ETH
Eidgenössische Technische Hochschule Zürich

Submodularity and ML: Theory and Applications – Part II

Andreas Krause

Data Science Summer School

Outline

1. What is Submodularity?
   Examples, connections
2. Submodular minimization
3. Submodular maximization

TODAY

4. Advanced Topics
   submodularity in deep learning, probabilistic inference, active learning, bandits, ...

Advanced Topics

- Submodularity and probabilistic inference
- Submodularity and deep learning
- Submodularity and interactive learning
- Submodularity and non-convex optimization

Submodularity and probabilistic inference

From optimization to *distributions*

Instead of optimization, we take a probabilistic approach

$$\operatorname*{optimize}_{S} F(S) \Rightarrow P(S) = \frac{1}{Z}\exp\big(\pm F(S)\big)$$

Log-<u>super</u>modular

$$P(S) = \frac{1}{Z}\exp\big(-F(S)\big)$$

Log-<u>sub</u>modular

$$P(S) = \frac{1}{Z}\exp\big(F(S)\big)$$

Equivalent to distrib. over binary vectors $X_i \in \{0,1\} \;\; \forall i \in V$

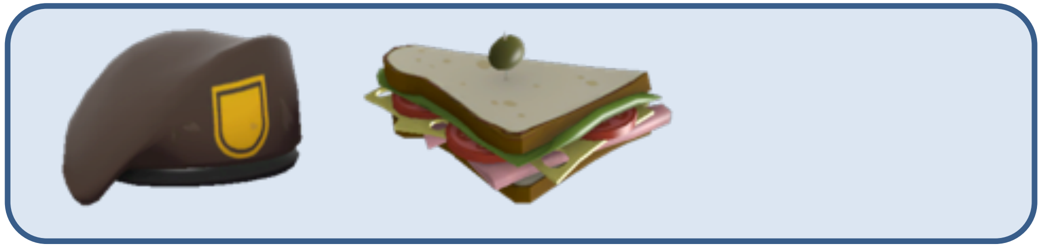
Potential benefits?

Use case: learning from data

Observe sets S_i

S_1

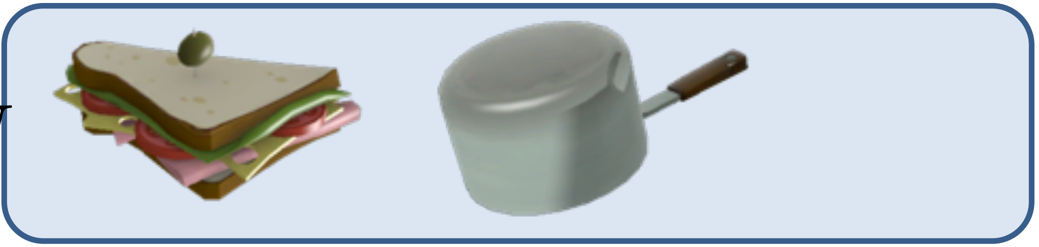
S_2



And so on…

...

S_N



Learn *P(S)*

$$F$$

s.t. S_i likely

under $P(S) \propto \exp(F(S))$

Example: Log-supermodular distributions

Attractive Ising model, Higher-order potentials

[c.f., Boros & Hammer '02, Taskar et al '04, Kohli et al '09]

$$P(S) \propto \exp\Big(-\sum_{i\in S} v_i - \sum_{i\in S, j\notin S} w_{i,j} - \sum_{\ell} \phi_\ell(|S\cap C_\ell|)\Big)$$

unary (modular)

pairwise (cut function)

higher-order (concave over cardinality)

Log-supermodular → Marginals?

Example: Log-submodular distributions

Determinantal point processes [Macchi '75; Kulesza & Taskar '12]

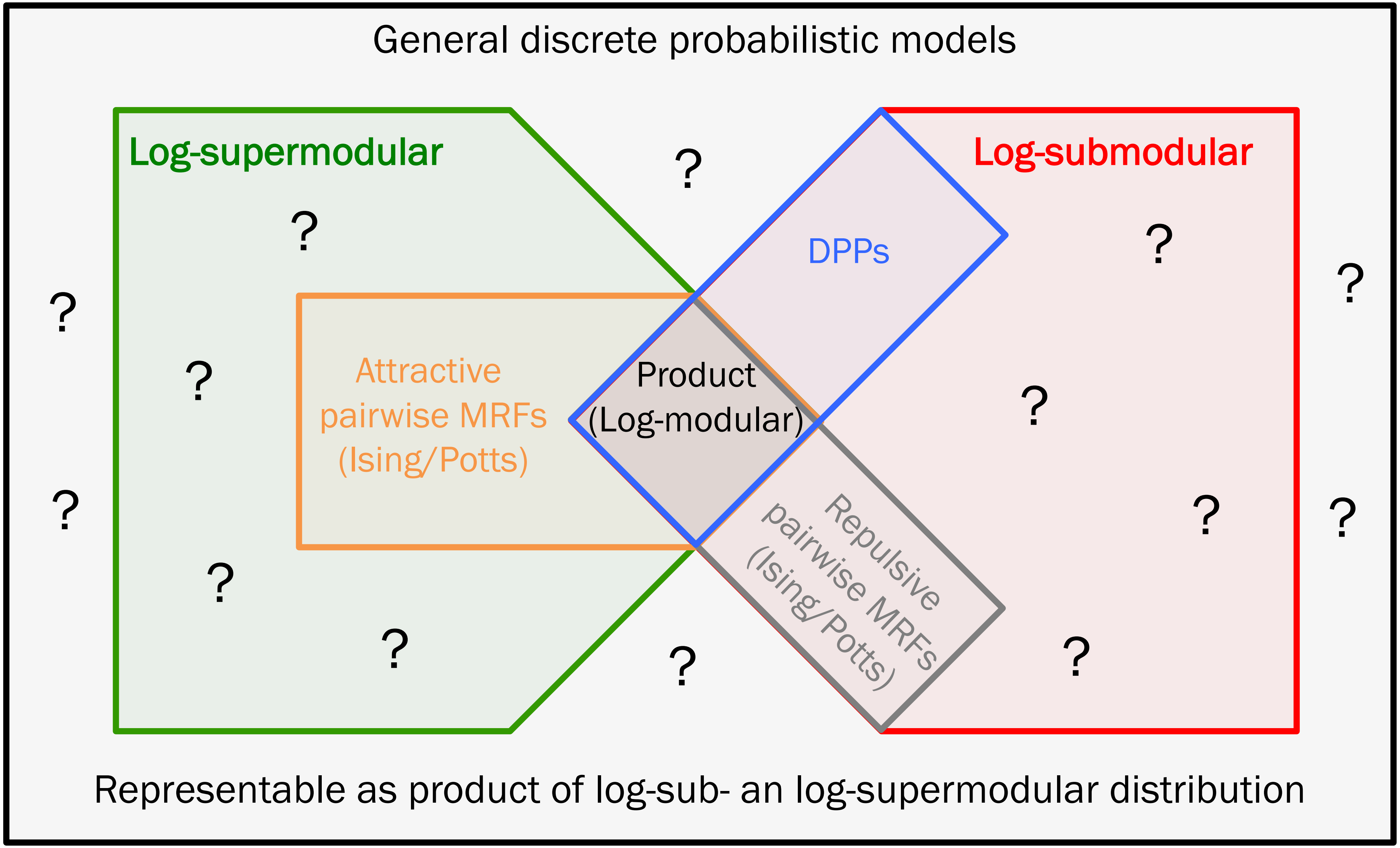
pos. definite kernel

$$P(S) \propto \exp \underbrace{\log |\mathbf{K}_{S,S}|}_{\text{Submodular}}$$

$$\mathbf{K}_{S,S} = \begin{pmatrix} k(i_1, i_1) & \dots & (i_1, i_{|S|}) \\ \vdots & & \vdots \\ k(i_{|S|}, i_1) & \dots & (i_{|S|}, i_{|S|}) \end{pmatrix}$$

(Macchi 75, Feder-Mihail 82, Borodin 02, Deshpande-Rademacher-Vempala-Wang 06, Borcea-Bränden 09, Borcea-Bränden-Liggett 09, Kulesza-Taskar 12, Anari-Oveis Gharan-Rezaei 16, Li-Jegelka-Sra 16, …)

Relation to other discrete prob. models



Key challenge:

Compute normalizing constant (partition function)

$$P(S) = \frac{1}{\mathcal{Z}} \exp(\pm F(S))$$

$$\mathcal{Z} = \sum_S \exp(\pm F(S))$$

#P-hard for both log-sub/supermodular distributions

Hard to approximate in both cases as well

[Goldberg & Jerrum '07, Sly & Sun'12]

$$P(e \in S) = \sum_{S: e \in S} P(S)$$

$$= \frac{\sum_{S: e \in S} \exp F(S)}{\sum_S \exp F(S)}$$

$$= \frac{\sum_{S' \subseteq V \setminus \{e\}} \exp F'(S')}{\sum_{S \subseteq V} \exp F(S)} = \frac{Z'}{Z}$$

$$F'(S) = F(S \cup \{e\}) \qquad F' : 2^{V \setminus \{e\}} \to \mathbb{R}$$

Existing approximate approaches

For low-order models ($|C_i|$ small, typically = 2),

$$P(S) \propto \exp(-\sum_i F_i(S \cap C_i)) \qquad \equiv P(\mathbf{X}) \propto \prod_i \Phi_i(\mathbf{X}_{C_i})$$

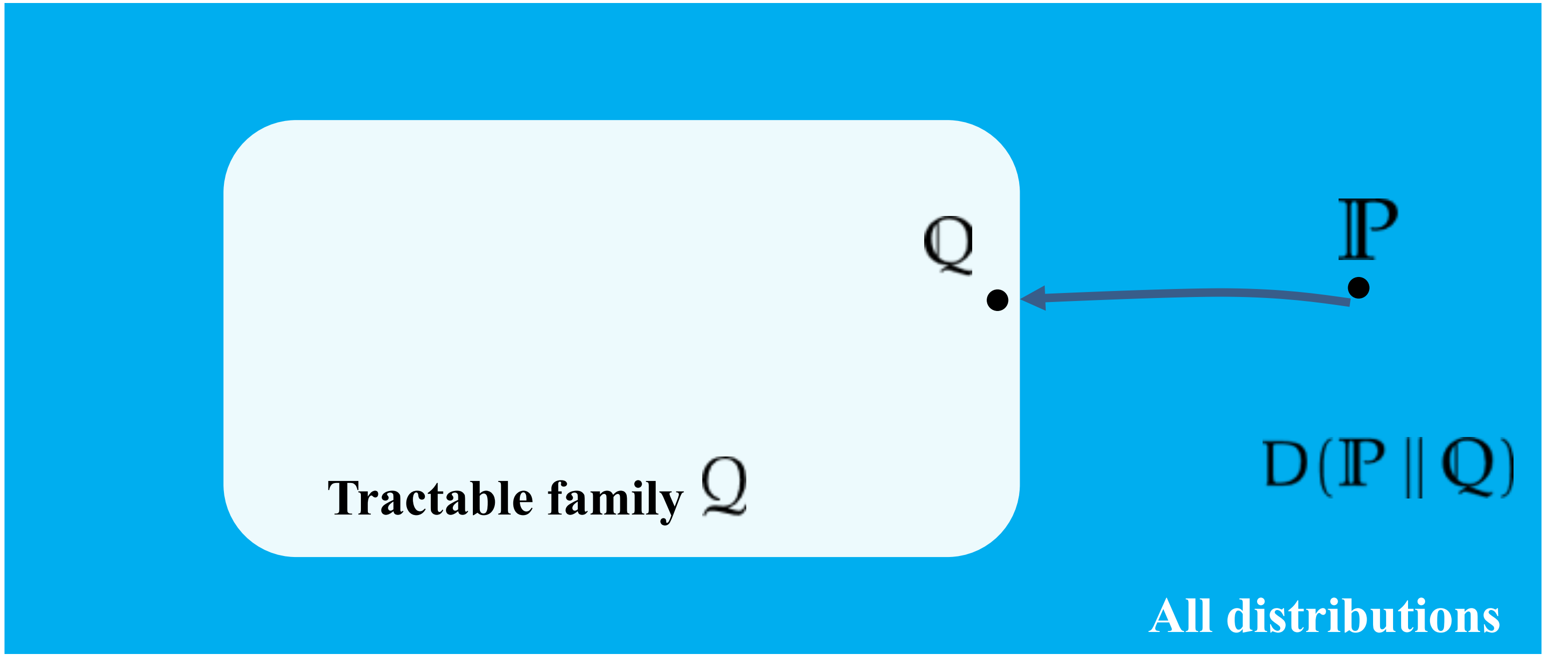
$$\mathbf{X} \in \{0,1\}^{|V|}$$

many heuristics for approximating Z:

- Mean-field and variants
- Belief propagation / sum-product and variants

Running time *exponential in model order* (max$_i$ $|C_i|$)

Variational Inference



[Djolonga]

Approximate Inference in General PSMs

[Djolonga & K. '14,'15,'16]

Variational approach to inference in log-sub/supermodular distributions and beyond

- Tractable optimization independent of model order
- Provides upper and lower bounds on Z
- Some guarantees on accuracy of $\log Z$
- For log-supermodular distributions, shares mode (i.e., preserves MAP configuration)

Our workhorse: *modular* functions

- Additive submodular functions:

