Quantum and Classical Algorithms for Approximate Submodular Function Minimization

Yassine Hamoudi, Patrick Rebentrost,

Ansis Rosmanis, Miklos Santha

arXiv: 1907.05378

1. Approximate Submodular Function Minimization

2. Quantum speed-up for Importance Sampling

1

Approximate Submodular Function Minimization

Submodular Function

A **submodular function** is a set function $F : 2^{[n]} \to \mathbb{R}$ satisfying the **diminishing returns property**:

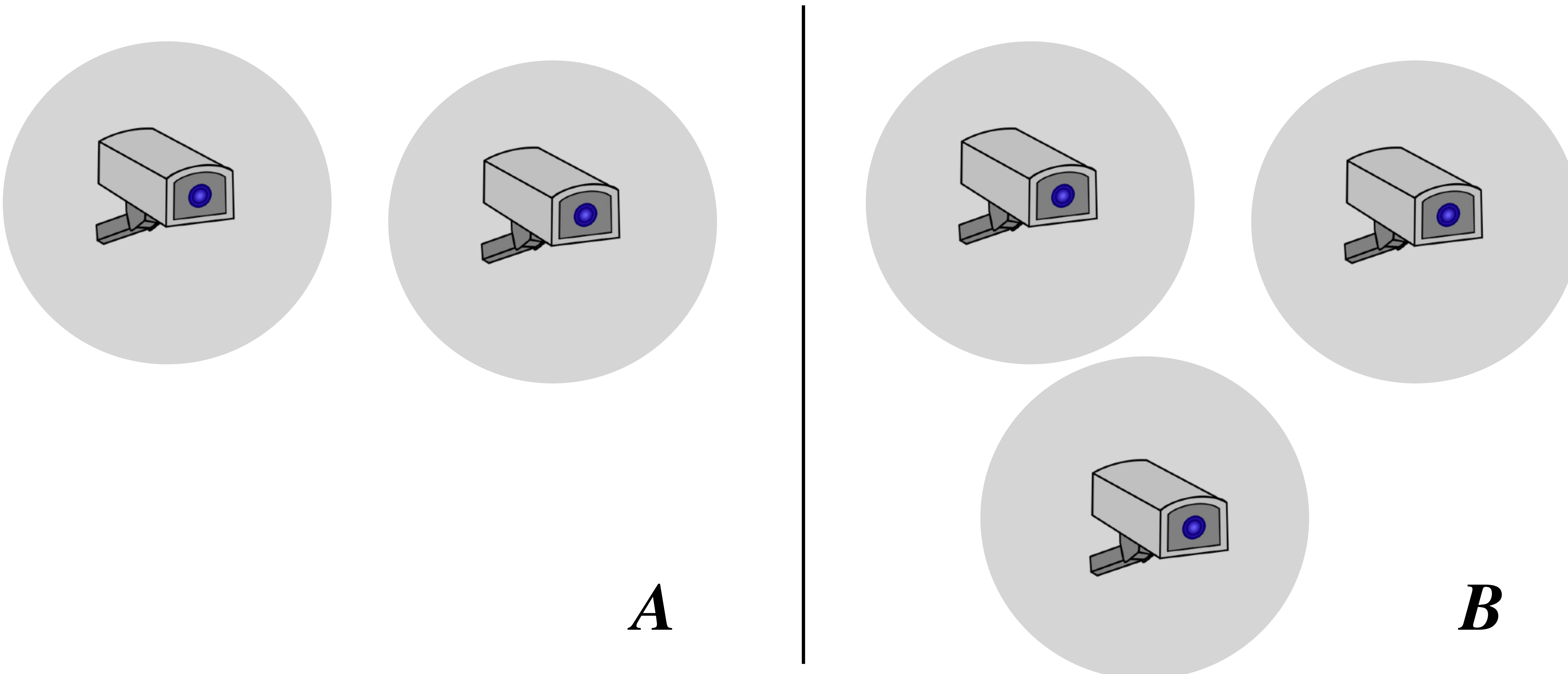
$$\forall A \subset B \subset [n] \text{ and } i \notin B, \quad F(A \cup \{i\}) - F(A) \geq F(B \cup \{i\}) - F(B)$$

Submodular Function

A **submodular function** is a set function $F : 2^{[n]} \to \mathbb{R}$ satisfying the **diminishing returns property**:

$$\forall A \subset B \subset [n] \text{ and } i \notin B, \;\; F(A \cup \{i\}) - F(A) \geq F(B \cup \{i\}) - F(B)$$

Example: area covered by cameras

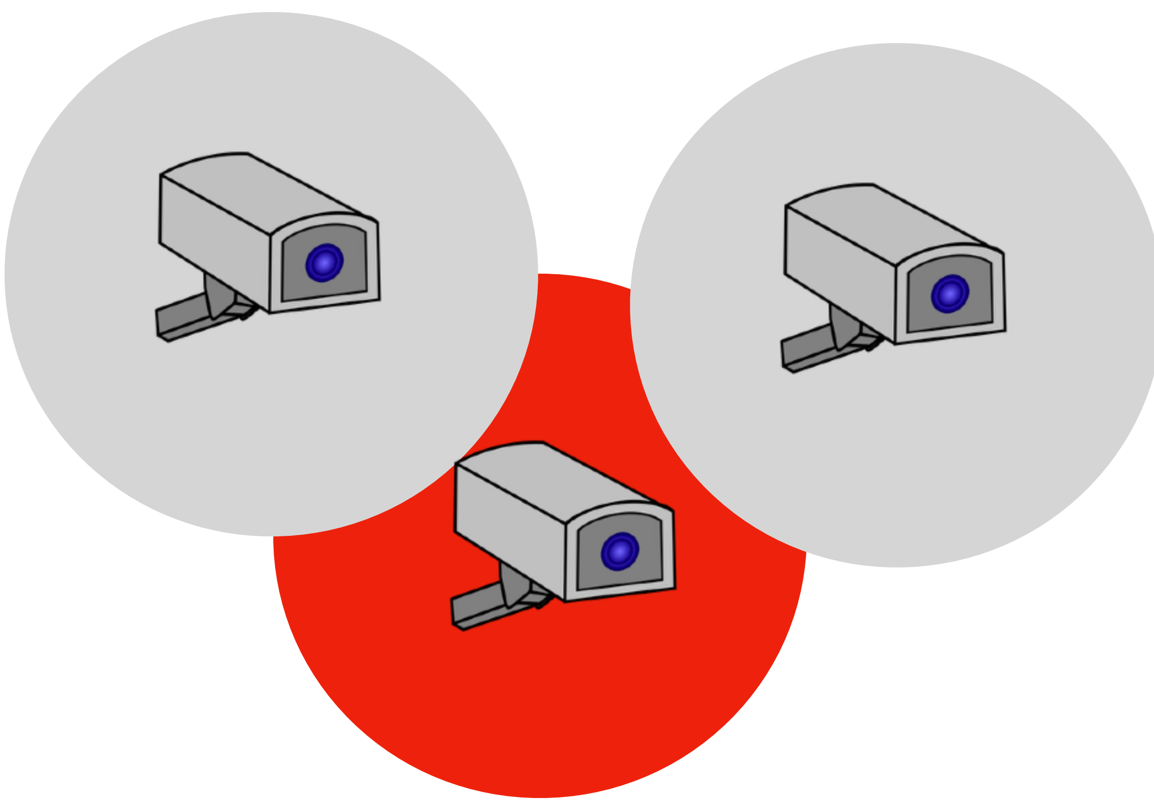


Submodular Function

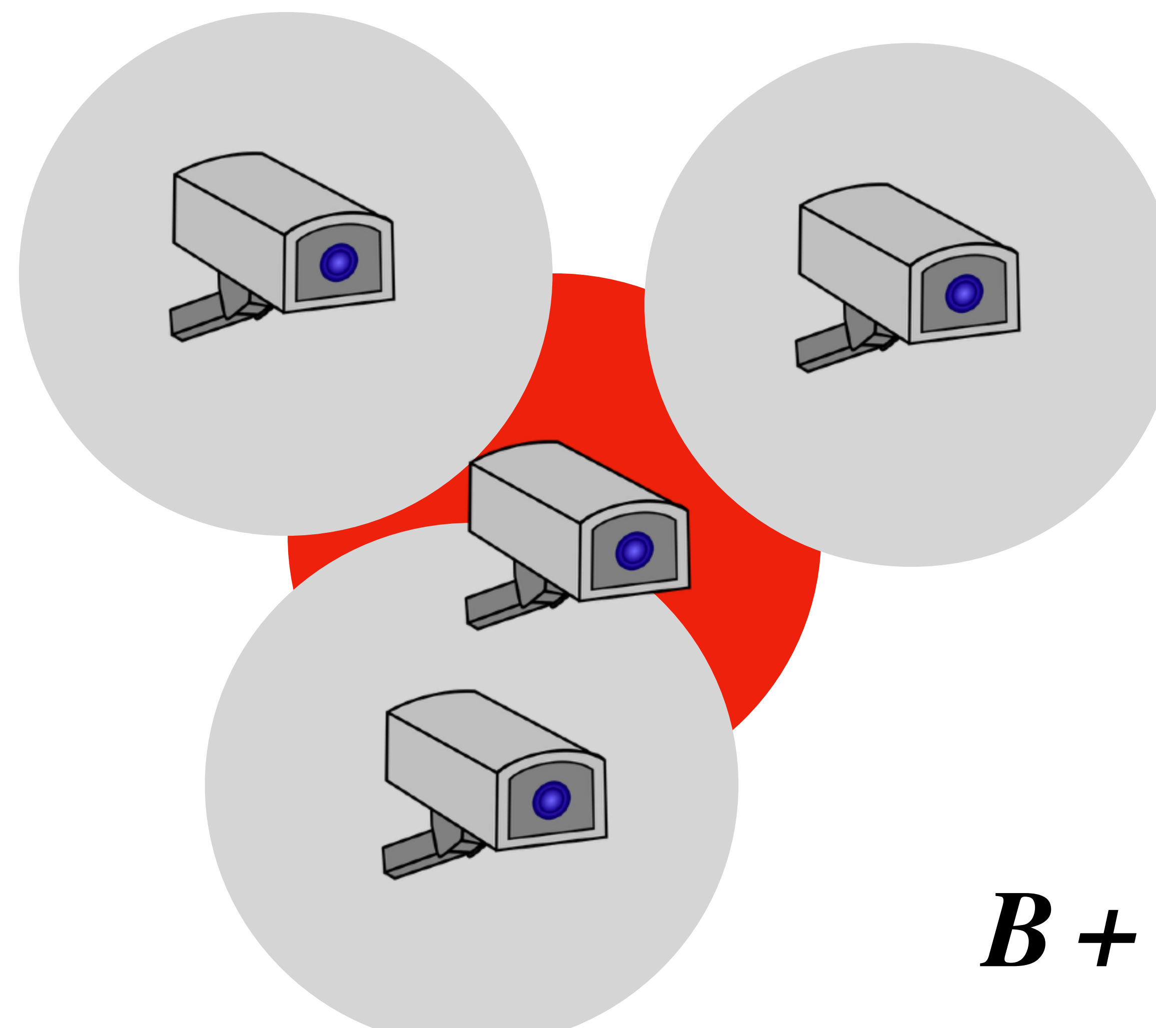
A **submodular function** is a set function $F : 2^{[n]} \to \mathbb{R}$ satisfying the **diminishing returns property**:

$$\forall A \subset B \subset [n] \text{ and } i \notin B, \;\; F(A \cup \{i\}) - F(A) \geq F(B \cup \{i\}) - F(B)$$

