed under nonnegative linear combinations!

Extremely useful fact!!

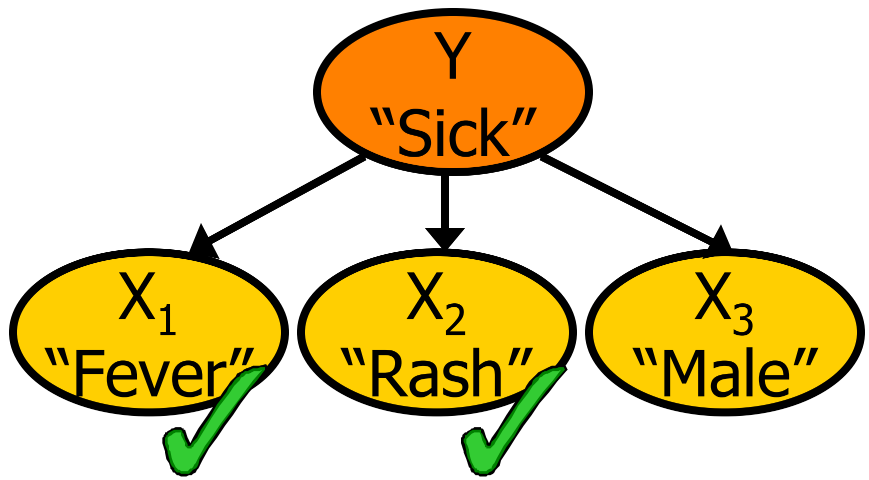
- $F_\theta(A)$ submodular $\Rightarrow \sum_\theta P(\theta) F_\theta(A)$ submodular!
- Multicriterion optimization:
  $F_1,\ldots,F_m$ submodular, $\lambda_i \geq 0 \Rightarrow \sum_i \lambda_i F_i(A)$ submodular

Example: Greedy algorithm for feature selection

- Given: finite set V of features, utility function $F(A) = IG(X_A; Y)$
- Want: $A^* \subseteq V$ such that

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

NP-hard!



<u>Greedy algorithm:</u>

Start with $A = \emptyset$

For i = 1 to k

$s^* := \text{argmax}_s F(A \cup \{s\})$

$A := A \cup \{s^*\}$

How well can this simple heuristic do?

Why is submodularity useful?

Theorem [Nemhauser et al '78]

Greedy maximization algorithm returns A_{greedy}:

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

~63%

- Greedy algorithm gives near-optimal solution!
- More details and exact statement later
- For info-gain: Guarantees best possible unless P = NP! [Krause, Guestrin UAI '05]