One number (weight) per element in V

$$m(S) = \sum_{i \in S} m_i$$

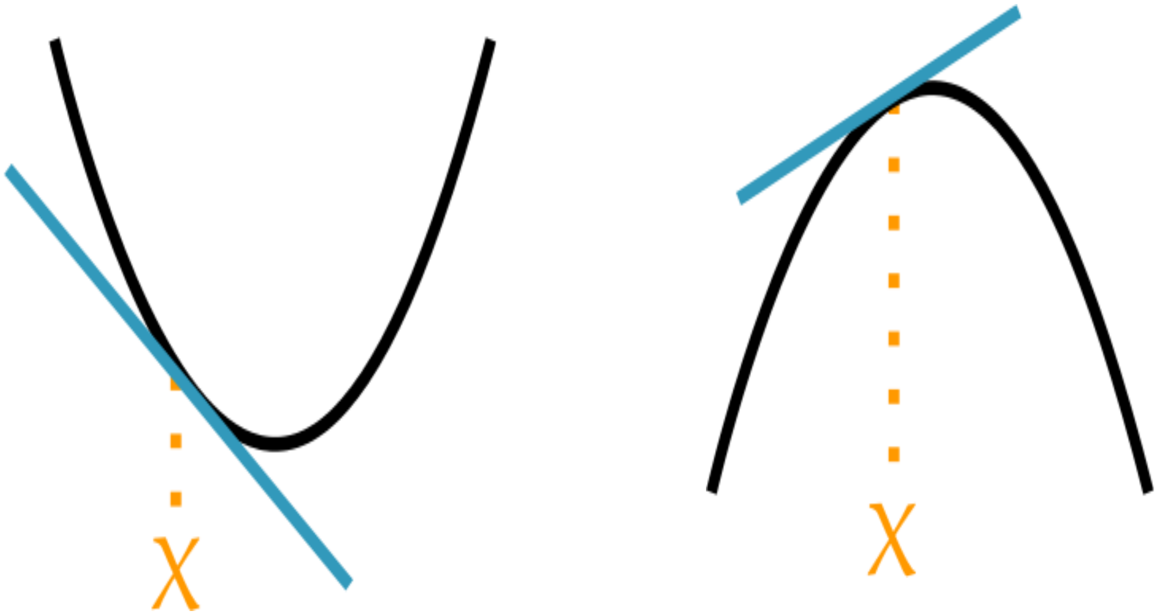
- Correspond to completely factorized distributions, with marginals

$$P(i \in S) = \Big(1 + \exp(-m_i)\Big)^{-1} = \sigma(m_i)$$

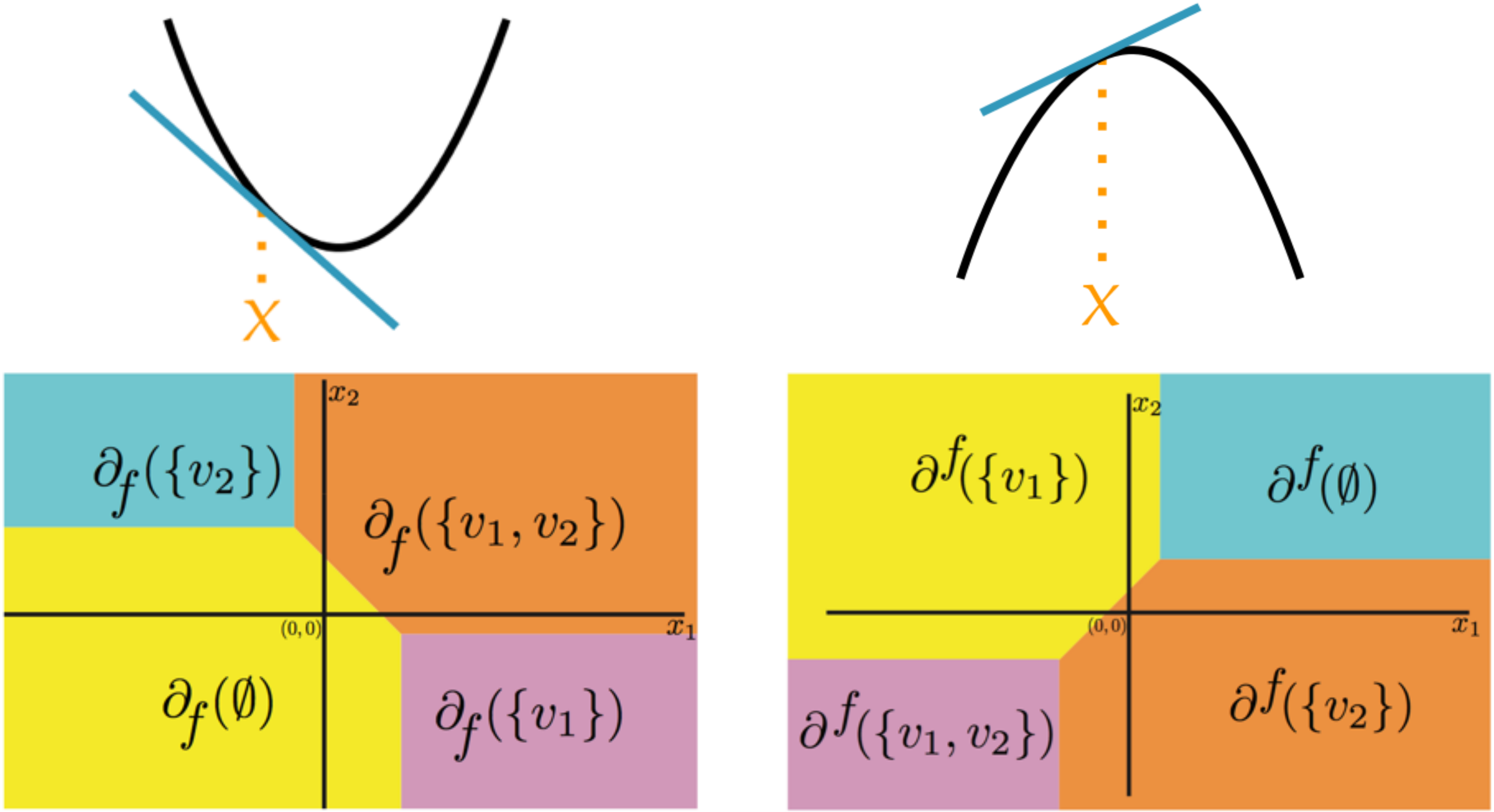
and analytic partition function $\sum_i \log\Big(1 + \exp(m_i)\Big)$

Sub- and superdifferentials

- Similar to convex functions, submodular functions have **sub**differentials [c.f. Fujishige '91]
- But they also have **super**differentials [c.f. Iyer, Jegelka, Bilmes'13]



Semigradient polyhedral structure



Use in optimization: [Jegelka & Bilmes `11, Iyer et al. ICML `13]

Courtesy: Jeff Bilmes

Key idea

Elements from the sub/superdifferentials bound *F*

$$x(A) \leqslant F(A) \leqslant y(A)$$

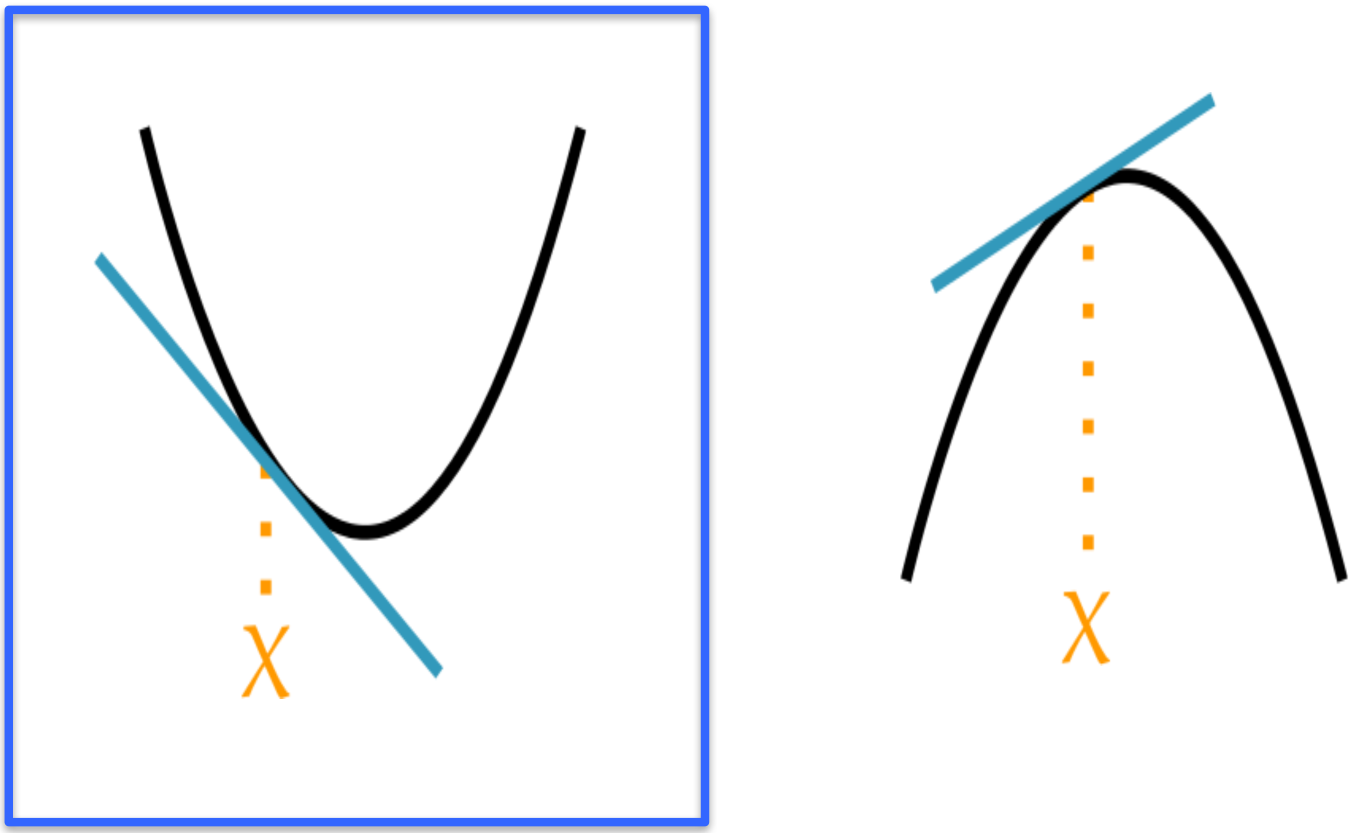
and hence yield bounds on the partition function

$$\textstyle\sum_{A \subseteq V} \exp(+x(A)) \leqslant \sum_{A \subseteq V} \exp(+F(A)) \leqslant \sum_{A \subseteq V} \exp(+y(A))$$

$$\textstyle\sum_{A \subseteq V} \exp(-x(A)) \geqslant \sum_{A \subseteq V} \exp(-F(A)) \geqslant \sum_{A \subseteq V} \exp(-y(A))$$

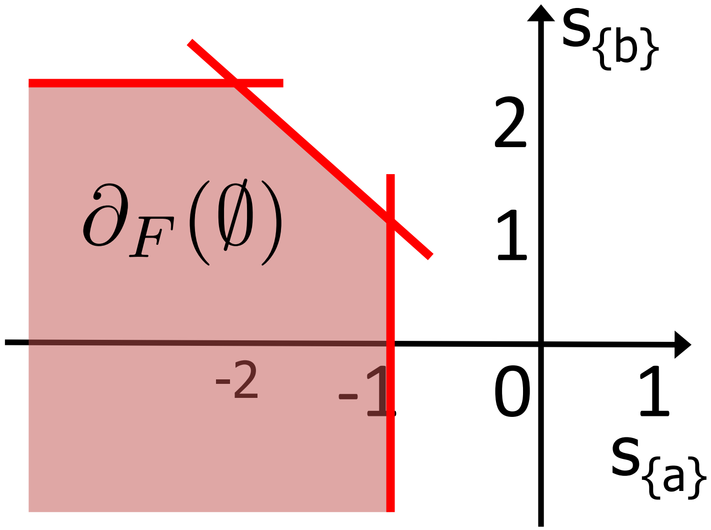
We optimize over these upper and lower bounds

Sub- and superdifferentials



Subgradients of submodular functions

$$\partial_F(X) = \{\mathbf{s} \in \mathbb{R}^n \mid \forall Y \subseteq V : F(Y) \geq F(X) + \mathbf{s}(Y) - \mathbf{s}(X)\}.$$



S	F(S)
{}	0
{a}	-1
{b}	2
{a,b}	0

- Exponential-size description ☹
- Efficient *O(n log n)* linear optimization ☺ [Edmonds/Fujishige]

Optimizing over subgradients

For any *X*, and any $\mathbf{s} \in \partial_F(X)$, get a bound on *Z:*

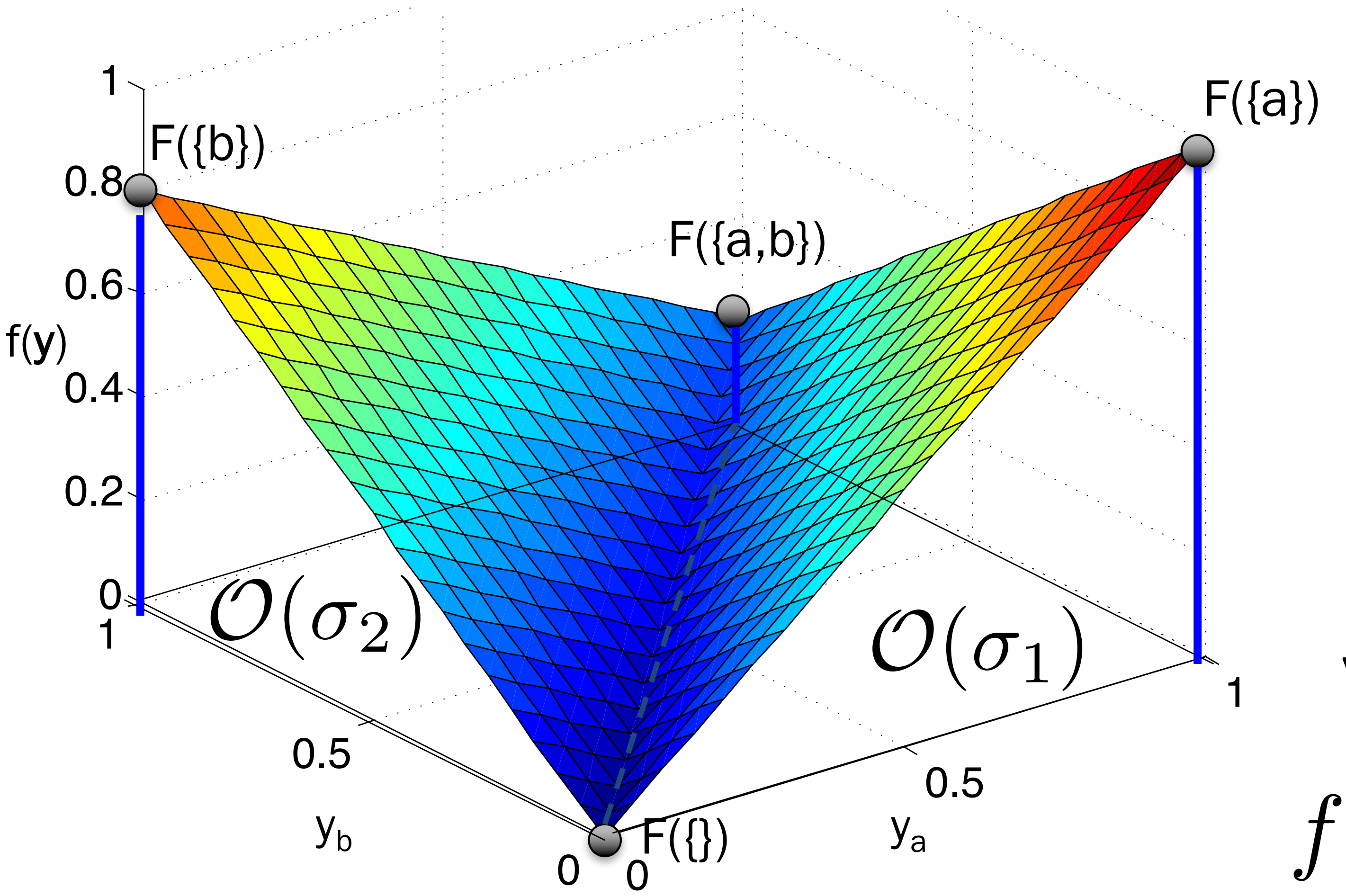
$$\sum_{A \subseteq V} \exp(-F(A)) \leq \underbrace{\sum_{A \subseteq V} \exp\Big(-\mathbf{s}(A) + \mathbf{s}(X) - F(X)\Big)}_{\mathcal{Z}_X^-(\mathbf{s})}$$