Example: area covered by cameras



$\boldsymbol{A + i}$



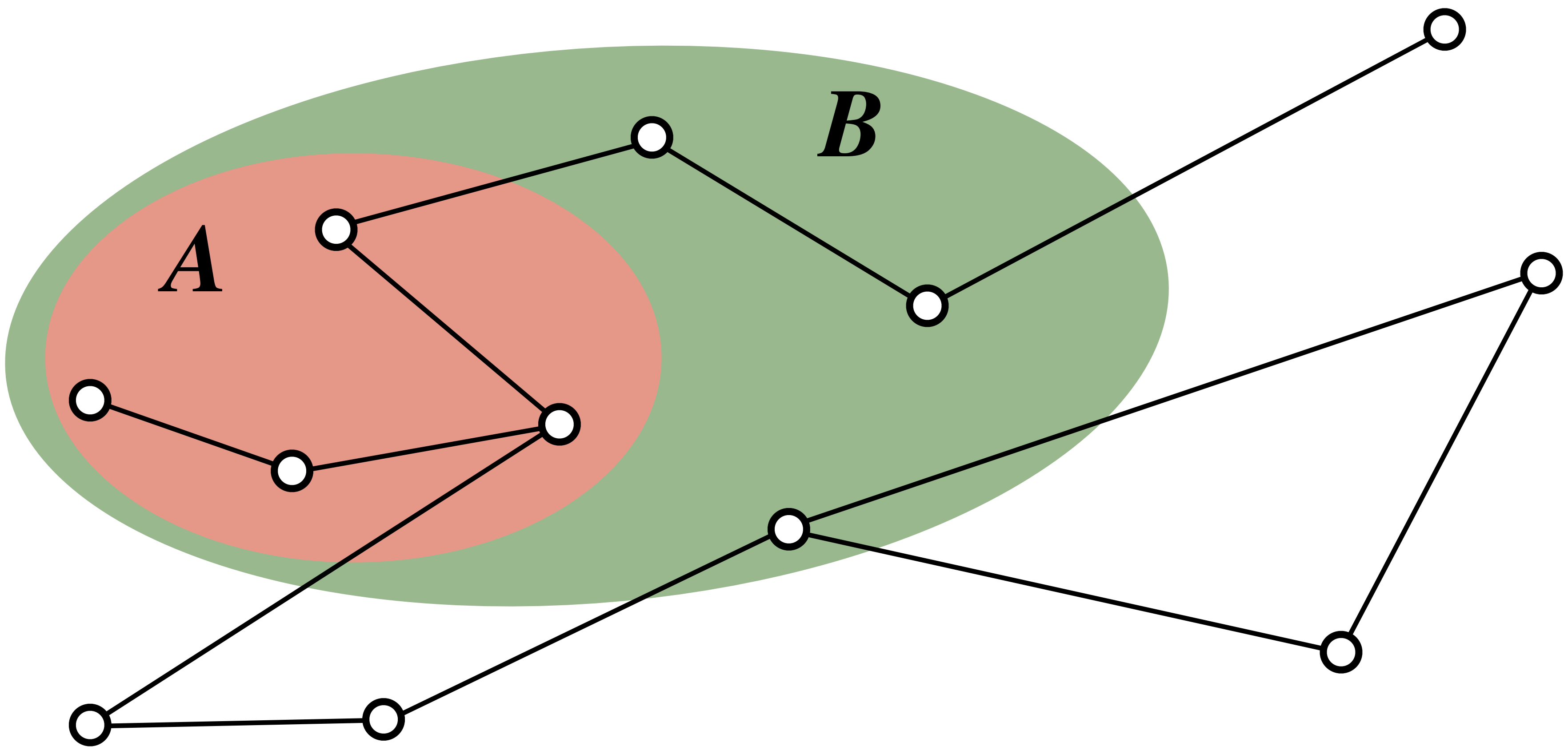
$\boldsymbol{B + i}$

Submodular Function

A **submodular function** is a set function $F : 2^{[n]} \to \mathbb{R}$ satisfying the **diminishing returns property**:

$$\forall A \subset B \subset [n] \text{ and } i \notin B, \quad F(A \cup \{i\}) - F(A) \geq F(B \cup \{i\}) - F(B)$$

Example: size of a cut



$|\mathrm{cut}(A)| = 2$

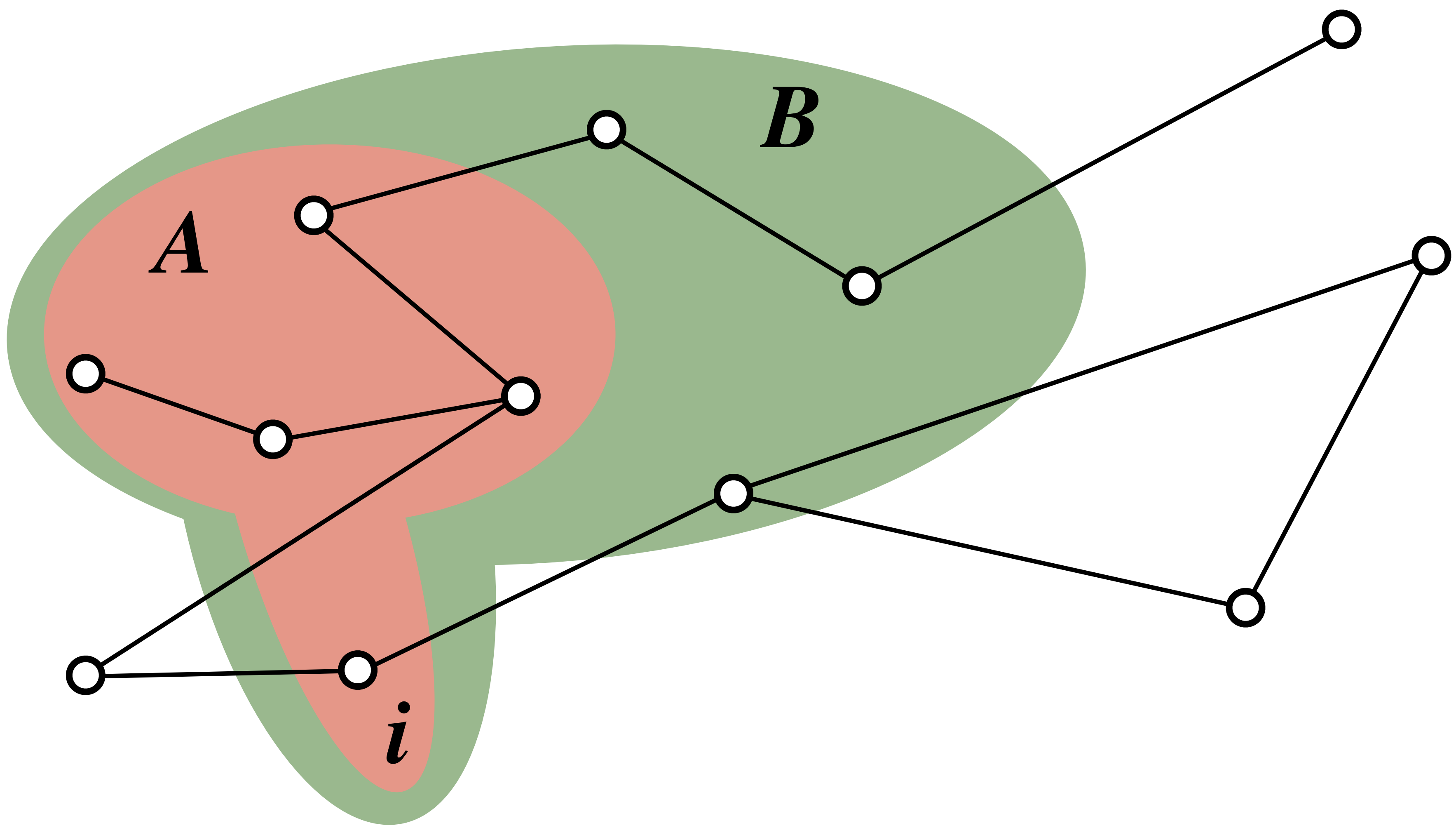
$|\mathrm{cut}(B)| = 5$

Submodular Function

A **submodular function** is a set function $F : 2^{[n]} \rightarrow \mathbb{R}$ satisfying the **diminishing returns property**:

$$\forall A \subset B \subset [n] \text{ and } i \notin B, \quad F(A \cup \{i\}) - F(A) \geq F(B \cup \{i\}) - F(B)$$

Example: size of a cut



|cut(A)| = 2

|cut(B)| = 5

|cut(A+i)| = 4

|cut(B+i)| = 6

Submodular Function

A **submodular function** is a set function $F : 2^{[n]} \to \mathbb{R}$ satisfying the **diminishing returns property**:

$$\forall A \subset B \subset [n] \text{ and } i \notin B, \quad F(A \cup \{i\}) - F(A) \geq F(B \cup \{i\}) - F(B)$$

Other examples:

- $F(S) = h(|S|)$ is submodular iff h **is concave**

Submodular Function

A **submodular function** is a set function $F : 2^{[n]} \to \mathbb{R}$ satisfying the **diminishing returns property**:

$$\forall A \subset B \subset [n] \text{ and } i \notin B, \quad F(A \cup \{i\}) - F(A) \geq F(B \cup \{i\}) - F(B)$$

Other examples:

- $F(S) = h(|S|)$ is submodular iff **h is concave**
- **rank** of a set of vectors
- **entropy** of random variables
- **coverage** functions
- …

Evaluation oracle access: given S obtain $F(S)$. (***time*** **=** *#queries to the oracle*)

Submodular Function Minimization

Evaluation oracle access: given S obtain $F(S)$. (***time*** = #*queries to the oracle*)

Submodular functions can be minimized in polynomial time

(Grotschel, Lovasz, Shrijver 1981)

Submodular Function Minimization

Evaluation oracle access: given S obtain $F(S)$. *(**time** = #queries to the oracle)*

Submodular functions can be minimized in polynomial time

(Grotschel, Lovasz, Shrijver 1981)

Exact Minimization: find $S^\star$ such that $F(S^\star) = \min_{S \subset [n]} F(S)$

- **Lee, Sidford, Wong 2015:** $\tilde{O}(n^3)$ or $\tilde{O}(n^2 \log M)$ where $M = \max |F(S)|$

  (lower bound: $\Omega(n)$)

Evaluation oracle access: given S obtain $F(S)$. *(**time** = #queries to the oracle)*

Submodular functions can be minimized in polynomial time

(Grotschel, Lovasz, Shrijver 1981)

Exact Minimization: find $S^\star$ such that $F(S^\star) = \min_{S \subset [n]} F(S)$

- **Lee, Sidford, Wong 2015:** $\tilde{O}(n^3)$ or $\tilde{O}(n^2 \log M)$ where $M = \max |F(S)|$

  (lower bound: $\Omega(n)$)

ε-Approx. Minimization: find $S^\star$ such that $F(S^\star) \leq \min_{S \subset [n]} F(S) + \epsilon$