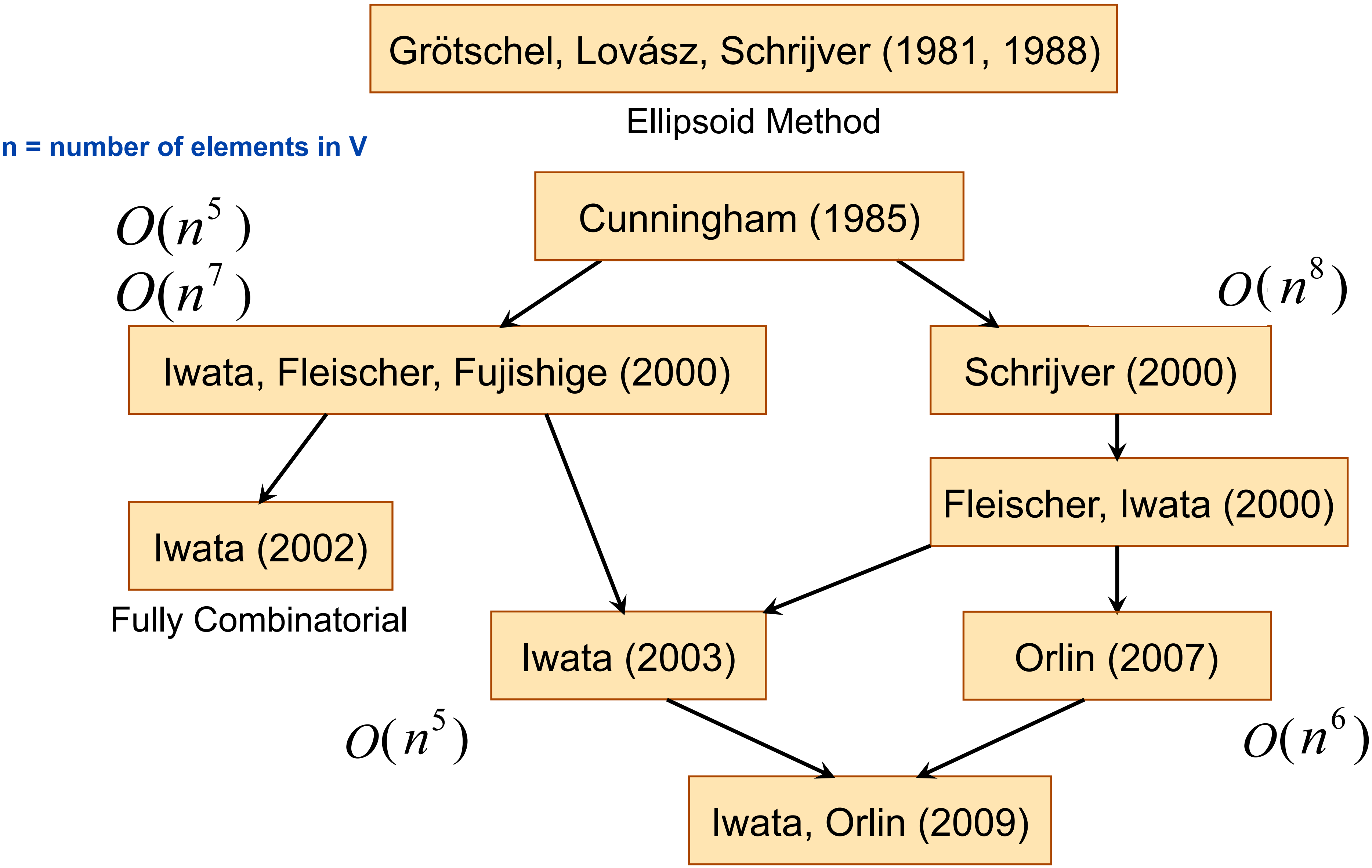
Submodularity in Machine Learning

Several problems in Machine Learning are submodular:

- Minimization: A* = argmin F(A)
  - Structure learning (A* = argmin $I(X_A; X_{V\setminus A})$)
  - Clustering
  - MAP inference in Markov Random Fields
  - …
- Maximization: A* = argmax F(A)
  - Feature selection
  - Active learning
  - Ranking
  - …

This and next slides taken from http://www.science.unitn.it/cirm/Iwata2.pdf, Iwata4.pdf
Prepared by Satoru Iwata

Submodular Function Minimization



Symmetric Submodular Functions

$f : 2^V \rightarrow \mathbf{R}$

Symmetric $f(X) = f(V \setminus X), \quad \forall X \subseteq V.$

Symmetric Submodular Function Minimization

$$\min\{f(X) \mid \phi \neq X \neq V\}?$$

$O(n^3)$ Queyranne (1998)

This and next slides taken from http://theory.stanford.edu/~jvondrak/data/submod-tutorial-2.pdf

CONSTRAINED SUBMODULAR MINIMIZATION

Constraint	Approximation	Hardness	hardness ref
Vertex cover	2	2 [UGC]	Khot,Regev '03
k-unif. hitting set	k	k [UGC]	Khot,Regev '03
k-way partition	$2-2/k$	$2-2/k$	Ene,V.,Wu '12
Facility location	$\log n$	$\log n$	Svitkina,Tardos '07
Set cover	n	$n/\log^2 n$	Iwata,Nagano '09
$\lvert S\rvert \geq k$	$\tilde{O}(\sqrt{n})$	$\tilde{\Omega}(\sqrt{n})$	Svitkina,Fleischer '09
Sparsest Cut	$\tilde{O}(\sqrt{n})$	$\tilde{\Omega}(\sqrt{n})$	Svitkina,Fleischer '09
Load Balancing	$\tilde{O}(\sqrt{n})$	$\tilde{\Omega}(\sqrt{n})$	Svitkina,Fleischer '09
Shortest path	$O(n^{2/3})$	$\Omega(n^{2/3})$	GKTW '09
Spanning tree	$O(n)$	$\Omega(n)$	GKTW '09

Submodular maximization overview

MONOTONE MAXIMIZATION

Constraint	Approximation	Hardness	technique
$\lvert S \rvert \leq k$	$1 - 1/e$	$1 - 1/e$	greedy
matroid	$1 - 1/e$	$1 - 1/e$	multilinear ext.
$O(1)$ knapsacks	$1 - 1/e$	$1 - 1/e$	multilinear ext.
k matroids	$k + \epsilon$	$k/\log k$	local search
k matroids & $O(1)$ knapsacks	$O(k)$	$k/\log k$	multilinear ext.

NON-MONOTONE MAXIMIZATION

Constraint	Approximation	Hardness	technique
Unconstrained	$1/2$	$1/2$	combinatorial
matroid	$1/e$	0.48	multilinear ext.
$O(1)$ knapsacks	$1/e$	0.49	multilinear ext.
k matroids	$k + O(1)$	$k/\log k$	local search
k matroids & $O(1)$ knapsacks	$O(k)$	$k/\log k$	multilinear ext.