Efficiently computable

To get best bound, need to optimize over *X* and $\mathbf{s} \in \partial_F(X)$

Looks like a difficult mixed discrete-continuous problem ☹

Recall: Lovász extension



A	F(A)
{}	0
{a}	1
{b}	.8
{a,b}	.2

$$\forall \mathbf{y} \in \mathcal{O}(\sigma)$$
$$f(\mathbf{y}) = \mathbf{f}_\sigma^T \mathbf{y}$$

$$\mathcal{O}(\sigma) = \{\mathbf{y} : y_{\sigma(n)} \leq y_{\sigma(n-1)} \leq \cdots \leq y_{\sigma(1)}\}$$
$$[\mathbf{f}_\sigma]_{\sigma(i)} = F(\sigma(i) \mid \{\sigma(1), \ldots, \sigma(i-1)\})$$

Theory: Variational inference in log-supermodular distributions

Theorem [Djolonga, K '15]: The following are equivalent:

$$\underbrace{\min_{X, \mathbf{s} \in \partial_F(X)} \mathcal{Z}_X^-(\mathbf{s})}$$

Minimize upper bound on partition function

$$\mathbf{y}^* = \underbrace{\operatorname*{argmin}_{\mathbf{y} \in \mathbb{R}^p} f(\mathbf{y}) + \frac{1}{2} ||\mathbf{y}||_2^2} \quad \text{for} \quad Q^*(i) = \frac{1}{1 + \exp(-y_i)}$$

Regularized Lovász extension f (aka min-norm-point)

$$Q^* = \underbrace{\operatorname*{minimize}_{Q \text{ fact. dist.}} D_\infty(P \,||\, Q)} \quad \text{for } D_\infty(P||Q) = \max_{S \subseteq V} \log \frac{P(S)}{Q(S)}$$

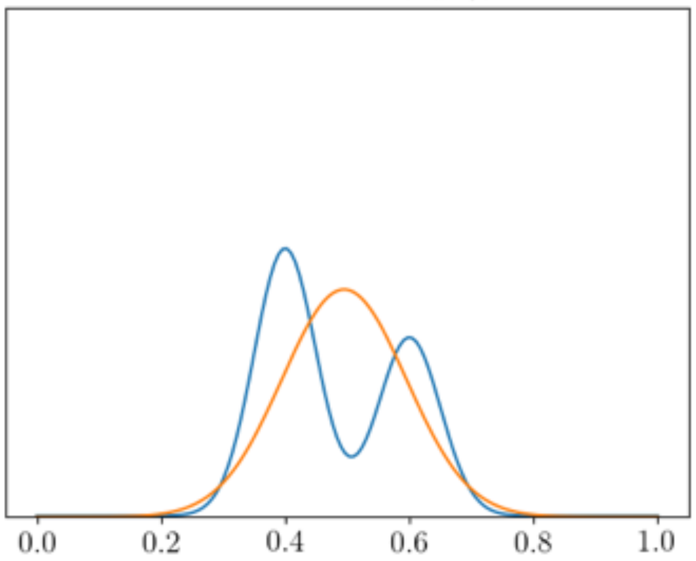
Minimize Renyi divergence

Illustration: Renyi divergence

Inclusive

$$D_\infty(\mathbb{P} \,\|\, \mathbb{Q}) = \log \max_{\mathbf{x}\in\mathcal{X}} \mathbb{P}(\mathbf{x})/\mathbb{Q}(\mathbf{x})$$

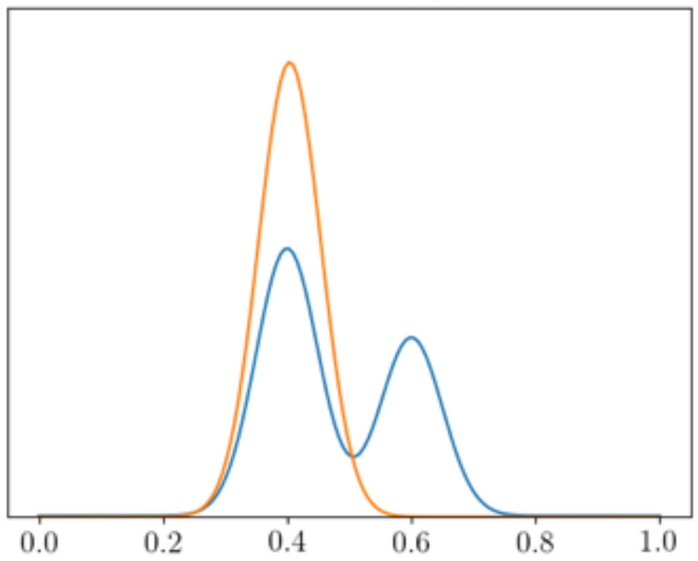
Minimizer of $D_\infty(\mathbb{P} \,\|\, \mathbb{Q}_\mu)$



Exclusive

$$D_\infty(\mathbb{Q} \,\|\, \mathbb{P}) = \log \max_{\mathbf{x}\in\mathcal{X}} \mathbb{Q}(\mathbf{x})/\mathbb{P}(\mathbf{x})$$

Minimizer of $D_\infty(\mathbb{Q}_\mu \,\|\, \mathbb{P})$

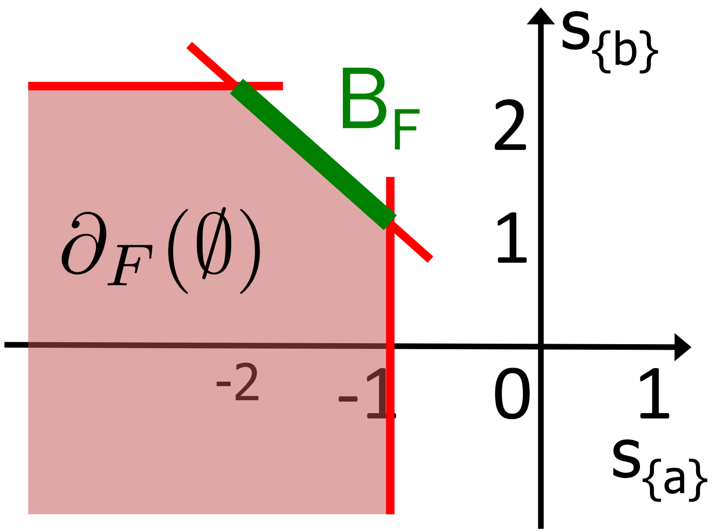


[Djolonga]

Proof sketch (i) ⇔ (ii)

Can show: Min. of $\min_{X, \mathbf{s} \in \partial_F(X)} \mathcal{Z}_X^-(\mathbf{s})$ attained at $X = \emptyset$,

and **s** restricted to base polytope B_F



For the resulting problem:

$$\operatorname*{argmin}_{\mathbf{s} \in B_F} \mathcal{Z}_\emptyset^-(\mathbf{s}) \equiv \operatorname*{argmin}_{\mathbf{s} \in B_F} \sum_{i \in V} \log(1 + \exp(s_i))$$

(sep. convex opt. over base polytope) [c.f. Nagano '07]

$$\equiv \operatorname*{argmin}_{\mathbf{s} \in B_F} \sum_{i \in V} s_i^2$$

(Fenchel duality) [c.f. Bach '11]

$$\equiv \operatorname*{argmin}_{\mathbf{s} \in \mathbb{R}^n} f(\mathbf{x}) + ||\mathbf{s}||_2^2$$

ETH

Connection to min-norm point (MNP) problem

Optimizing variational bound $\equiv$ Min-norm-point problem!

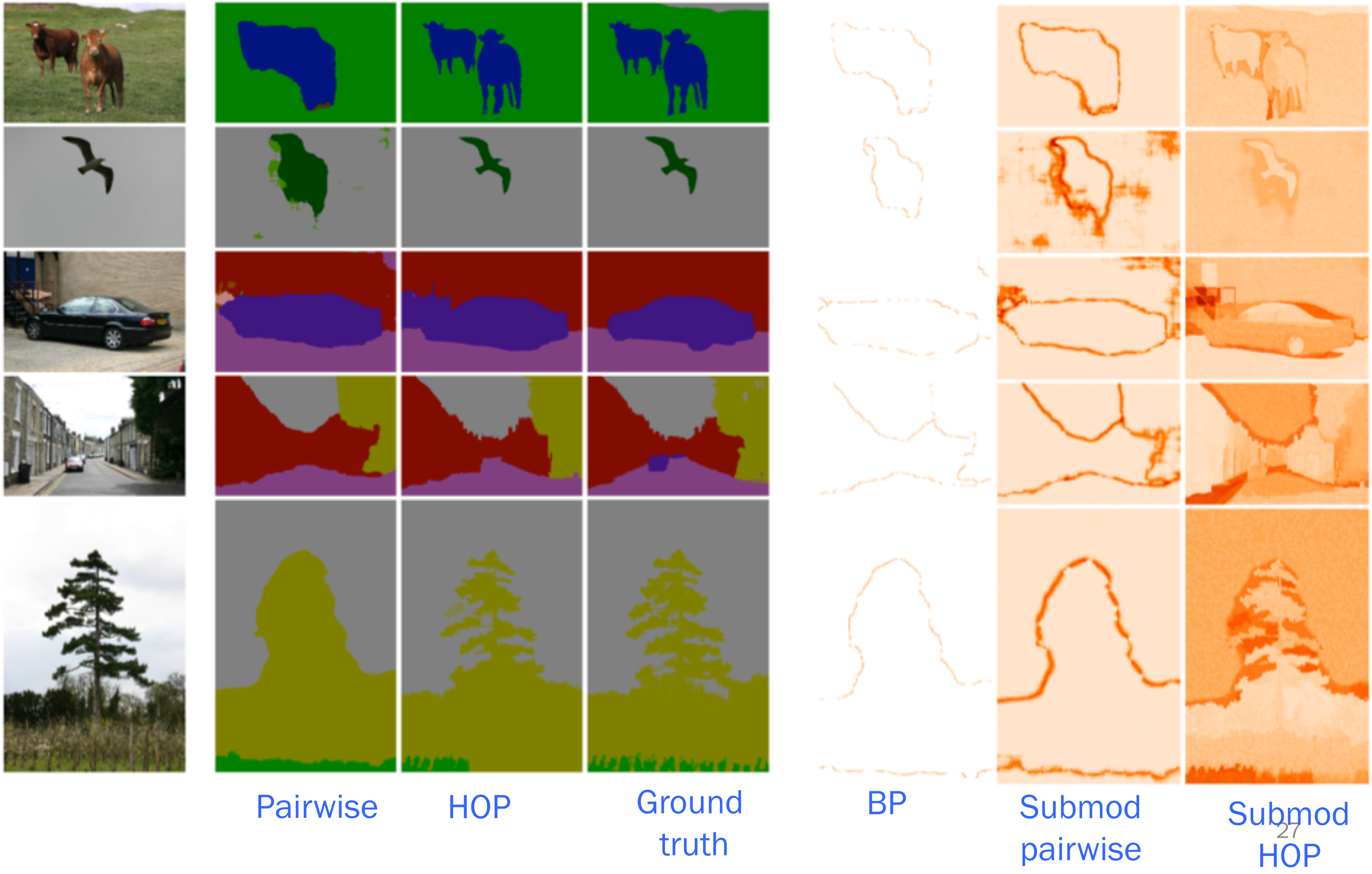
Algorithmic implications:

- Solvable in strongly polynomial time via poly. many SFMin, or pseudo-polynomial time via Fujishige-Wolfe's algorithm [Chakrabarty et al '14]
- In practice fast algorithms based on convex optimization, exploiting special structure [e.g., Jegelka et al '13, Nishihara et al '14]

Corollary: Thesholding the solution at ½ gives a MAP configuration (i.e., approximation shares mode)