Submodular Function Minimization

Evaluation oracle access: given S obtain $F(S)$. *(**time** = #queries to the oracle)*

Submodular functions can be minimized in polynomial time
(Grotschel, Lovasz, Shrijver 1981)

Exact Minimization: find $S^\star$ such that $F(S^\star) = \min_{S \subset [n]} F(S)$

- **Lee, Sidford, Wong 2015:** $\tilde{O}(n^3)$ or $\tilde{O}(n^2 \log M)$ where $M = \max |F(S)|$

  (lower bound: $\Omega(n)$)

ε-Approx. Minimization: find $S^\star$ such that $F(S^\star) \leq \min_{S \subset [n]} F(S) + \epsilon$

- **Chakrabarty, Lee, Sidford, Wong 2017:** $\tilde{O}(n^{5/3}/\epsilon^2)$

Submodular Function Minimization

Evaluation oracle access: given S obtain $F(S)$. *(**time** = #queries to the oracle)*

Submodular functions can be minimized in polynomial time
(Grotschel, Lovasz, Shrijver 1981)

Exact Minimization: find $S^\star$ such that $F(S^\star) = \min_{S \subset [n]} F(S)$

- **Lee, Sidford, Wong 2015:** $\tilde{O}(n^3)$ or $\tilde{O}(n^2 \log M)$ where $M = \max |F(S)|$

  (lower bound: $\Omega(n)$)

ε-Approx. Minimization: find $S^\star$ such that $F(S^\star) \leq \min_{S \subset [n]} F(S) + \epsilon$

- **Chakrabarty, Lee, Sidford, Wong 2017:** $\tilde{O}(n^{5/3}/\epsilon^2)$
- **Our result:** $\tilde{O}(n^{3/2}/\epsilon^2)$ (classical) or $\tilde{O}(n^{5/4}/\epsilon^{5/2})$ (quantum)

Submodular Function Minimization

Evaluation oracle access: given S obtain $F(S)$. *(**time** = #queries to the oracle)*

<u>Submodular functions can be minimized in polynomial time</u>
(Grotschel, Lovasz, Shrijver 1981)

Exact Minimization: find $S^\star$ such that $F(S^\star) = \min_{S \subset [n]} F(S)$

- **Lee, Sidford, Wong 2015:** $\tilde{O}(n^3)$ or $\tilde{O}(n^2 \log M)$ where $M = \max |F(S)|$

  (lower bound: $\Omega(n)$)

ε-Approx. Minimization: find $S^\star$ such that $F(S^\star) \leq \min_{S \subset [n]} F(S) + \epsilon$

- **Chakrabarty, Lee, Sidford, Wong 2017:** $\tilde{O}(n^{5/3}/\epsilon^2)$
- **Our result:** $\tilde{O}(n^{3/2}/\epsilon^2)$ (classical) or $\tilde{O}(n^{5/4}/\epsilon^{5/2})$ (quantum)
- **Axelrod, Liu, Sidford 2019:** $\tilde{O}(n/\epsilon^2)$ (classical)