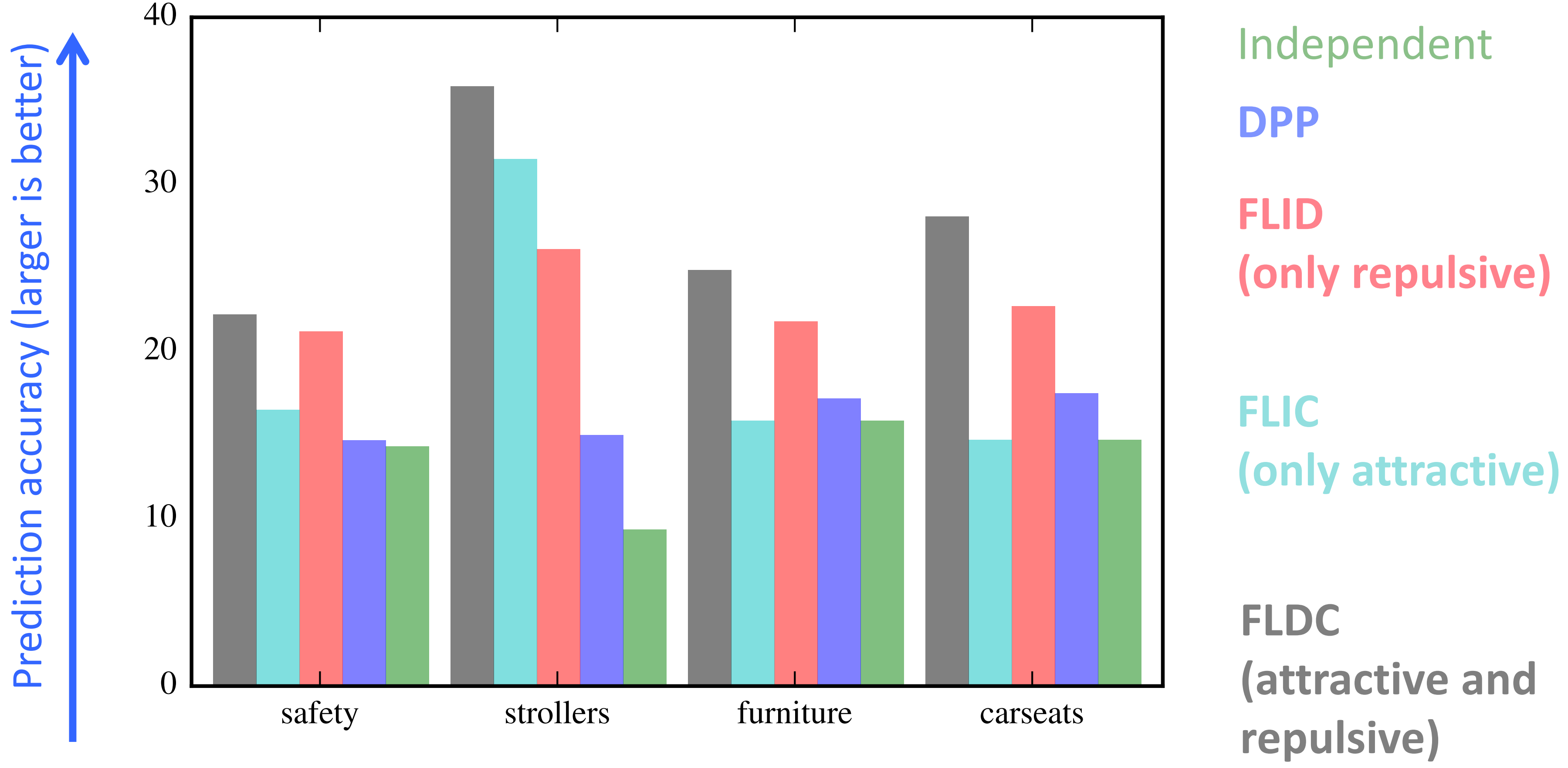
Application: Image Segmentation

[Zhang, Djolonga, Krause, ICCV'15; MSRC-21 data]



Model comparison: Product recommendation task

[data from Gillenwater et al.'14]

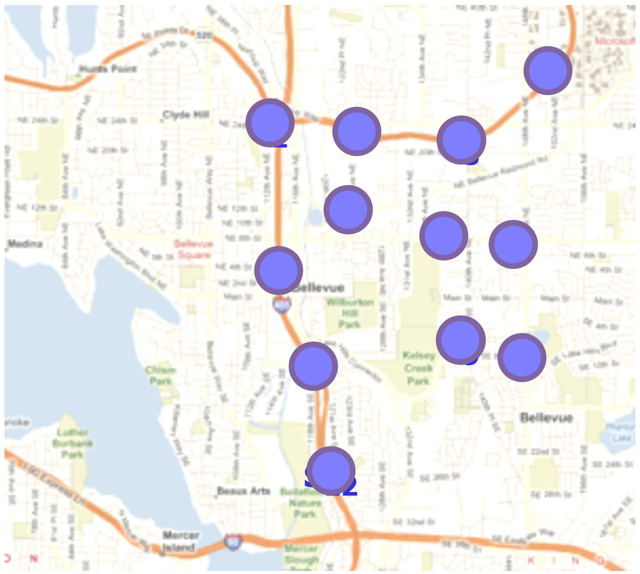


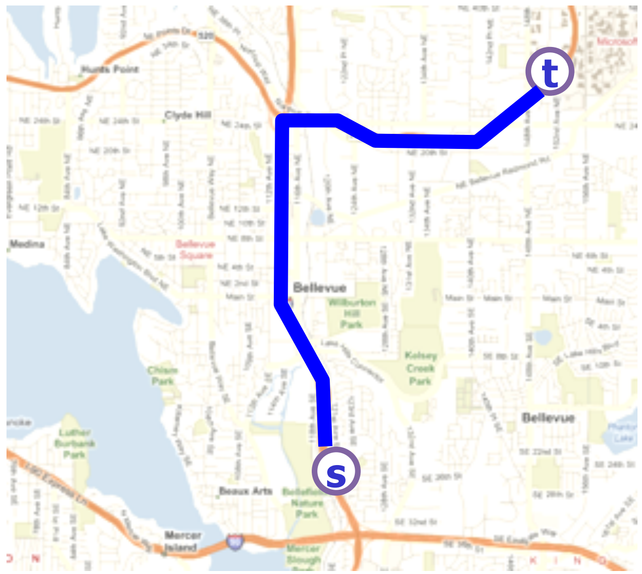
Submodularity and Deep Learning

Data-driven decision making

Data → Model → Decision

$$x_1, \ldots, x_n \qquad \hat{\theta} = \operatorname*{argmax}_{\theta} P(x_{1:n} \mid \theta) \qquad \hat{S} = \operatorname*{argopt}_{S \in \mathcal{I}} F(S \mid \hat{\theta})$$





Data-driven decision making

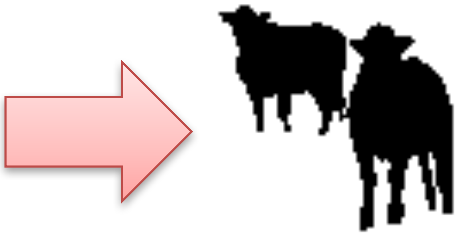
Data → Model → Decision

$$x_1, \ldots, x_n \qquad \hat{\theta} = \operatorname*{argmax}_{\theta} P(x_{1:n} \mid \theta) \qquad \hat{S} = \operatorname*{argopt}_{S \in \mathcal{I}} F(S \mid \hat{\theta})$$

How can we reason about making complex (combinatorial) decisions from data?

Motivation: Structured prediction

[Lafferty et al ‘01, Collins ‘02, Taskar ‘04, Tsochantaridis et al ’05, …]



$\mathbf{x}_1$ S_1

$\vdots$ $\vdots$

$\mathbf{x}_n$ S_n

$$S_i = \underset{S}{\operatorname{argmin}}\, F(S \mid \mathbf{x}_i, \theta)$$

E.g., minimum cut

$$F(S) = \sum_{i \in S} v_i + \sum_{i \in S, j \notin S} w_{i,j}$$

ETH

Motivation: Attention / Interpretability

[Mnih et al'14, Martins & Astudillo'16, Niculae & Blondel '17, ...]

Task: Given text T and hypothesis H predict whether T entails H:

- T = "A band is playing on a stage at a concert and the attendants are dancing to the music"
- H = "No one is dancing"

Want "Interpretability": Besides predicting the answer, tell me which sparse subset S of input is most relevant:

- Rationale: "attendants are dancing"

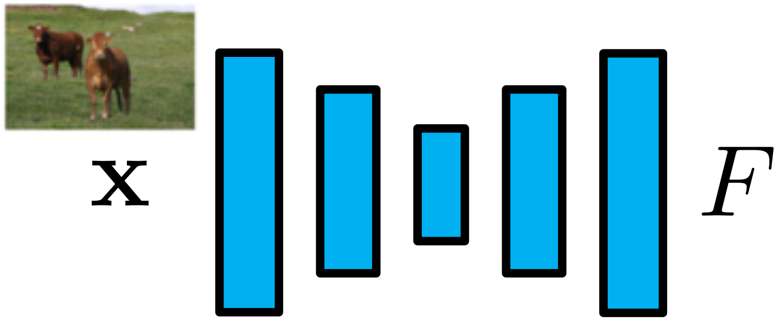
Attention ≅ input dependent sparsity

[Mnih et al'14, Martins & Astudillo'16, Niculae & Blondel '17, ...]

Sparse estimation $\quad \mathbf{x} \mapsto g(\mathbf{x}_S; \theta)$

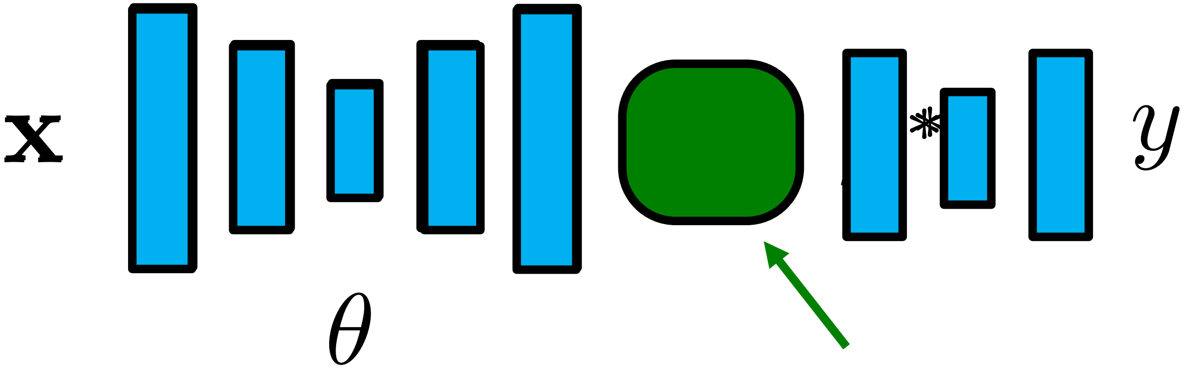
"Attention" $\quad \mathbf{x} \mapsto g(\mathbf{x}_{S(\mathbf{x};\theta_1)}; \theta_2)$

Differentiable Discrete Optimization



$$\boxed{S^* = \operatorname*{argmin}_S F(S)}$$

e.g., min-cut, MST, etc.



Differentiable! ☺

➔ Train model end-to-end (via backpropagation and SGD)

Smoothing via probabilistic modeling

$$\operatorname*{minimize}_S F(S \mid \theta) \quad \Rightarrow \quad P(S \mid \theta) = \frac{1}{\mathcal{Z}} \exp\big(-F(S \mid \theta)\big)$$

E.g., submodular minimization

Log-supermodular distribution

- Log-likelihood of S provides differentiable objective! ☺
- Key challenge: Normalizer $\mathcal{Z}$ is typically intractable! ☹

- Can we leverage structure of the discrete problem to obtain efficiently computable differentiable objectives?

Differentiable learning of Submodular Functions

[with Djolonga, NIPS 2017]

Given data $D = \{(\mathbf{x}_1, S_1), \ldots, (\mathbf{x}_n, S_n)\}$ and parametrized family of functions, $F(S \mid \mathbf{x}, \theta)$ want

$$\theta^* = \arg\max_\theta \sum \log Q_i^*(S_i)$$

$$\text{s.t. } Q_i^* = \arg\min_Q D_\infty\Big(P(\cdot \mid \mathbf{x}_i, \theta) || Q\Big)$$

Want to learn parameters to maximize a posteriori probability under variational approximation Q

We show how to compute gradients of this objective

Variational inference in PSMs

Theorem [Djolonga, K '15]: The solution of

$$Q^* = \underset{Q \text{ fact. dist.}}{\text{minimize}}\ D_\infty(P \,\|\, Q)$$ for

is given by $Q^*(i) = \dfrac{1}{1+\exp(-y_i)}$ where

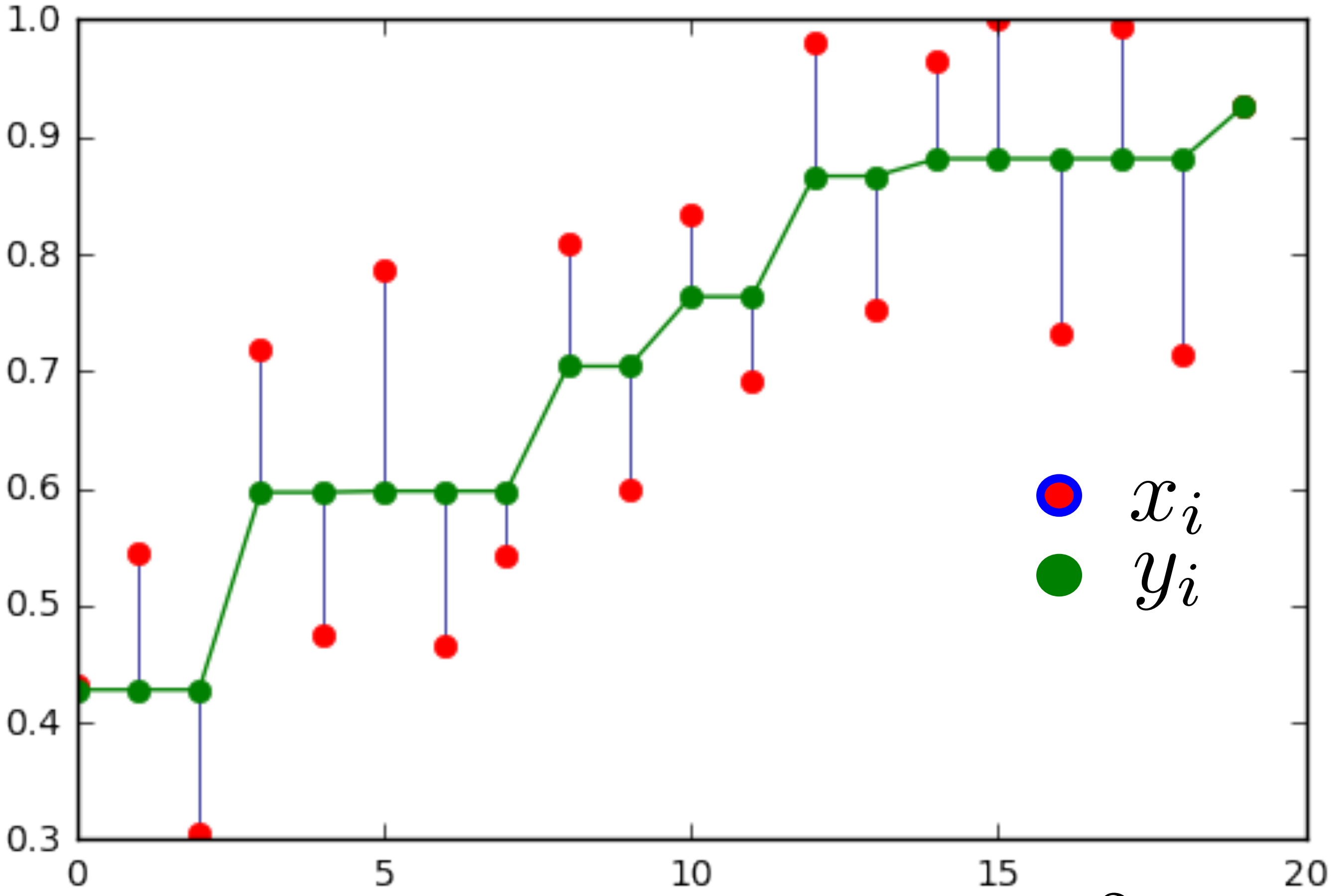
$$\mathbf{y}^* = \underset{\mathbf{y}\in\mathbb{R}^n}{\operatorname{argmin}}\, f(\mathbf{y}) + \frac{1}{2}||\mathbf{y}||_2^2 = \underset{\sigma,\mathbf{y}\in\mathcal{O}(\sigma)}{\operatorname{argmin}}\ \mathbf{f}_\sigma^T\mathbf{y} + \frac{1}{2}||\mathbf{y}||_2^2$$

$$= \underset{\sigma,\mathbf{y}\in\mathcal{O}(\sigma)}{\operatorname{argmin}}\ \underbrace{||\mathbf{f}_\sigma^T + \mathbf{y}||_2^2}_{\text{Isotonic regression}}$$

[cf Bach '11]

Key challenge: Argmin differentiation of isotonic regression!

Isotonic Regression

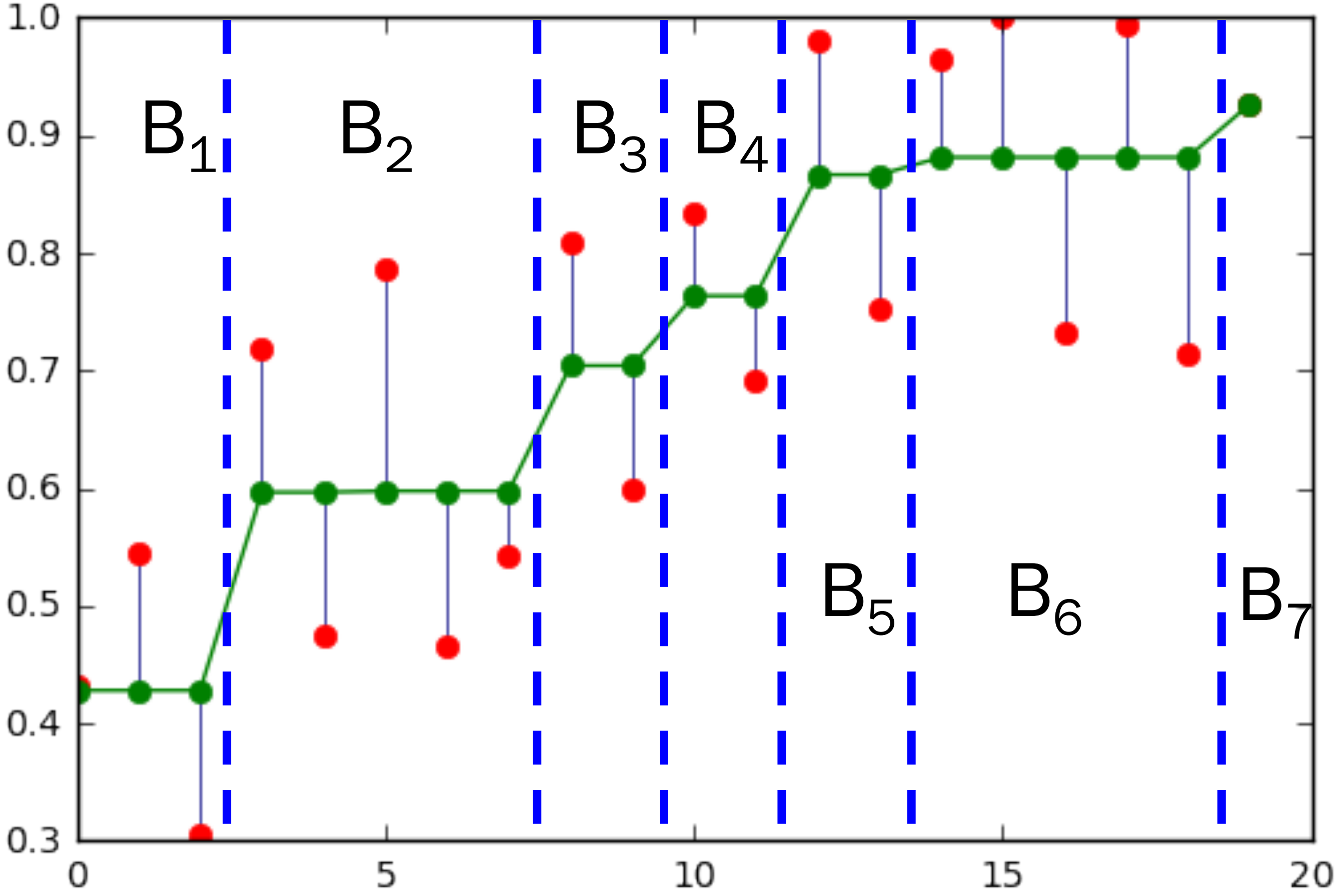


$$\mathbf{y}^* = \underset{\mathbf{y} \in \mathcal{O}}{\operatorname{argmin}} ||\mathbf{y} - \mathbf{x}||_2^2$$

where $\mathcal{O} = \{\mathbf{y} : y_1 \leq \cdots \leq y_p\}$

Isotonic Regression

$$\mathbf{y}^* = \underset{\mathbf{y} \in \mathcal{O}}{\operatorname{argmin}} \, ||\mathbf{y} - \mathbf{x}||_2^2$$



$$C_k = \begin{pmatrix} \frac{1}{k} & \cdots & \frac{1}{k} \\ \vdots & & \vdots \\ \frac{1}{k} & \cdots & \frac{1}{k} \end{pmatrix}$$

$$\frac{\partial \mathbf{y}^*}{\partial \mathbf{x}} = \Lambda(\mathbf{y}^*) = \operatorname{blockdiag}(C_{|B_1|}, \ldots, C_{|B_m|})$$

Differentiable learning of PSMs

Theorem: If $\nabla_\theta F(A \mid \theta)$ exists for all $A \subseteq V$, then the approximate Jacobians

$$J_\sigma = \frac{\partial}{\partial \theta} \operatorname*{argmin}_{\mathbf{y} \in \mathcal{O}(\sigma)} ||\mathbf{f}_\sigma(\theta)^T + \mathbf{y}||_2^2$$

are independent of σ. Can multiply in linear time.

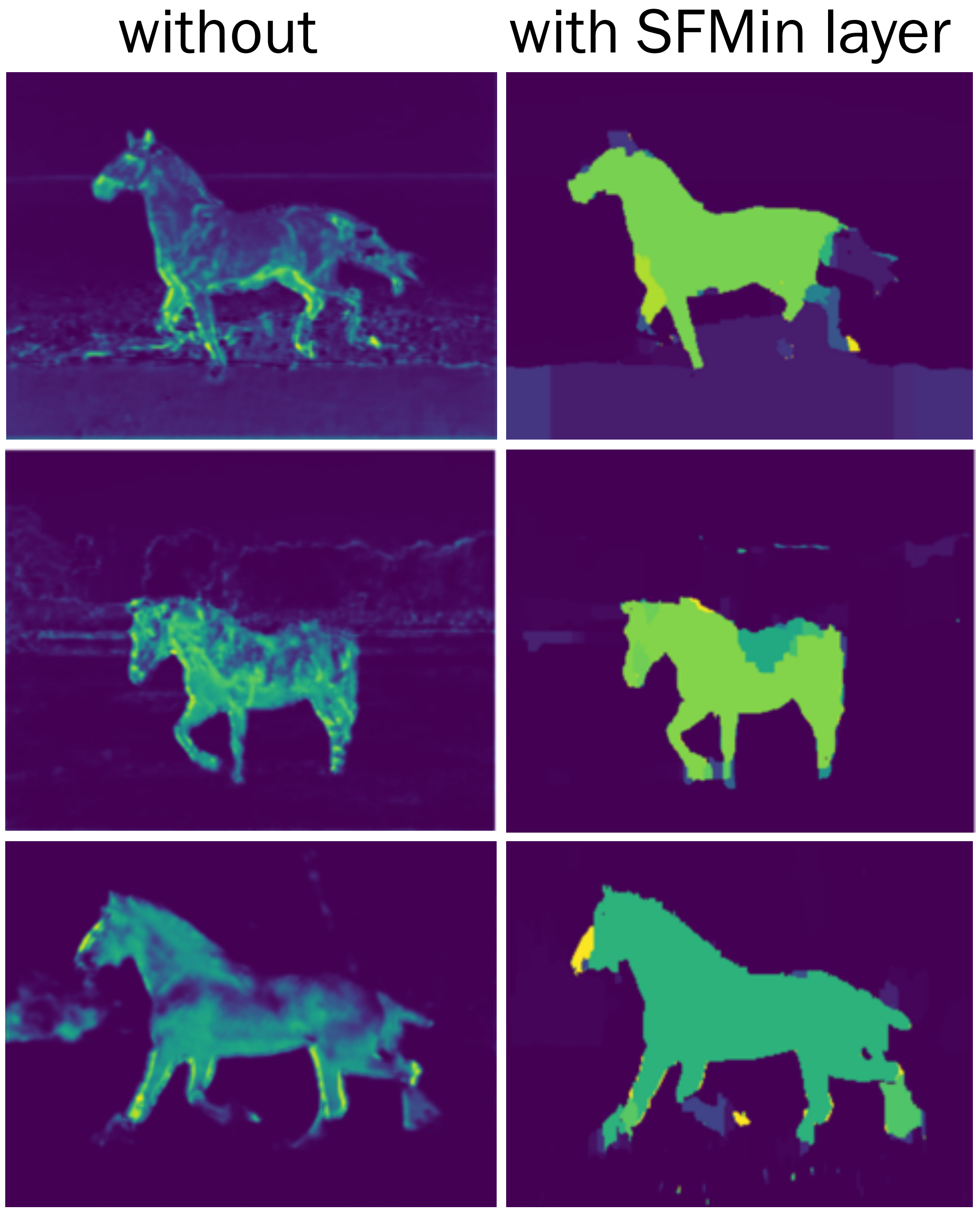
Theorem: Under some conditions* the approximation is exact!

Theorem: For mixtures $F(S \mid \theta) = \sum_{i=1}^{k} \theta_i F_i(S)$

can* compute the exact Jacobian in polynomial time

*see paper

Application: Segmentation



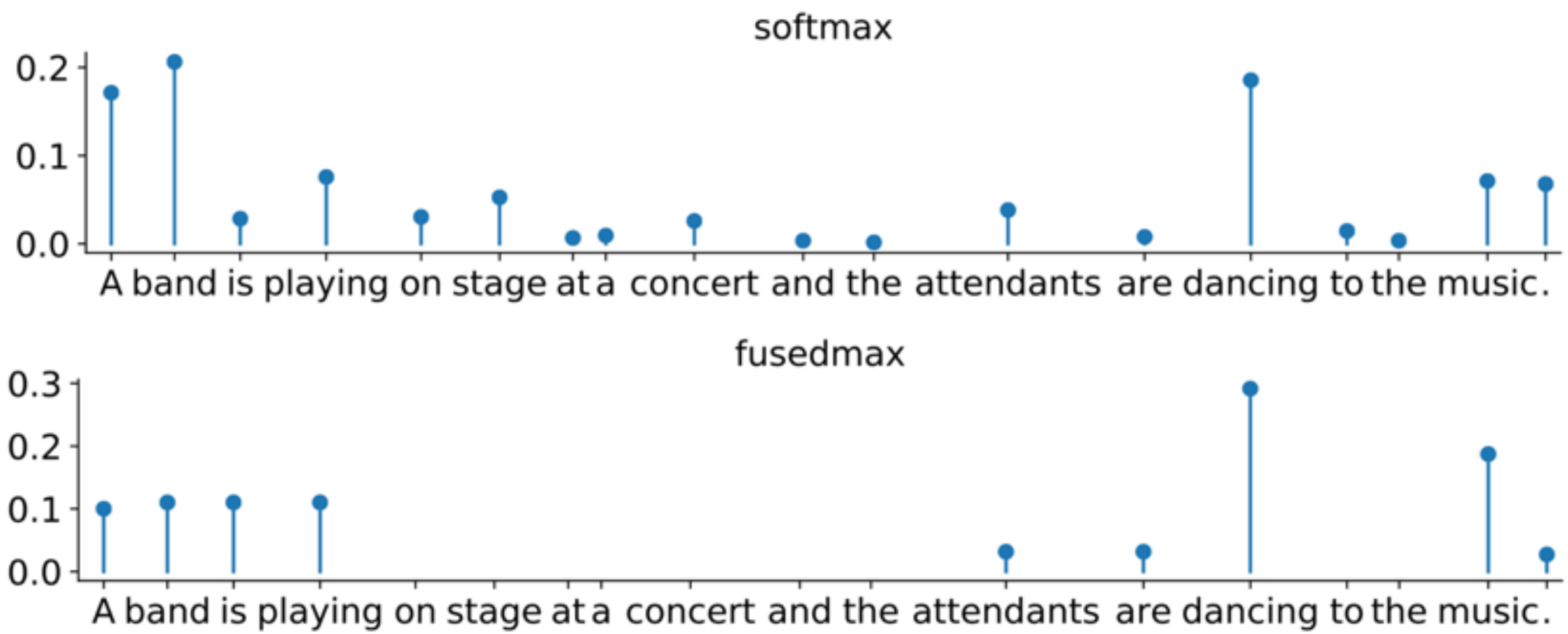
Trained on only 0.1% of labeled pixels!