Discrete Optimization

Set function: $F : 2^{[n]} \to \mathbb{R}$

Discrete Optimization

Set function: $F : 2^{[n]} \to \mathbb{R}$

$\downarrow$

Continuous Optimization

Lovász extension: $f : [0,1]^n \to \mathbb{R}$

Lovász Extension

Discrete Optimization

Set function: $F : 2^{[n]} \to \mathbb{R}$

$$\downarrow$$

Continuous Optimization

Lovász extension: $f : [0,1]^n \to \mathbb{R}$

$n = 2$

$F(\emptyset) = 0$

$F(\{1\}) = 10$

$F(\{2\}) = 6$

$F(\{1,2\}) = 3$

Lovász Extension

Discrete Optimization

Set function: $F : 2^{[n]} \to \mathbb{R}$

Continuous Optimization

Lovász extension: $f : [0,1]^n \to \mathbb{R}$

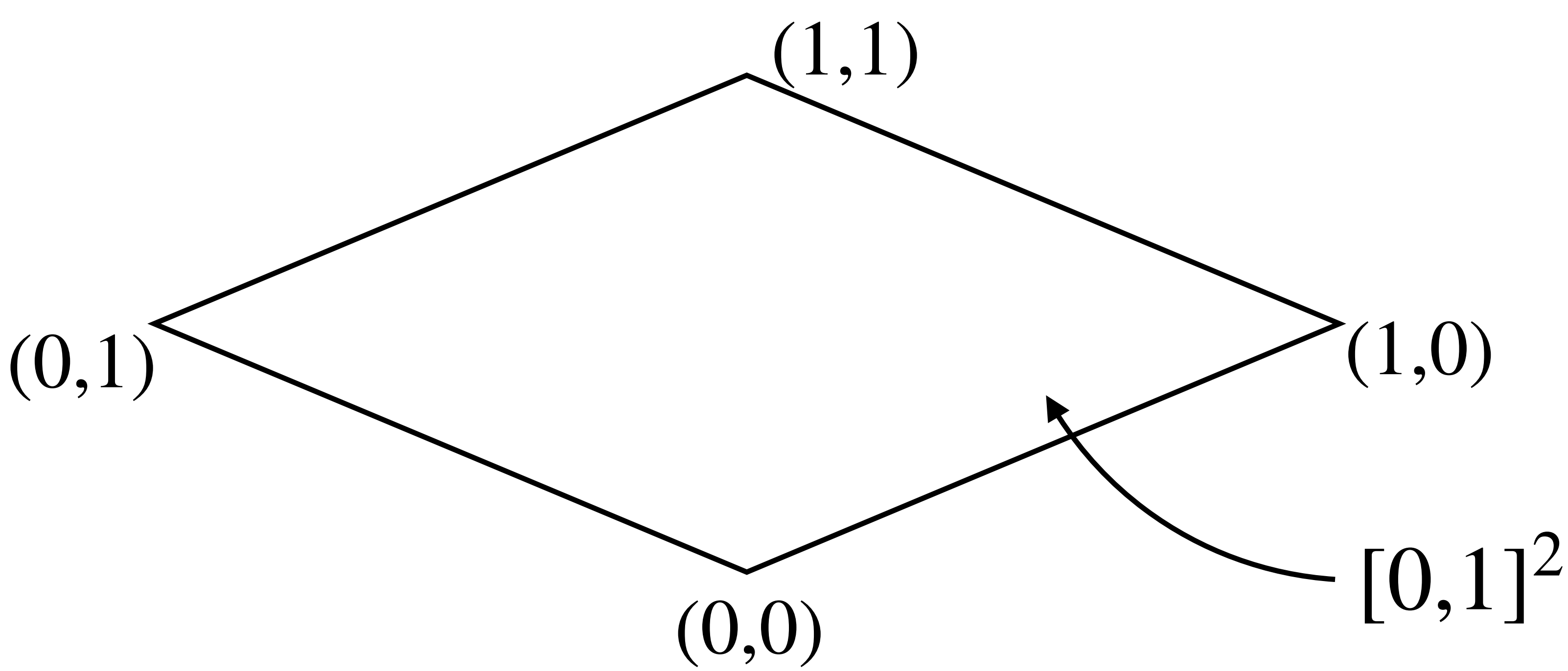
$$n = 2$$

$$F(\emptyset) = 0$$

$$F(\{1\}) = 10$$

$$F(\{2\}) = 6$$

$$F(\{1,2\}) = 3$$



Lovász Extension

Discrete Optimization

Set function: $F : 2^{[n]} \to \mathbb{R}$

Continuous Optimization

Lovász extension: $f : [0,1]^n \to \mathbb{R}$

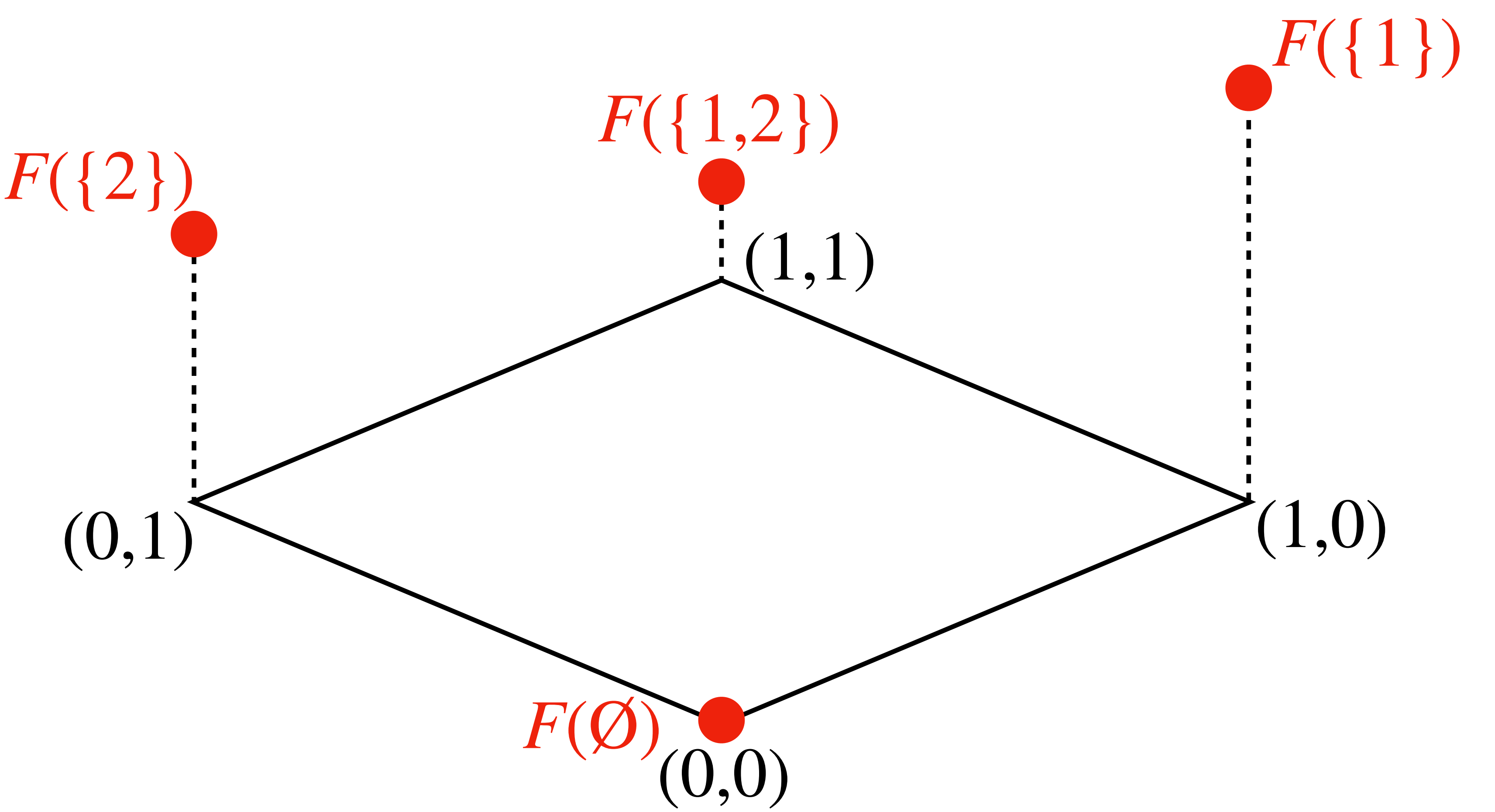
$n = 2$

$F(\emptyset) = 0$

$F(\{1\}) = 10$

$F(\{2\}) = 6$

$F(\{1,2\}) = 3$



Lovász Extension

Discrete Optimization

Set function: $F : 2^{[n]} \to \mathbb{R}$

Continuous Optimization

Lovász extension: $f : [0,1]^n \to \mathbb{R}$

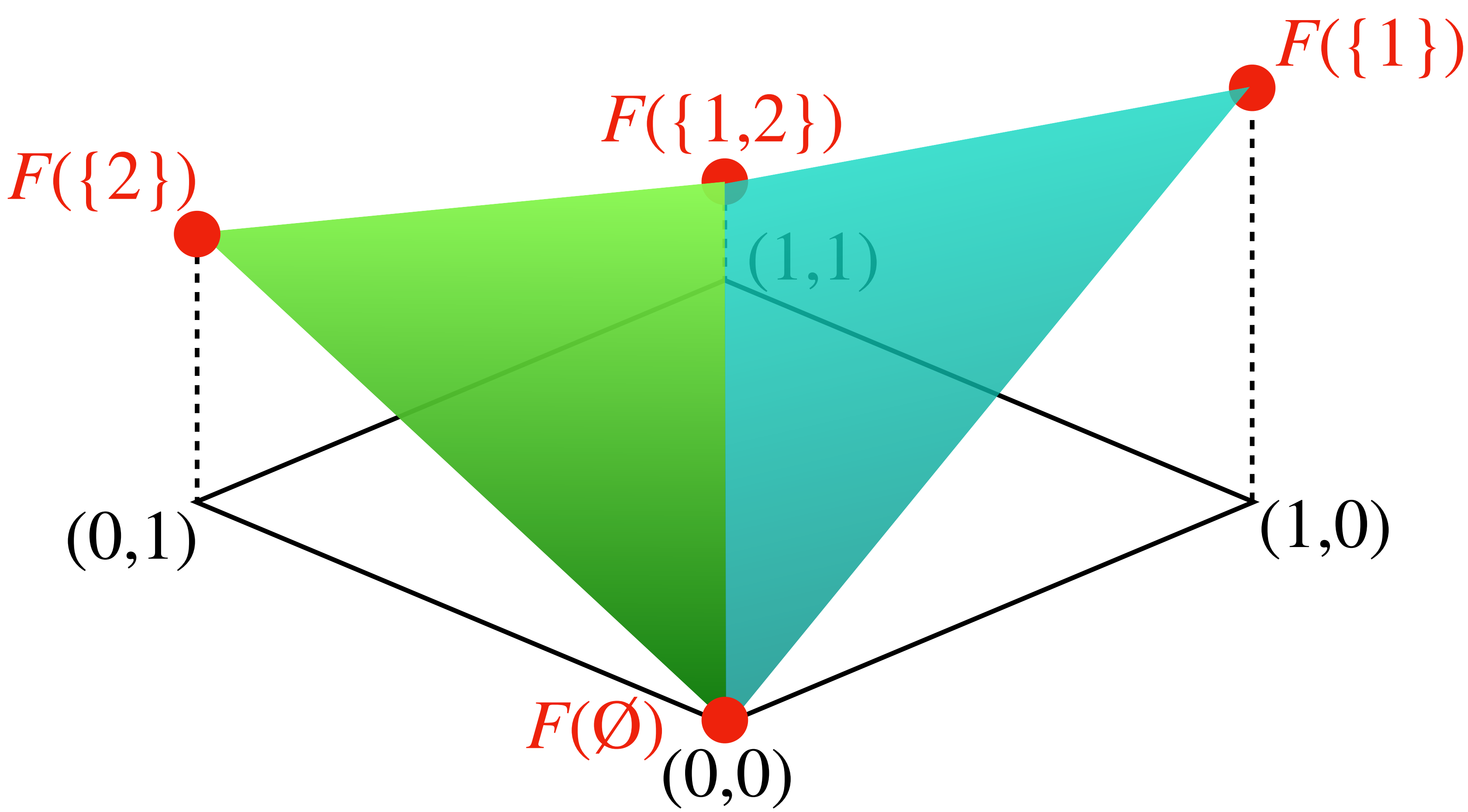
$n = 2$

$F(\emptyset) = 0$

$F(\{1\}) = 10$

$F(\{2\}) = 6$

$F(\{1,2\}) = 3$



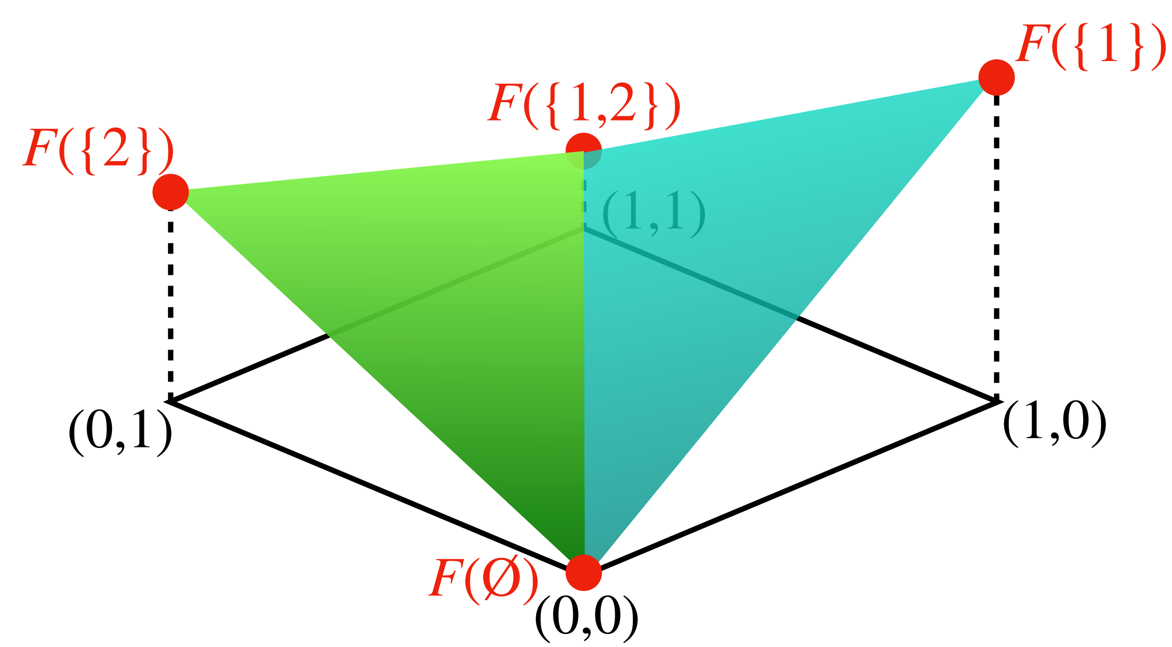
Discrete Optimization

Set function: $F : 2^{[n]} \to \mathbb{R}$

Continuous Optimization

Lovász extension: $f : [0,1]^n \to \mathbb{R}$

The Lovász extension is:



Lovász Extension

Discrete Optimization

Set function: $F : 2^{[n]} \to \mathbb{R}$

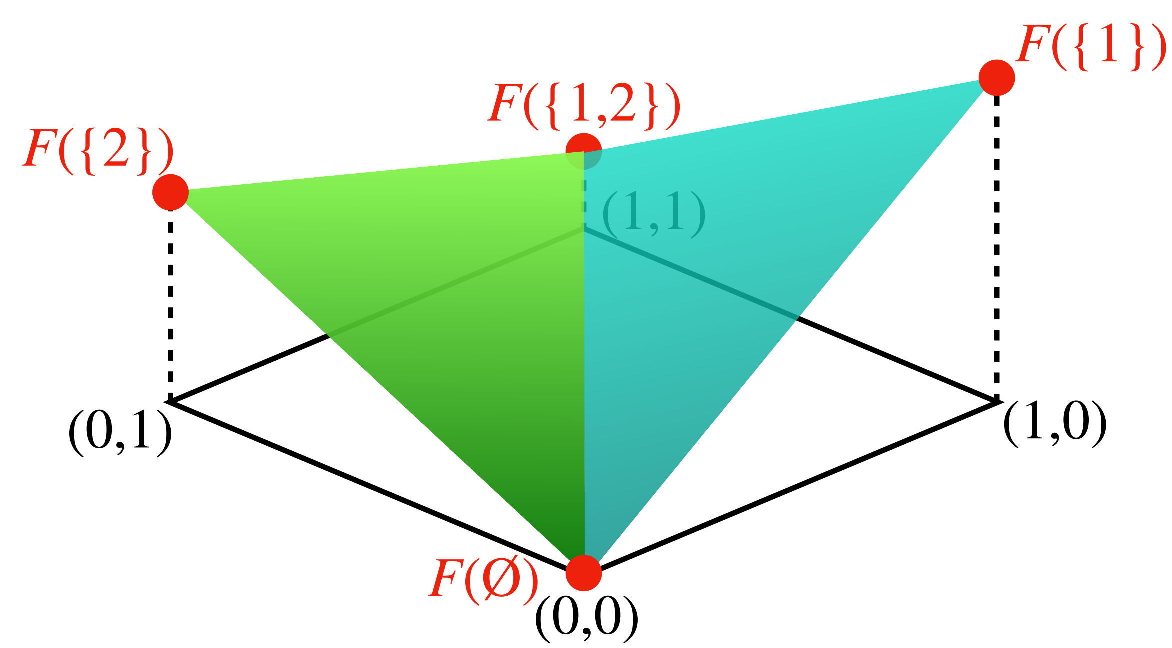
$\downarrow$

Continuous Optimization

Lovász extension: $f : [0,1]^n \to \mathbb{R}$

The Lovász extension is:

- Piecewise linear



Discrete Optimization

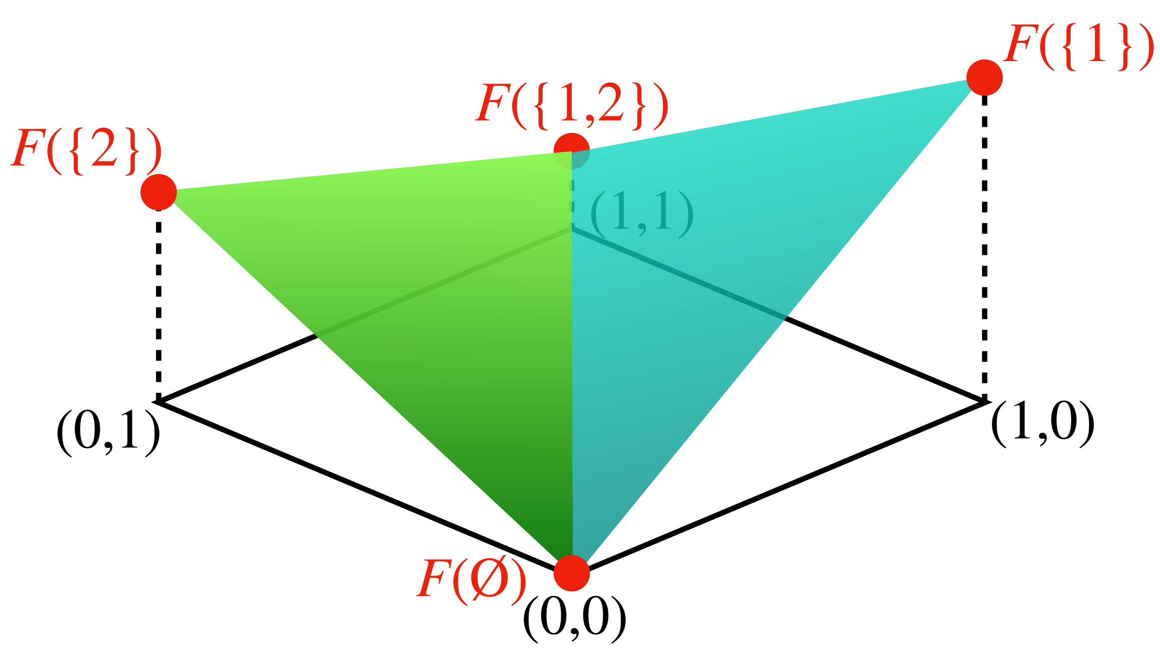
Set function: $F : 2^{[n]} \to \mathbb{R}$