	CNN	CNN+ SFMin
Acc.	.81	.91
NLL	.39	.27

Application: Textual entailment

[Niculae and Blondel NIPS ’17]

- *Fusedmax* attn. mechanism of Niculae and Blondel is a special case, obtained by concatenating 2 SFMin layers
- **Task**: Does sentence *T* entail hypothesis *H* (here *H*=“no one is dancing”)



Differentiable submodular maximization

- Similar results for submodular maximization [with Tschiatschek, Sahin, IJCAI'18]
- Key idea: Directly define a distribution over sets through the (double) greedy algorithm
- Tractable, differentiable likelihood
  ➔ Gradient-based learning!
- Applications to recommender systems and image collection summarization

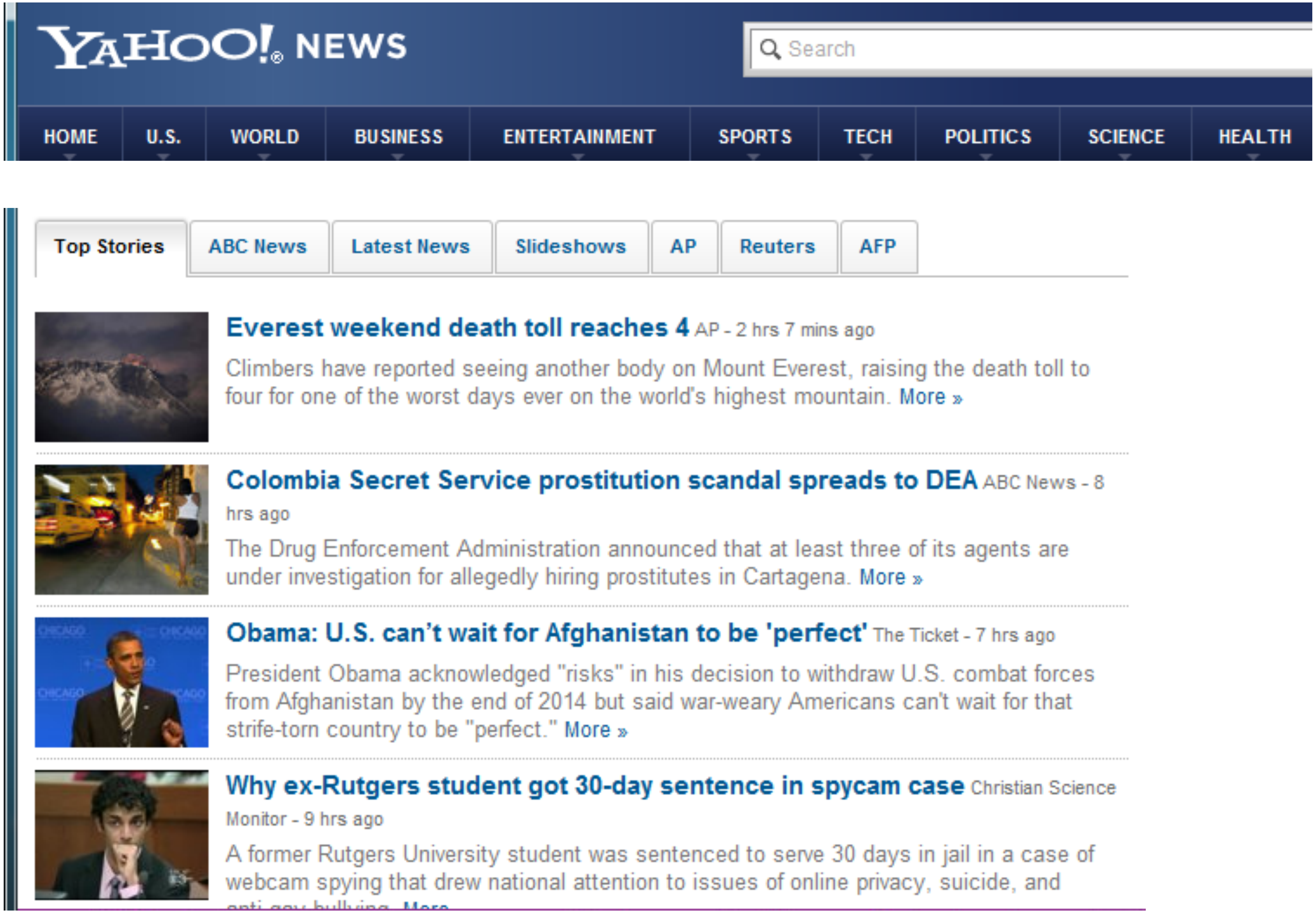
Submodularity and Interactive Learning

Learning to optimize submodular functions

- Online submodular optimization
  - Learn to pick a sequence of sets to maximize a sequence of (unknown) submodular functions
  - *Application*: Making diverse recommendations

- Adaptive submodular optimization
  - Gradually build up a set, taking into account feedback
  - *Application*: Experimental design / Active learning / Active Teaching / ...

News recommendation



Application: Diverse Recommendations

“Google to DOJ: Let us prove to users that NSA isn't snooping on them”
“US tech firms push for govt transparency on securityReuters”
“Internet Companies Call For More Disclosure of Surveillance”
“NSA scandal: Twitter and Microsoft join calls to disclose data requests”
“NSA Secrecy Prompts a Pushback”

“Google to DOJ: Let us prove to users that NSA isn't snooping on them”
“Storms Capable of Producing Derecho Possible in Midwest Today”
“Ohio kidnap suspect pleads not guilty”
“Five takeaways from Spurs-Heat in Game 3 of the NBA Finals”
“Samsung Unveils Galaxy S4 Zoom With 16MP Camera”

Prefer recommendations that are both *relevant* and *diverse*

[Yue]

Simple model

- We‘re given a set of articles V
- Each round:
  - A user appears, interested in a subset S_t of the articles
  - We recommend a set of articles A_t
  - The user clicks on any displayed article that she is interested in

$$F_t(A_t) = \min(|A_t \cap S_t|, 1)$$

- **Goal:** Maximize the total #of clicks $\sum_t F_t(A_t)$
- **Challenge:**
  - We don‘t know which articles the user is interested in!

Online maximization of submodular functions [Streeter, Golovin NIPS '08]

Pick sets	A_1	A_2	A_3	…	A_T	Observe *either* F_t, or only $F_t(A_t)$
SFs	F_1	F_2	F_3	…	F_T	
Reward	$r_1=F_1(A_1)$	r_2	r_3	…	r_T	Total: $\sum_t r_t \rightarrow \max$

Time

Goal: Want to choose $A_1,\ldots A_t$ s.t. the regret

$$R_T = \max_{|A|\leq k} \sum_{t=1}^{T} F_t(A) - \sum_{t=1}^{T} F_t(A_t)$$

grows sublinearly, i.e., $R_T/T \rightarrow 0$

For k=1, many good algorithms known! ☺

But what if k>1?

Online Greedy Algorithm

[Streeter & Golovin, NIPS `08]

Replace each stage of greedy algorithm with a multi-armed bandit algorithm.

Select $\{a_1, \quad a_2, \quad a_3, \quad \ldots, \quad a_k\}$

Feedback to $\mathcal{E}_j$ for action $\mathbf{a}_j$ is (unbiased est. of)

$F_t(\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_{j-1}, \mathbf{a}_j\}) - F_t(\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_{j-1}\})$

Online maximization of submodular functions
[Streeter, Golovin NIPS '08]

Theorem
Online greedy algorithm chooses $A_1,\ldots,A_T$ s.t. for any sequence $F_1,\ldots,F_T$

$$\sum_{t=1}^{T} F_t(A_t) \geq (1 - 1/e) \max_{|A|\leq k} \sum_{t=1}^{T} F_t(A) - O\left(nT^{2/3}\right)$$

Can get 'no-regret' over greedy algorithm in hindsight

I.e., can learn ``enough'' about F to optimize greedily!

Stochastic linear submodular bandits

[Yue & Guestrin '11]

- Basic submodular bandit algorithm has slow convergence
- Can do better if we make stronger assumptions
  - Submodular function is linear combination of m SFs

$$F(S) = \sum_{i=1}^{m} w_i F_i(S)$$

  - We evaluate it up to (stochastic) noise*

$$F_t(S) = F(S) + \text{noise}$$

➔ LSBGreedy algorithm

*some independence conditions

User Study [Yue & Guestrin '11]

- Real data: >10k articles
- T=10 days, rec. 10 articles per day

- 27 users rate articles, aim to maximize #likes

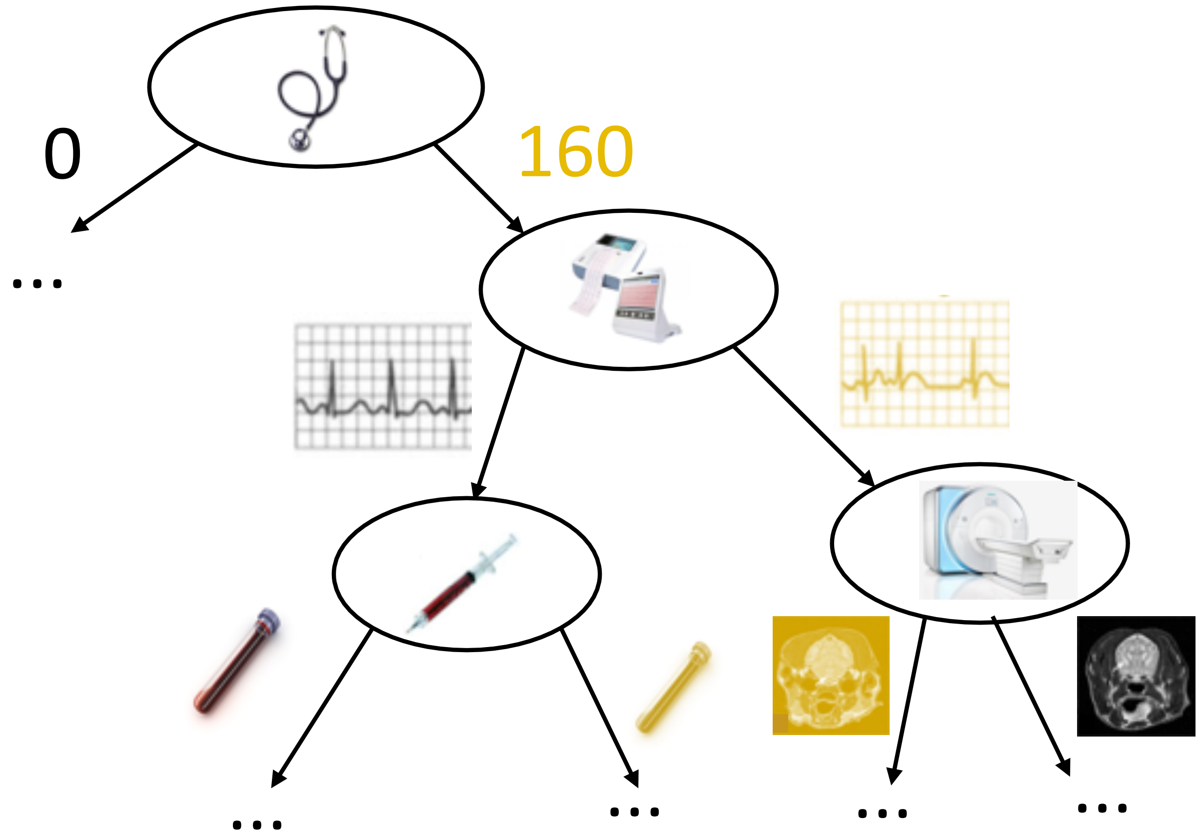
> "Google to DOJ: Let us prove to users that NSA isn't snooping on them" ✓
> "Storms Capable of Producing Derecho Possible in Midwest Today" ✓
> "Ohio kidnap suspect pleads not guilty" ✗
> "Five takeaways from Spurs-Heat in Game 3 of the NBA Finals" ✓
> "Samsung Unveils Galaxy S4 Zoom With 16MP Camera" ✗

- LSBGreedy outperforms baselines that fail to …
  - adapt weights (no personalization)
  - address the exploration—exploitation tradeoff
  - model diversity explicitly

Other results on online submodular optimization

- Online submodular maximization
  - No (1-1/e) regret for ranking, matroids [Streeter, Golovin, Krause 2009, 2014]
  - Kernelized submodular bandits [Chen, Krause, Karbasi '2017]
  - Online continuous submodular optimization [Chen, Hassani, Karbasi '2018]
- Online submodular coverage
  - Min-cost / Min-sum submodular cover [Streeter & Golovin NIPS 2008, Guillory & Bilmes NIPS 2011]
- Online Submodular Minimization
  - Unconstrained [Hazan & Kale NIPS 2009]
  - Constrained [Jegelka & Bilmes ICML 2011]
- See also the „submodular secretary problem“

Active learning / diagnosis



Is there a notion of submodularity for sequential decision tasks?

Problem Statement

Given:

- Items (tests, experiments, unlabeled ex., …) $V=\{1,\ldots,n\}$
- Associated with random variables $X_1,\ldots,X_n$ taking values in O
- Objective: $f : 2^V \times O^V \to \mathbb{R}$

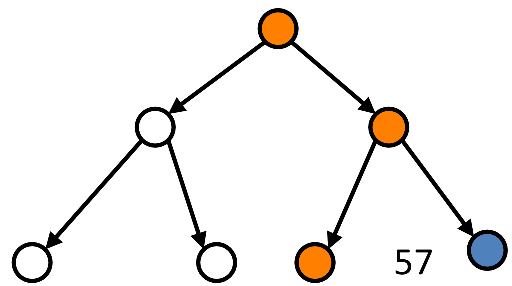
Want: Policy π that maps observation $\mathbf{x}_A$ to next item

Value of policy π: $$F(\pi) = \sum_{\mathbf{x}_V} P(\mathbf{x}_V) f(\pi(\mathbf{x}_V), \mathbf{x}_V)$$

Tests run by π
if world in state $\mathbf{x}_V$

Want $$\pi^* \in \operatorname*{argmax}_{|\pi| \le k} F(\pi)$$

NP-hard (also hard to approximate!)



Adaptive greedy *policy*

- Suppose we've seen $X_A = x_A$.
- Conditional expected benefit of adding item *s*:

$$\Delta(s \mid \mathbf{x}_A) = \mathbb{E}\Big[\underbrace{f(A \cup \{s\}, \mathbf{x}_V) - f(A, \mathbf{x}_V)}_{\text{Benefit if world in state } \mathbf{x}_V} \mid \mathbf{x}_A\Big]$$

Conditional on observations $\mathbf{x}_A$

Adaptive Greedy policy:

Start with $A = \emptyset$

For i = 1:k

– Pick $s_k \in \operatorname{argmax}_s \Delta(s \mid \mathbf{x}_A)$

– Observe $X_{s_k} = x_{s_k}$

– Set $A \leftarrow A \cup \{s_k\}$

When does this adaptive greedy policy work?

Adaptive submodularity

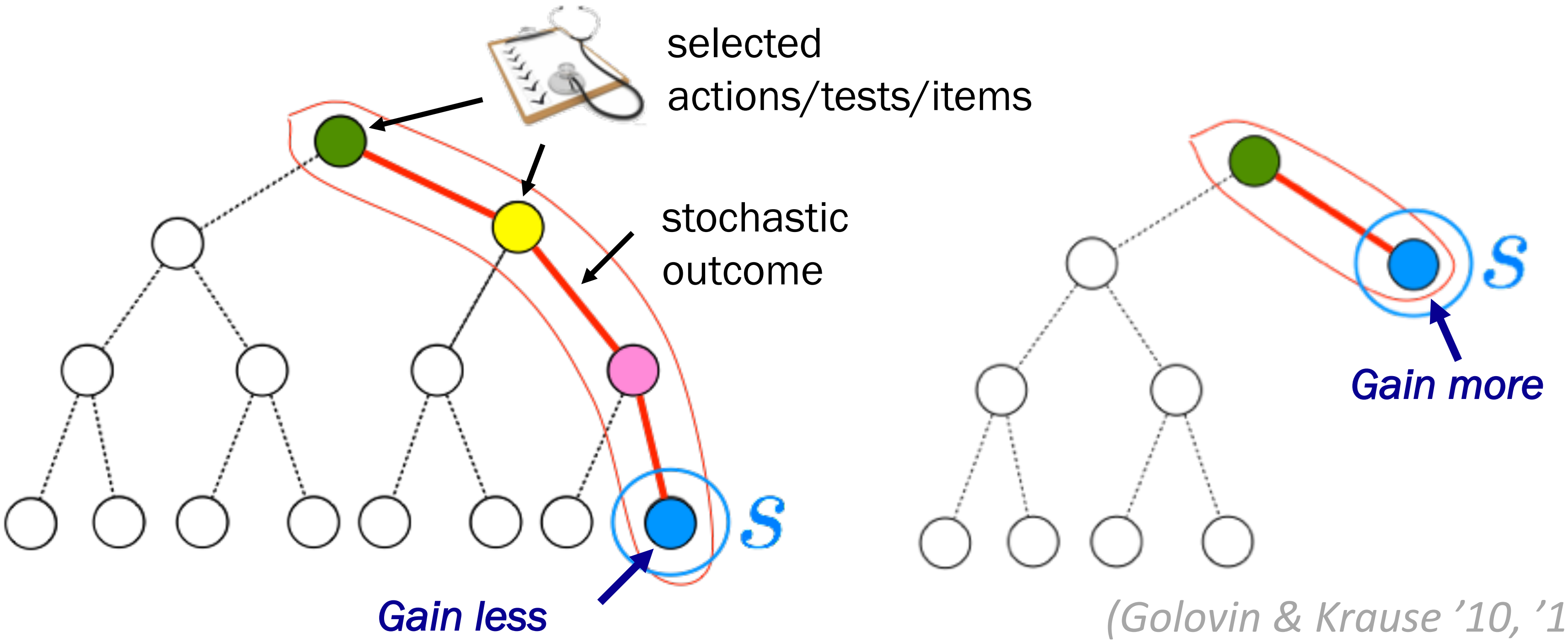
Adaptive monotonicity:

$$\Delta(s \mid \mathbf{x}_A) \geq 0$$

x_B observes more than x_A

Adaptive submodularity:

$$\Delta(s \mid \mathbf{x}_A) \geq \Delta(s \mid \mathbf{x}_B) \quad \text{whenever} \quad \mathbf{x}_A \preceq \mathbf{x}_B$$



(Golovin & Krause '10, '11)

Adaptive submodularity

Theorem: If *f* is adaptive submodular and adaptive monotone w.r.t. to distribution *P*, then

$$F(\pi_{\text{Greedy}}) \geq (1 - 1/e) F(\pi_{\text{OPT}})$$

Strictly generalizes (Nemhauser, Wolsey & Fisher '78)

Many other results can be "lifted" to the adaptive setting

(Golovin & Krause '10, '11)

ETH

From sets to policies

Submodularity → Adaptive submodularity

Submodularity

Applies to: set functions

$$\Delta_F(s \mid A) = F(A \cup \{s\}) - F(A)$$

$$\Delta_F(s \mid A) \geq 0$$

$$A \subseteq B \Rightarrow \Delta_F(s \mid A) \geq \Delta_F(s \mid B)$$

$$\max_A F(A)$$

Greedy algorithm provides

- *(1-1/e)* for max. w card. const.
- *1/(p+1)* for p-indep. systems
- *log Q* for min-cost-cover
- *4* for min-sum-cover

Adaptive submodularity

policies, value functions

$$\Delta_F(s \mid \mathbf{x}_A) = \mathbb{E}\Big[f(A \cup \{s\}, \mathbf{x}_V) - f(A, \mathbf{x}_V) \mid \mathbf{x}_A\Big]$$

$$\Delta_F(s \mid \mathbf{x}_A) \geq 0$$

$$\mathbf{x}_A \preceq \mathbf{x}_B \Rightarrow \Delta_F(s \mid \mathbf{x}_A) \geq \Delta_F(s \mid \mathbf{x}_B)$$

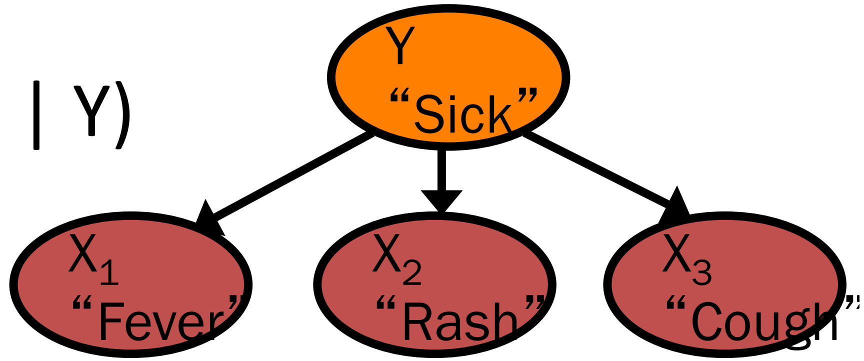
$$\max_\pi F(\pi)$$

Greedy policy provides

- *(1-1/e)* for max. w card. const.
- *1/(p+1)* for p-indep. systems
- *log^2 Q* for min-cost-cover*
- *4* for min-sum-cover

Optimal Diagnosis

- Prior over diseases P(Y)
- Deterministic test outcomes P($\mathbf{X}_V$ | Y)

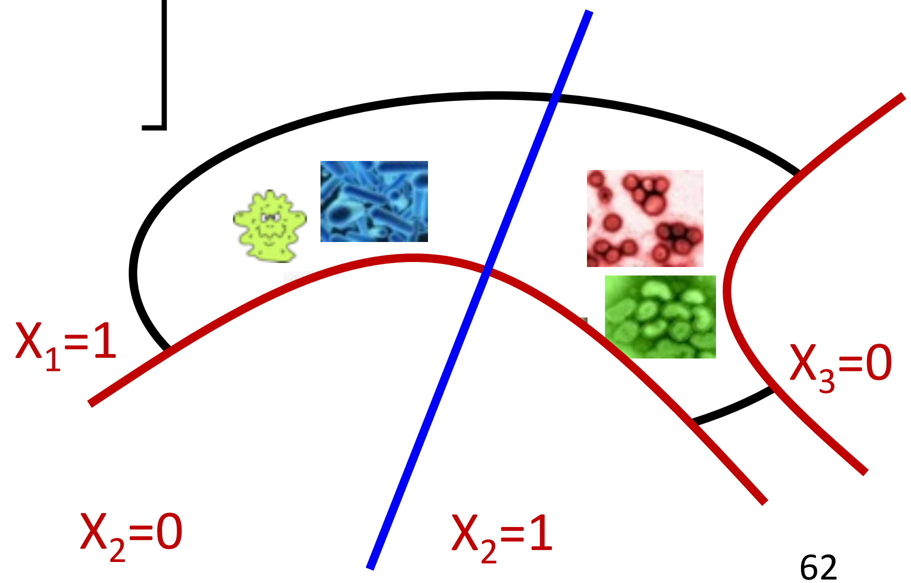


- How should we test to eliminate all incorrect hypotheses?

$$\Delta(t \mid x_A) = \mathbb{E}\left[\begin{array}{l}\text{mass ruled out}\\ \text{by } t \text{ if we}\\ \text{know } x_A\end{array}\right]$$

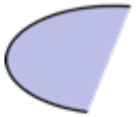
"Generalized binary search"

Equivalent to max. infogain



OD is Adaptive Submodular

Objective = probability mass of hypotheses you have ruled out.

$b_0 := \mathbb{P}($  $)$

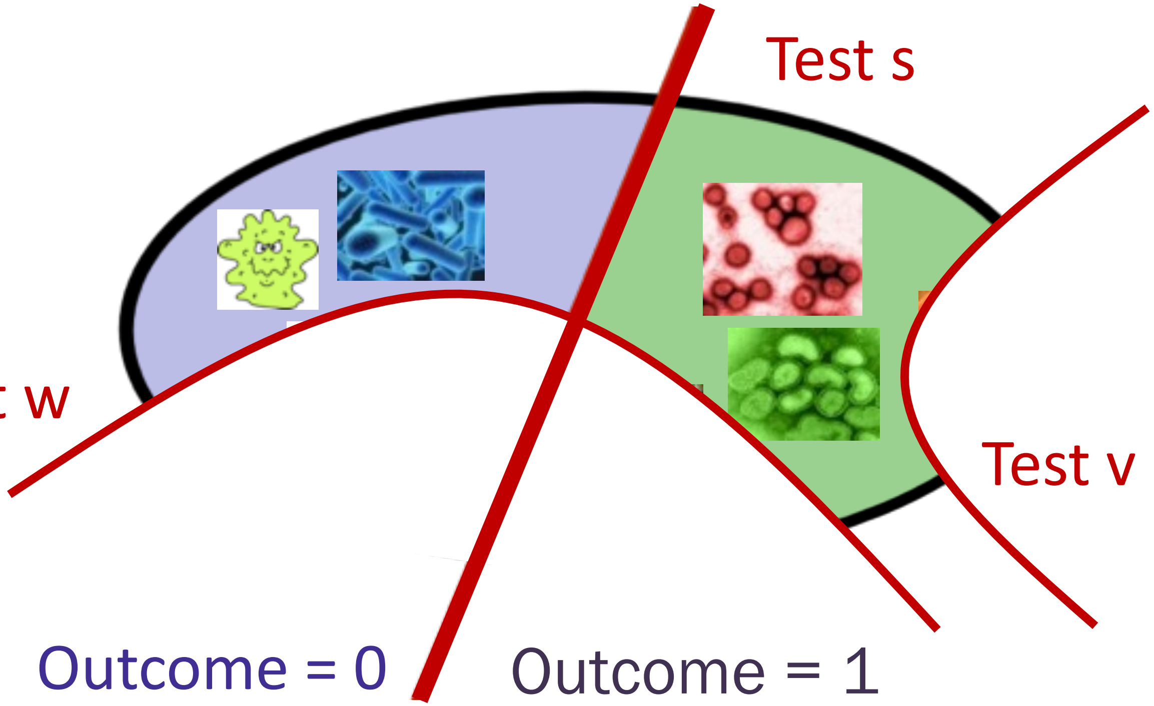
$g_0 := \mathbb{P}($  $)$

$$\Delta(s \mid \{\}) = \frac{2 g_0 b_0}{g_0 + b_0}$$

$b_1 := \mathbb{P}($  $)$

$g_1 := \mathbb{P}($  $)$

$$\Delta(s \mid \mathbf{x}_{v,w}) = \frac{2 g_1 b_1}{g_1 + b_1}$$

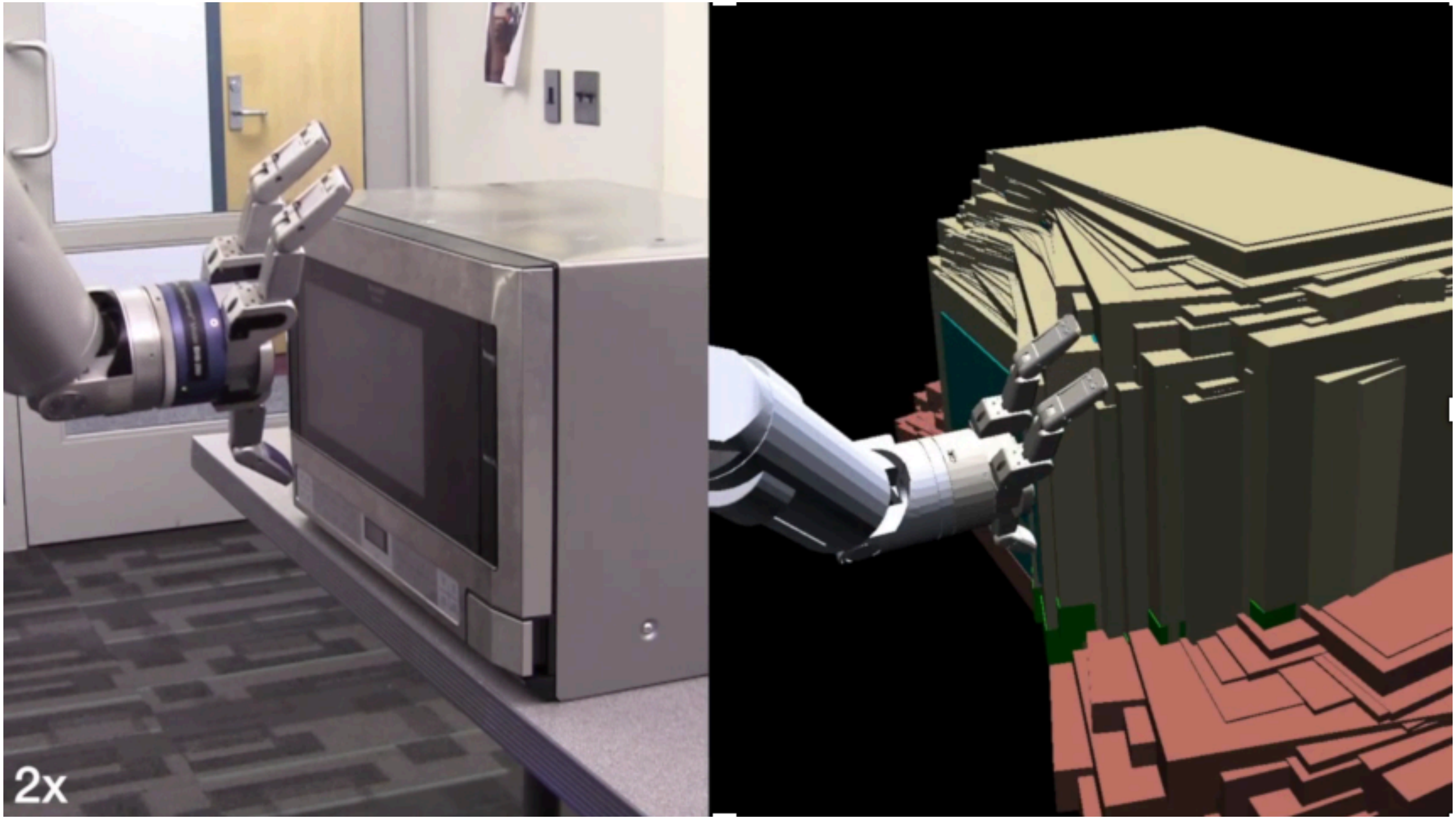


$b_0 \geq b_1, \quad g_0 \geq g_1$

Not hard to show that

$$\Delta(s \mid \{\}) \geq \Delta(s \mid \mathbf{x}_{v,w})$$

Application: Touch-based localization



(Chen-Javdani-Karbasi-Bagnell-Srinivasa-Krause '15)

Application: Adaptive teaching
[Hunziker, Singla et al, arXiv 2018]

toy

Spielzeug

dessert

Nachtisch

Given limited instruction time and multiple concepts to learn, what is a good **learning schedule**?