Continuous Optimization

Lovász extension: $f : [0,1]^n \to \mathbb{R}$

The Lovász extension is:

- Piecewise linear
- **Convex** iff F is submodular (Lovász 1983)



Discrete Optimization

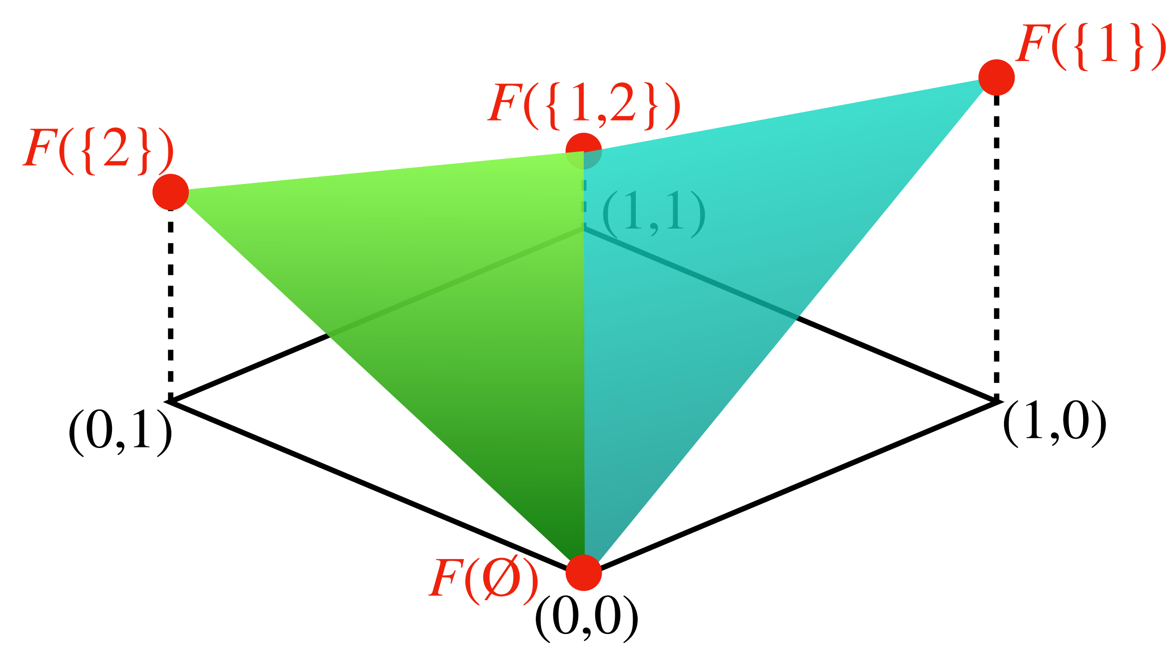
Set function: $F : 2^{[n]} \to \mathbb{R}$

Continuous Optimization

Lovász extension: $f : [0,1]^n \to \mathbb{R}$

The Lovász extension is:

- Piecewise linear
- **Convex** iff F is submodular (Lovász 1983)
- Evaluable using n queries to F.



Exact Minimization:

- **Lee, Sidford, Wong 2015:**

Cutting plane method
on the Lovász extension

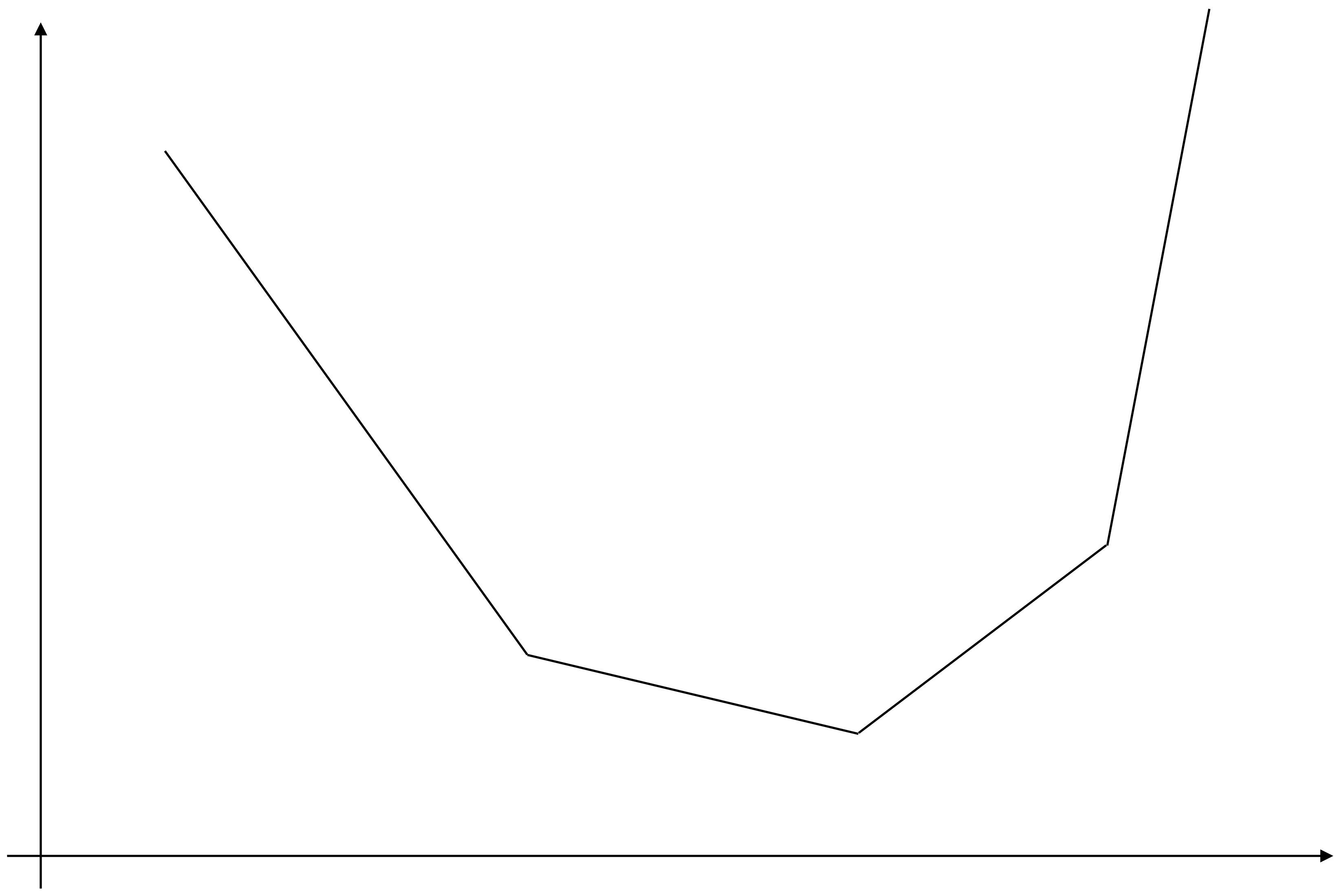
ε-Approx. Minimization:

- **Chakrabarty, Lee, Sidford, Wong 2017:**
- **Our result:**
- **Axelrod, Liu, Sidford 2019:**

Stochastic Subgradient Descent
on the Lovász extension

Stochastic Subgradient Descent

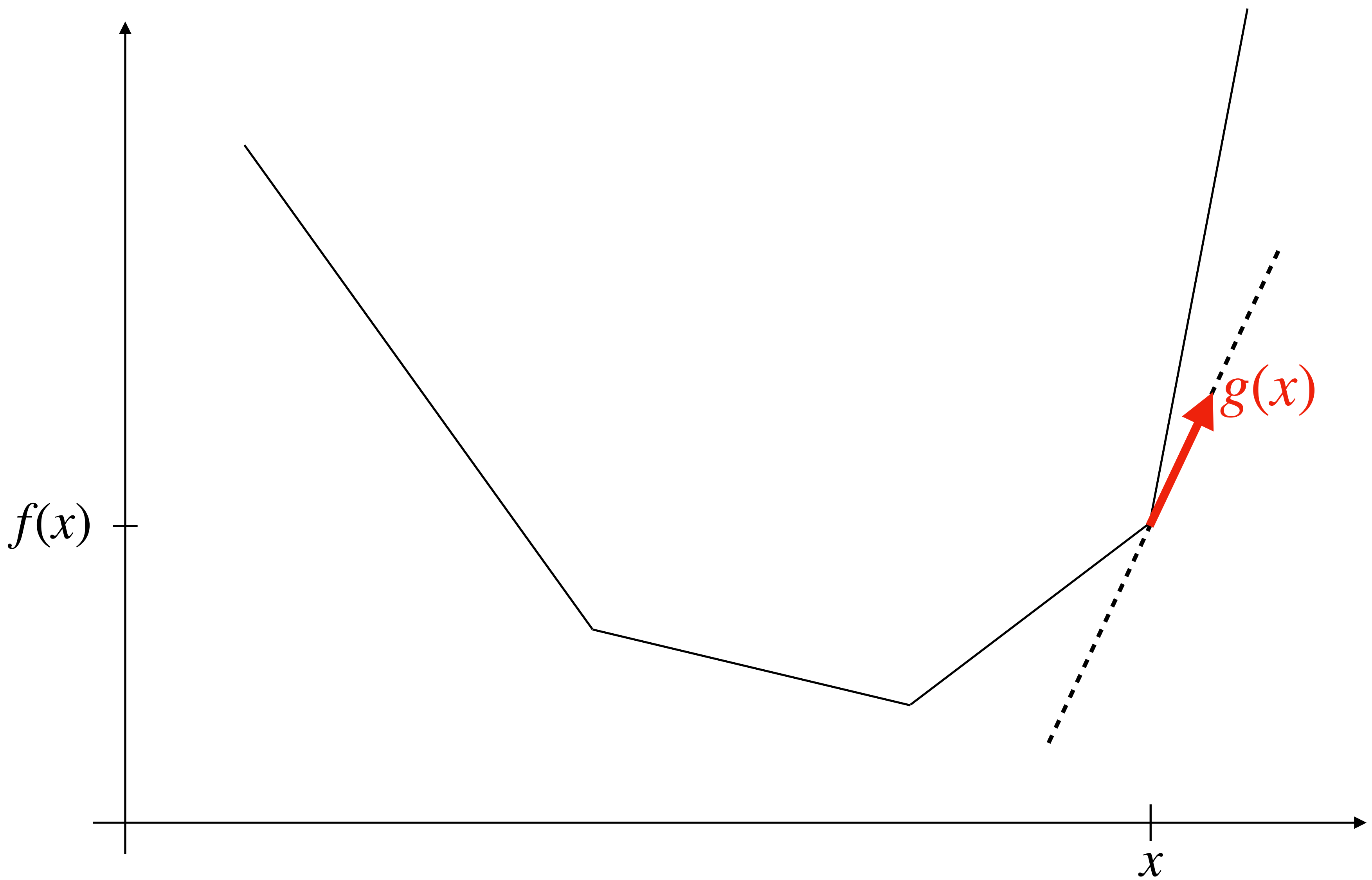
Convex function $f : C \to \mathbb{R}$ on a convex set C. *(not necessarily differentiable)*



Stochastic Subgradient Descent

Convex function $f : C \to \mathbb{R}$ on a convex set C. *(not necessarily differentiable)*

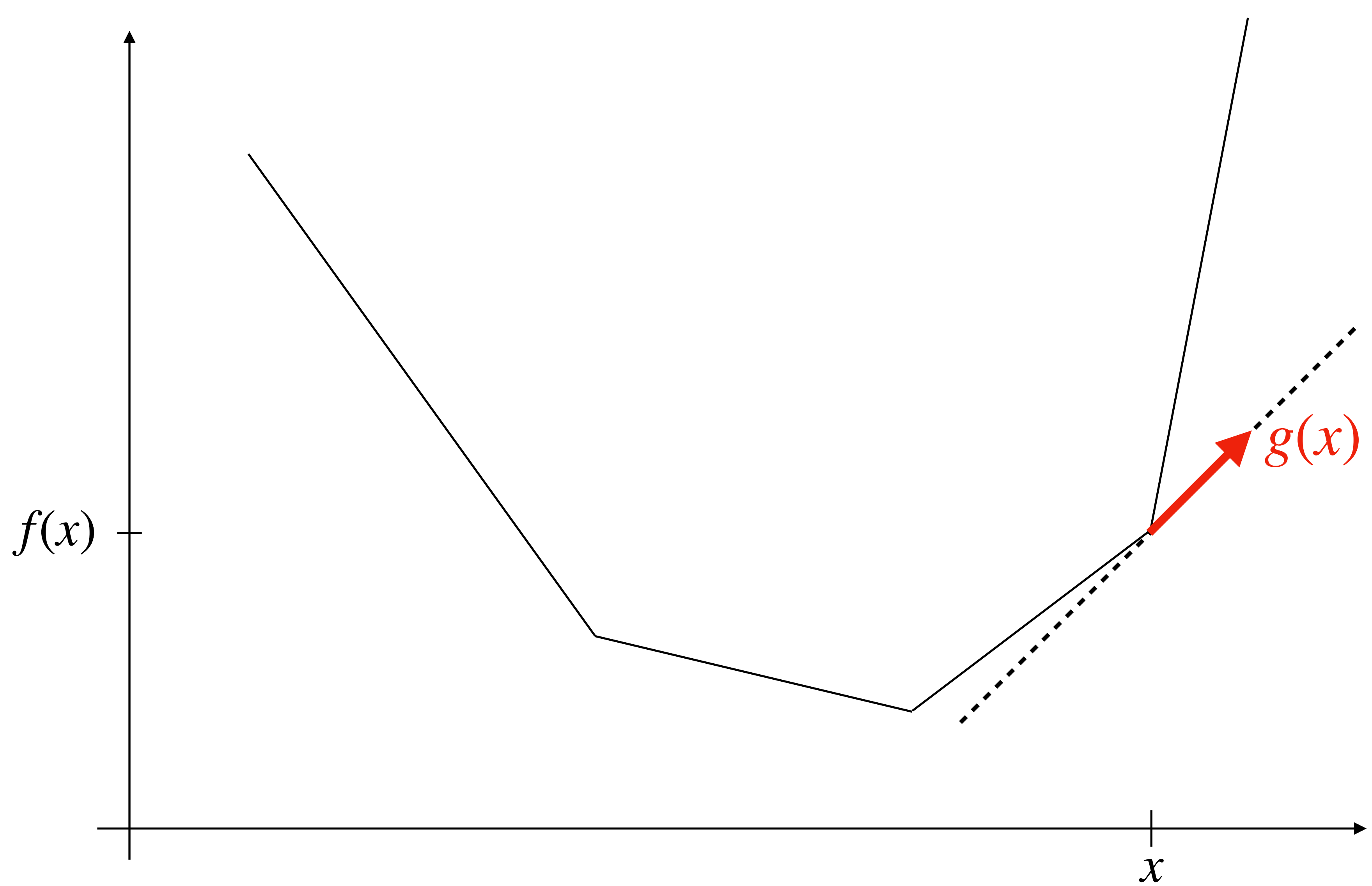
Subgradient at x: slope $g(x)$ of any line that is below the graph of f and intersects it at x.



Stochastic Subgradient Descent

Convex function $f : C \to \mathbb{R}$ on a convex set C. *(not necessarily differentiable)*

Subgradient at x: slope $g(x)$ of any line that is below the graph of f and intersects it at x.

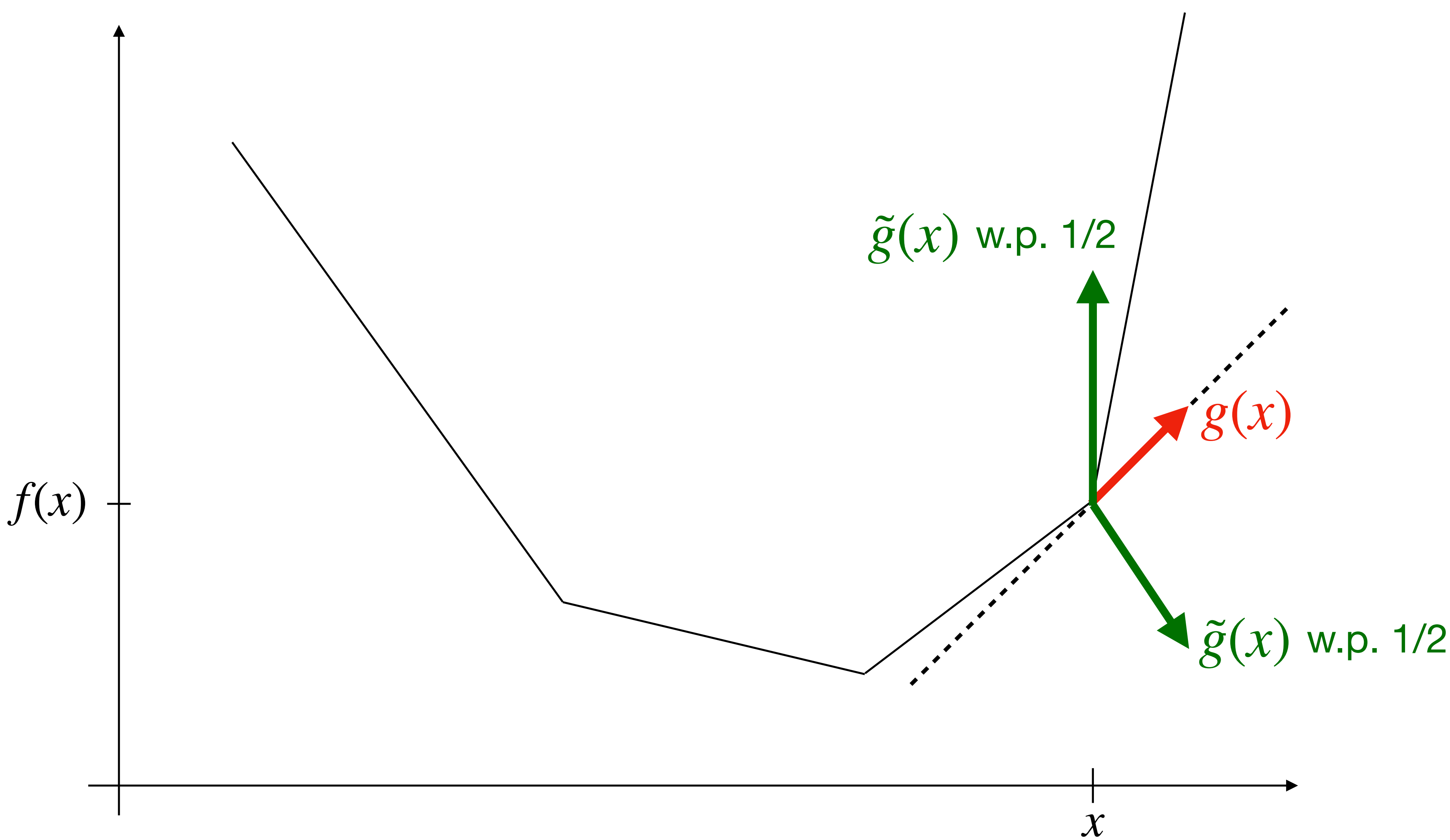


Stochastic Subgradient Descent

Convex function $f : C \to \mathbb{R}$ on a convex set C. *(not necessarily differentiable)*

Subgradient at x: slope $g(x)$ of any line that is below the graph of f and intersects it at x.

Stochastic Subgradient at x: random variable $\tilde{g}(x)$ satisfying $E[\tilde{g}(x)] = g(x)$



Stochastic Subgradient Descent

Convex function $f : C \to \mathbb{R}$ on a convex set C. *(not necessarily differentiable)*

Subgradient at x: slope $g(x)$ of any line that is below the graph of f and intersects it at x.

Stochastic Subgradient at x: random variable $\tilde{g}(x)$ satisfying $E[\tilde{g}(x)] = g(x)$

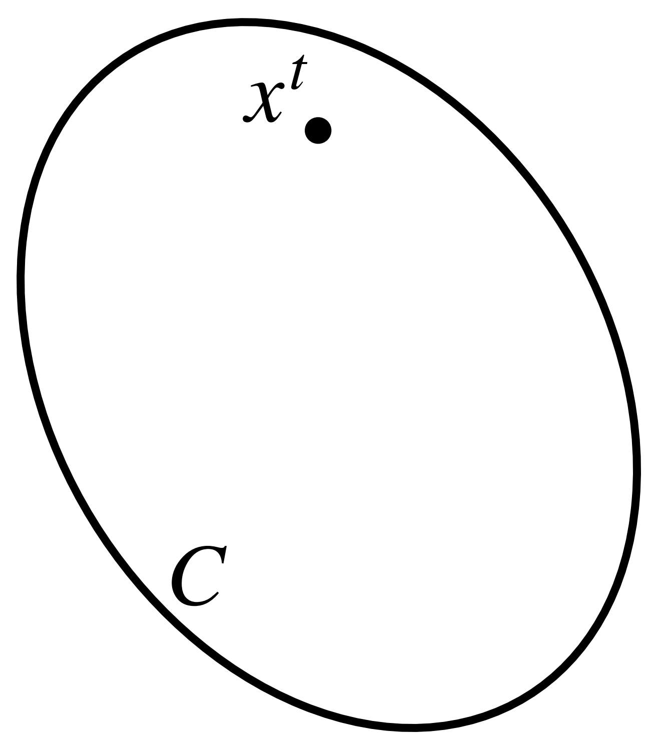
(projected) **Stochastic Subgradient Descent**

Convex function $f : C \to \mathbb{R}$ on a convex set C. *(not necessarily differentiable)*

Subgradient at x: slope $g(x)$ of any line that is below the graph of f and intersects it at x.