How should we **adapt** the learning schedule based on the learner‘s performance history?

ETH

Sequential decision making with SFs

- Adaptive submodularity / interactive submodular cover
  (Golovin & Krause'10; Guillory & Bilmes'10)
- Online learning with submodular functions
  (Golovin & Streeter '08; Hazan & Kale '09)
- Submodular secretary problems
  (Bateni-Hajiaghayi-Zadimoghaddam'09)
- Streaming algorithms for submodular optimization
  (Gomes & Krause'10, Chakrabarti & Kale'13, Badanidiyuru-Mirzasoleiman-Karbasi-Krause'14)
- Submodular functions over sequences
  (Zhang-Wang-Chong-Pezeshki-Moran'13; Tschiatschek-Singla-Krause'17)

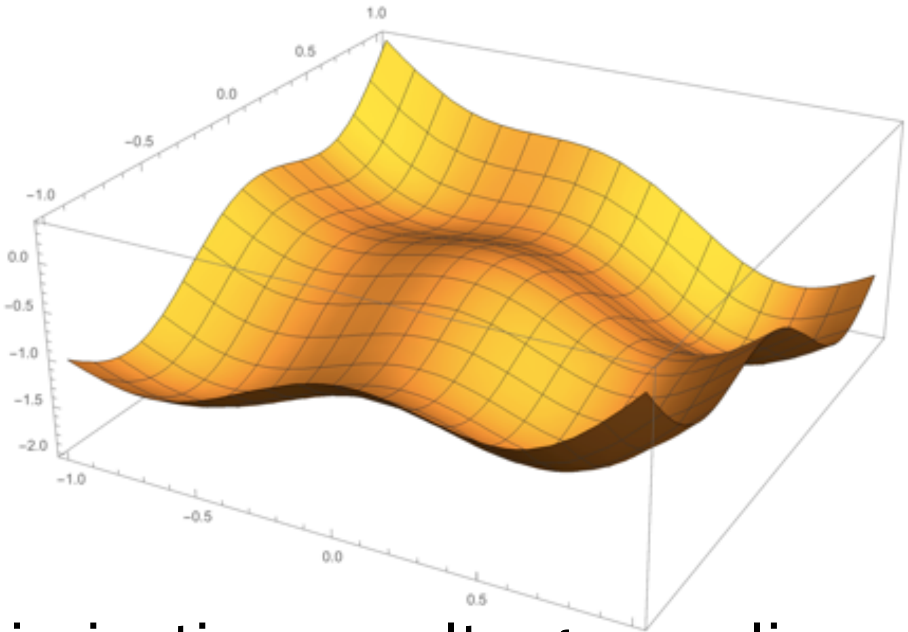
Continuous Submodularity and non-convex optimization

Submodularity more generally

- Lattices and continuous functions

$$f(x) + f(y) \;\geq\; f(x \vee y) + f(x \wedge y)$$

subclass: diminishing returns (DR) – submodular fn's



- Many optimization results generalize

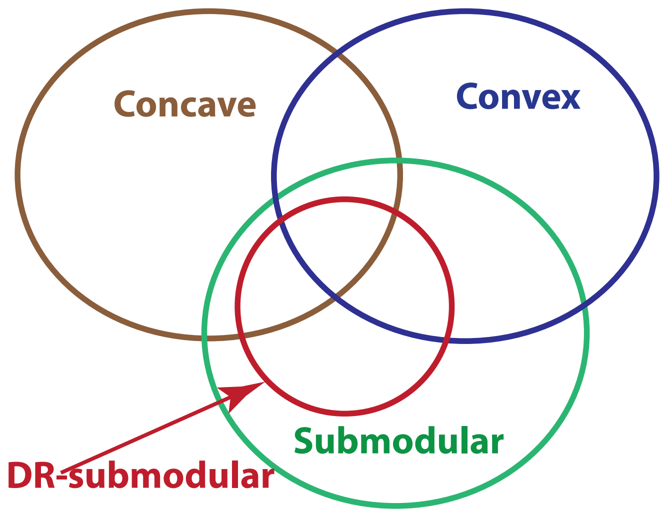
(Milgrom-Shannon 94; Topkis 98; Murota 03; Kapralov-Post-Vondrak 10; Soma et al 2014-16; Bach 2015; Ene & Nguyen 2016; Bian-Mirzasoleiman-Buhmann-Krause 16)

Characterizations - Overview

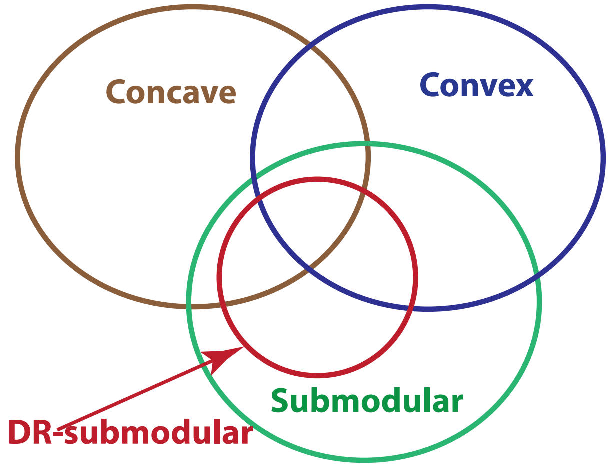
Condition	Submodular $f(\cdot)$	Convex $g(\cdot), \lambda \in [0,1]$
0^{th} order	$f(\boldsymbol{x}) + f(\boldsymbol{y}) \geq f(\boldsymbol{x} \vee \boldsymbol{y}) + f(\boldsymbol{x} \wedge \boldsymbol{y})$	$\lambda g(\boldsymbol{x}) + (1-\lambda) g(\boldsymbol{y}) \geq g(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y})$
1^{st} order	**weak DR** (Diminishing Returns)	$g(\boldsymbol{y}) - g(\boldsymbol{x}) \geq \langle \nabla g(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle$
2^{nd} order	$\dfrac{\partial^2 f(\boldsymbol{x})}{\partial x_i \partial x_j} \leq 0, \forall i \neq j$	$\nabla^2 g(\boldsymbol{x}) \succcurlyeq 0$ (PSD)

$\vee$: coordinate-wise max. ("JOIN" in lattice theory)
$\wedge$: coordinate-wise min. ("MEET" in lattice theory)

$\boldsymbol{x}$	$\boldsymbol{y}$	$\boldsymbol{x} \vee \boldsymbol{y}$	$\boldsymbol{x} \wedge \boldsymbol{y}$
2	1	2	1
0	2	2	0
4	3	4	3



Submodular & DR-Submodular



$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T H \boldsymbol{x} + h^T \boldsymbol{x} + c,$$

$$H = \begin{bmatrix} -1 & -2 \\ -2 & -1 \end{bmatrix},$$ eigenvalues: $\binom{1}{-3}$

Condition	Submodular $f(\cdot)$	DR-Submodular $f'(\cdot)$
0th order	$f(\boldsymbol{x}) + f(\boldsymbol{y}) \geq f(\boldsymbol{x} \vee \boldsymbol{y}) + f(\boldsymbol{x} \wedge \boldsymbol{y})$	$f'(\boldsymbol{x}) + f'(\boldsymbol{y}) \geq f'(\boldsymbol{x} \vee \boldsymbol{y}) + f'(\boldsymbol{x} \wedge \boldsymbol{y})$ & coordinate-wise concave
1st order	weak DR	DR
2nd order	$\dfrac{\partial^2 f(\boldsymbol{x})}{\partial x_i \partial x_j} \leq 0, \forall i \neq j$	$\dfrac{\partial^2 f'(\boldsymbol{x})}{\partial x_i \partial x_j} \leq 0, \forall i, j$

1st Order Condition – Diminishing Returns

weak DR: $\forall \boldsymbol{a} \leq \boldsymbol{b}, \forall i \; s.t. \; a_i = b_i, \forall k \geq 0,$ it holds,

$$f(k\boldsymbol{e}_i + \boldsymbol{a}) - f(\boldsymbol{a}) \geq f(k\boldsymbol{e}_i + \boldsymbol{b}) - f(\boldsymbol{b})$$

why called 1st order? implies the relation between the directional derivatives in directions $\boldsymbol{e}_i$: $\nabla_{\boldsymbol{e}_i} f(\boldsymbol{a}) \geq \nabla_{\boldsymbol{e}_i} f(\boldsymbol{b})$

Lemma: Submodularity ⇔ **weak DR**

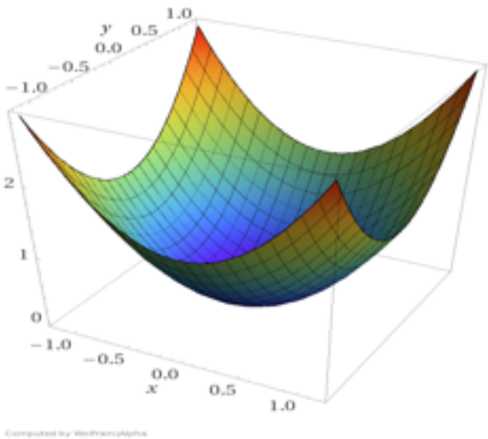
Applies for all submodular set, integer-lattice and continuous functions

DR: $\forall \boldsymbol{a} \leq \boldsymbol{b}, \forall i, \quad \forall k \geq 0,$ it holds,

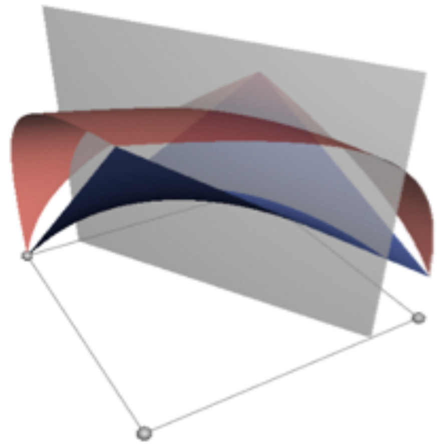
$$f(k\boldsymbol{e}_i + \boldsymbol{a}) - f(\boldsymbol{a}) \geq f(k\boldsymbol{e}_i + \boldsymbol{b}) - f(\boldsymbol{b})$$

Relation to Non-Convex Optimization

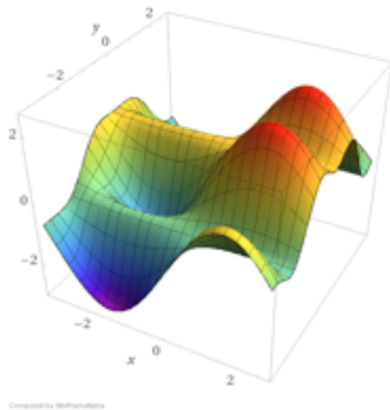
- In general, only guarantee converging to stationary points assuming smoothness
- Continuous Submodular Optimization: constant approximation guarantees with poly. algorithms



Convex



Continuous Submodular



Non-convex

A Summary of Main Results

Can minimize in polynomial time [Bach '15]	- Based on generalization of Lovász-extension
Monotone DR-submodular max. with down-closed convex constraints [Bian-Baharan-Buhmann-Krause '17]	- Hardness result: $1 - 1/e$ (unless RP=NP) - **Optimal** algorithm: A Frank-Wolfe Variant
Non-monotone DR-submodular max. with down-closed box constraints [Bian-Buhmann-Krause '18]	- Hardness result: $1/2$ (unless RP=NP) - **Optimal** algorithm: DR-DoubleGreedy
Non-monotone DR-submodular max. with general *convex* constraints [Bian-Levy-Krause-Buhmann '17]	- Hardness result: Open problem - Shrunken Frank-Wolfe: $1/e$ guarantee

ETH

What we did not cover

- Stochastic submodular optimization
- Learning submodular functions
  - Uniform approximation, PMAC model, optimization from samples
- Game theory
  - Equilibria in cooperative (supermodular) games / fair allocations
  - Price of anarchy in non-cooperative games
  - Mechanism design with submodular optimization
  - Solving submodular matrix games
- Generalizations of submodular functions
  - Bi-submodularity, tree-submodularity
  - Discrete convex analysis
  - XOS/Subadditive functions
  - Continuous submodular optimization
- Solving non-submodular problems via submodularity
  - Submodularity ratio / supermodular degree
  - Submodular surrogates
  - Submodular/supermodular procedure

Conclusions

- Discrete optimization abundant in ML applications
- Fortunately, some of those have structure: submodularity
- Submodularity can be exploited to develop efficient, scalable algorithms with strong guarantees
- Can handle complex constraints
- Useful for probabilistic inference, deep learning, interactive learning (online, adaptive, ...), ...