Stochastic Subgradient at x: random variable $\tilde{g}(x)$ satisfying $E[\tilde{g}(x)] = g(x)$

(projected) **Stochastic Subgradient Descent**



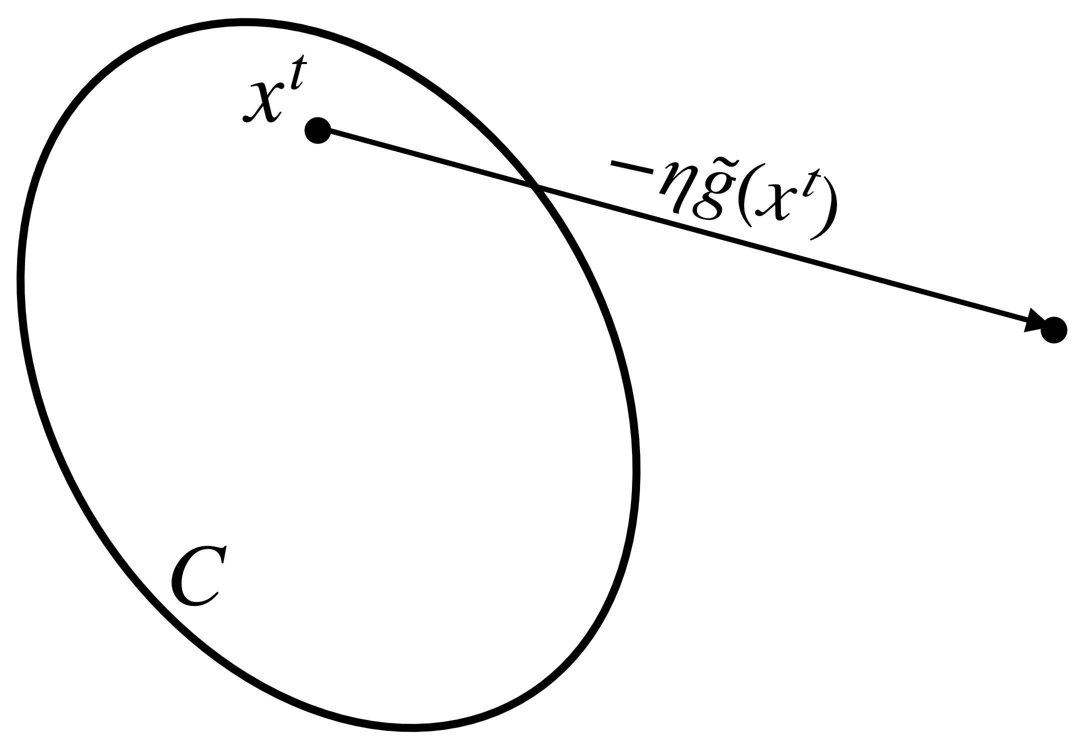
Stochastic Subgradient Descent

Convex function $f : C \to \mathbb{R}$ on a convex set C. *(not necessarily differentiable)*

Subgradient at x: slope $g(x)$ of any line that is below the graph of f and intersects it at x.

Stochastic Subgradient at x: random variable $\tilde{g}(x)$ satisfying $E[\tilde{g}(x)] = g(x)$

(projected) **Stochastic Subgradient Descent**



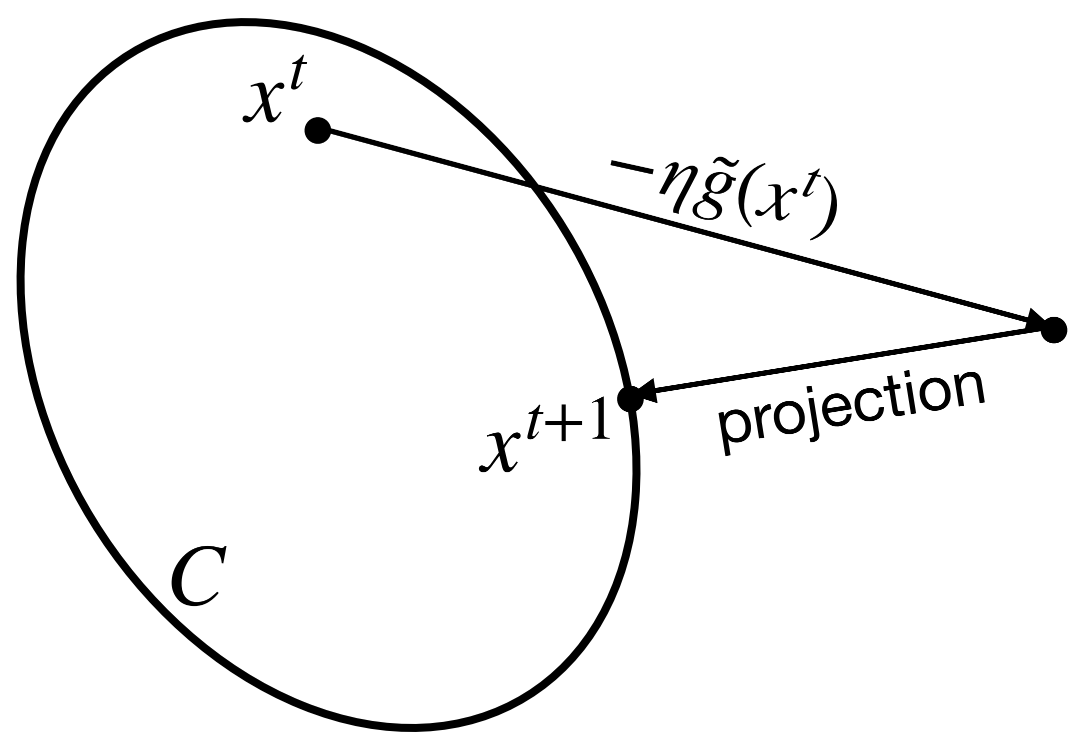
Stochastic Subgradient Descent

Convex function $f : C \to \mathbb{R}$ on a convex set C. *(not necessarily differentiable)*

Subgradient at x: slope $g(x)$ of any line that is below the graph of f and intersects it at x.

Stochastic Subgradient at x: random variable $\tilde{g}(x)$ satisfying $E[\tilde{g}(x)] = g(x)$

(projected) **Stochastic Subgradient Descent**



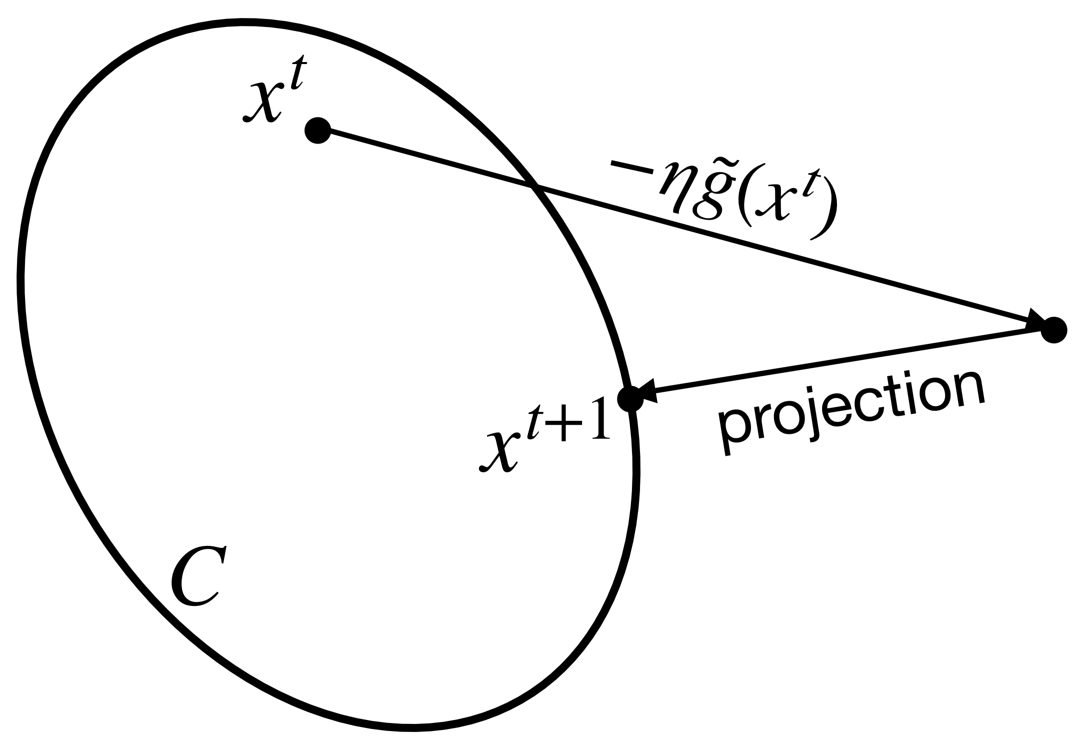
Stochastic Subgradient Descent

Convex function $f : C \to \mathbb{R}$ on a convex set C. *(not necessarily differentiable)*

Subgradient at x: slope $g(x)$ of any line that is below the graph of f and intersects it at x.

Stochastic Subgradient at x: random variable $\tilde{g}(x)$ satisfying $E[\tilde{g}(x)] = g(x)$

(projected) **Stochastic Subgradient Descent**



If $\tilde{g}(x)$ has **low variance** then the number of steps is the same as if we were using $g(x)$.

Stochastic Subgradient for the Lovász extension

For the Lovász extension f, there exists a subgradient $g(x)$ such that:

Stochastic Subgradient for the Lovász extension

For the Lovász extension f, there exists a subgradient $g(x)$ such that:

- each coordinate $g(x)_i$ can be computed with two queries to F

Stochastic Subgradient for the Lovász extension

For the Lovász extension f, there exists a subgradient $g(x)$ such that:

- each coordinate $g(x)_i$ can be computed with two queries to F
- subgradient descent requires $O(n/\epsilon^2)$ steps to get an ε-minimizer of f

(Jegelka, Bilmes 2011) and (Hazan, Kale 2012)

Stochastic Subgradient for the Lovász extension

For the Lovász extension f, there exists a subgradient $g(x)$ such that:

- each coordinate $g(x)_i$ can be computed with two queries to F
- subgradient descent requires $O(n/\epsilon^2)$ steps to get an ε-minimizer of f

(Jegelka, Bilmes 2011) and (Hazan, Kale 2012)

A **stochastic subgradient** $\tilde{g}(x)$ for $g(x)$ can be computed in time:

Stochastic Subgradient for the Lovász extension

For the Lovász extension f, there exists a subgradient $g(x)$ such that:

- each coordinate $g(x)_i$ can be computed with two queries to F
- subgradient descent requires $O(n/\epsilon^2)$ steps to get an ε-minimizer of f

(Jegelka, Bilmes 2011) and (Hazan, Kale 2012)

A **stochastic subgradient** $\tilde{g}(x)$ for $g(x)$ can be computed in time:

- **Chakrabarty, Lee, Sidford, Wong 2017:** $\tilde{O}(n^{2/3})$

Stochastic Subgradient for the Lovász extension

For the Lovász extension f, there exists a subgradient $g(x)$ such that:

- each coordinate $g(x)_i$ can be computed with two queries to F
- subgradient descent requires $O(n/\epsilon^2)$ steps to get an ε-minimizer of f

(Jegelka, Bilmes 2011) and (Hazan, Kale 2012)

A **stochastic subgradient** $\tilde{g}(x)$ for $g(x)$ can be computed in time:

- **Chakrabarty, Lee, Sidford, Wong 2017:** $\tilde{O}(n^{2/3})$
- **Our result:** $\tilde{O}(n^{1/2})$ (classical) or $\tilde{O}(n^{1/4}/\epsilon^{1/2})$ (quantum)

Stochastic Subgradient for the Lovász extension

For the Lovász extension *f*, there exists a subgradient $g(x)$ such that:

- each coordinate $g(x)_i$ can be computed with two queries to F
- subgradient descent requires $O(n/\epsilon^2)$ steps to get an ε-minimizer of f

(Jegelka, Bilmes 2011) and (Hazan, Kale 2012)

A **stochastic subgradient** $\tilde{g}(x)$ for $g(x)$ can be computed in time:

- **Chakrabarty, Lee, Sidford, Wong 2017:** $\tilde{O}(n^{2/3})$
- **Our result:** $\tilde{O}(n^{1/2})$ (classical) or $\tilde{O}(n^{1/4}/\epsilon^{1/2})$ (quantum)
- **Axelrod, Liu, Sidford 2019:** $\tilde{O}(1)$

Stochastic Subgradient for the Lovász extension

First attempt to construct $\tilde{g}(x)$:

Stochastic Subgradient for the Lovász extension

First attempt to construct $\tilde{g}(x)$:

For any non-zero vector $u \in R^n$, define the random variable

Stochastic Subgradient for the Lovász extension

First attempt to construct $\tilde{g}(x)$:

For any non-zero vector $u \in R^n$, define the random variable

i-th coordinate

$$\hat{u} = (0, \ \dots \ , 0 \ , \operatorname{sgn}(u_i)\|u\|_1 \ , 0 \ , \ \dots \ , 0)$$

where i is sampled with probability $p_i = \dfrac{|u_i|}{\|u\|_1}$

Stochastic Subgradient for the Lovász extension

First attempt to construct $\tilde{g}(x)$:

For any non-zero vector $u \in R^n$, define the random variable

i-th coordinate

$$\hat{u} = (0, \ \ldots \ , 0 \ , \ \mathrm{sgn}(u_i)\|u\|_1 \ , 0 \ , \ \ldots \ , 0)$$

where i is sampled with probability $p_i = \dfrac{|u_i|}{\|u\|_1}$

Unbiased: $$E[\hat{u}] = \sum_i \frac{|u_i|}{\|u\|_1} \mathrm{sgn}(u_i)\|u\|_1 \cdot \vec{e}_i = \sum_i u_i \cdot \vec{e}_i = u$$

Stochastic Subgradient for the Lovász extension

First attempt to construct $\tilde{g}(x)$:

For any non-zero vector $u \in R^n$, define the random variable

i-th coordinate

$$\hat{u} = (0, \ \ldots \ , 0 \ , \operatorname{sgn}(u_i)\|u\|_1 \ , 0 \ , \ \ldots \ , 0)$$

where i is sampled with probability $p_i = \dfrac{|u_i|}{\|u\|_1}$

Unbiased: $$E[\hat{u}] = \sum_i \frac{|u_i|}{\|u\|_1} \operatorname{sgn}(u_i)\|u\|_1 \cdot \vec{e}_i = \sum_i u_i \cdot \vec{e}_i = u$$

2nd moment: $$E[\|\hat{u}\|_2^2] = \sum_i \frac{|u_i|}{\|u\|_1} \operatorname{sgn}(u_i)^2 \|u\|_1^2 = \|u\|_1^2$$

Stochastic Subgradient for the Lovász extension

First attempt to construct $\tilde{g}(x)$:

For any non-zero vector $u \in R^n$, define the random variable

i-th coordinate

$$\hat{u} = (0, \ \ldots \ , 0 \ , \operatorname{sgn}(u_i)\|u\|_1 \ , 0 \ , \ \ldots \ , 0)$$

where i is sampled with probability $p_i = \dfrac{|u_i|}{\|u\|_1}$

Unbiased: $$E[\hat{u}] = \sum_i \frac{|u_i|}{\|u\|_1} \operatorname{sgn}(u_i)\|u\|_1 \cdot \vec{e}_i = \sum_i u_i \cdot \vec{e}_i = u$$

2nd moment: $$E[\|\hat{u}\|_2^2] = \sum_i \frac{|u_i|}{\|u\|_1} \operatorname{sgn}(u_i)^2 \|u\|_1^2 = \|u\|_1^2$$

For the Lovász extension: $u = g(x)$ and $\|g(x)\|_1 = O(1)$ (low variance) ✓

(Jegelka, Bilmes 2011)

Stochastic Subgradient for the Lovász extension

First attempt to construct $\tilde{g}(x)$:

For any non-zero vector $u \in R^n$, define the random variable

i-th coordinate

$$\hat{u} = (0, \ \ldots \ , 0 \ , \operatorname{sgn}(u_i)\|u\|_1 \ , 0 \ , \ \ldots \ , 0)$$

where i is sampled with probability $p_i = \frac{|u_i|}{\|u\|_1}$ ✗ **Hard to sample** **(Importance sampling)**

Unbiased: $$E[\hat{u}] = \sum_i \frac{|u_i|}{\|u\|_1} \operatorname{sgn}(u_i)\|u\|_1 \cdot \vec{e}_i = \sum_i u_i \cdot \vec{e}_i = u$$

2nd moment: $$E[\|\hat{u}\|_2^2] = \sum_i \frac{|u_i|}{\|u\|_1} \operatorname{sgn}(u_i)^2 \|u\|_1^2 = \|u\|_1^2$$

For the Lovász extension: $u = g(x)$ and $\|g(x)\|_1 = O(1)$ (low variance) ✓

(Jegelka, Bilmes 2011)

Stochastic Subgradient for the Lovász extension

Second attempt:

Stochastic Subgradient for the Lovász extension

Second attempt:

Tool: there is an unbiased estimate $\tilde{d}(x, y)$ of $d(x, y) = g(y) - g(x)$ that can be computed efficiently when $x - y$ is **sparse.**

(Chakrabarty, Lee, Sidford, Wong 2017)

Stochastic Subgradient for the Lovász extension

Second attempt:

Tool: there is an unbiased estimate $\tilde{d}(x, y)$ of $d(x, y) = g(y) - g(x)$ that can be computed efficiently when $x - y$ is **sparse.**

(Chakrabarty, Lee, Sidford, Wong 2017)

Our construction:

Stochastic Subgradient for the Lovász extension

Second attempt:

Tool: there is an unbiased estimate $\tilde{d}(x, y)$ of $d(x, y) = g(y) - g(x)$ that can be computed efficiently when $x - y$ is **sparse.**

(Chakrabarty, Lee, Sidford, Wong 2017)

Our construction:

$$x^0 \longrightarrow \hat{g}(x^0)$$

Stochastic Subgradient for the Lovász extension

Second attempt:

Tool: there is an unbiased estimate $\tilde{d}(x,y)$ of $d(x,y) = g(y) - g(x)$ that can be computed efficiently when $x - y$ is **sparse.**

(Chakrabarty, Lee, Sidford, Wong 2017)

Our construction:

$$x^0 \longrightarrow \hat{g}(x^0)$$

$$x^1 \longrightarrow \hat{g}(x^0) + \tilde{d}(x^0, x^1)$$

Stochastic Subgradient for the Lovász extension

Second attempt:

Tool: there is an unbiased estimate $\tilde{d}(x, y)$ of $d(x, y) = g(y) - g(x)$ that can be computed efficiently when $x - y$ is **sparse.**

(Chakrabarty, Lee, Sidford, Wong 2017)

Our construction:

$$x^0 \longrightarrow \hat{g}(x^0)$$

$$x^1 \longrightarrow \hat{g}(x^0) + \tilde{d}(x^0, x^1)$$

$$x^2 \longrightarrow \hat{g}(x^0) + \tilde{d}(x^0, x^2)$$

Stochastic Subgradient for the Lovász extension

Second attempt:

Tool: there is an unbiased estimate $\tilde{d}(x, y)$ of $d(x, y) = g(y) - g(x)$ that can be computed efficiently when $x - y$ is **sparse.**

(Chakrabarty, Lee, Sidford, Wong 2017)

Our construction: (T = parameter to be optimized)

$$
\begin{array}{rcl}
x^0 & \longrightarrow & \hat{g}(x^0) \\
x^1 & \longrightarrow & \hat{g}(x^0) + \tilde{d}(x^0, x^1) \\
x^2 & \longrightarrow & \hat{g}(x^0) + \tilde{d}(x^0, x^2) \\
 & \vdots & \\
x^{T-1} & \longrightarrow & \hat{g}(x^0) + \tilde{d}(x^0, x^{T-1})
\end{array}
$$

Stochastic Subgradient for the Lovász extension

Second attempt:

Tool: there is an unbiased estimate $\tilde{d}(x, y)$ of $d(x, y) = g(y) - g(x)$ that can be computed efficiently when $x - y$ is **sparse.**

(Chakrabarty, Lee, Sidford, Wong 2017)

Our construction: (T = parameter to be optimized)

$$x^0 \longrightarrow \hat{g}(x^0)$$

$$x^1 \longrightarrow \hat{g}(x^0) + \tilde{d}(x^0, x^1)$$

$$x^2 \longrightarrow \hat{g}(x^0) + \tilde{d}(x^0, x^2)$$

$$\vdots$$

$$x^{T-1} \longrightarrow \hat{g}(x^0) + \tilde{d}(x^0, x^{T-1})$$

$$x^T \longrightarrow \hat{g}(x^T)$$

Stochastic Subgradient for the Lovász extension

Second attempt:

Tool: there is an unbiased estimate $\tilde{d}(x, y)$ of $d(x, y) = g(y) - g(x)$ that can be computed efficiently when $x - y$ is **sparse.**

(Chakrabarty, Lee, Sidford, Wong 2017)

Our construction: (T = parameter to be optimized)

$$x^0 \longrightarrow \hat{g}(x^0)$$

$$x^1 \longrightarrow \hat{g}(x^0) + \tilde{d}(x^0, x^1)$$

$$x^2 \longrightarrow \hat{g}(x^0) + \tilde{d}(x^0, x^2)$$

$$\vdots$$

$$x^{T-1} \longrightarrow \hat{g}(x^0) + \tilde{d}(x^0, x^{T-1})$$

$$x^T \longrightarrow \hat{g}(x^T)$$

$$x^{T+1} \longrightarrow \hat{g}(x^T) + \tilde{d}(x^T, x^{T+1})$$

Stochastic Subgradient for the Lovász extension

Second attempt:

Tool: there is an unbiased estimate $\tilde{d}(x, y)$ of $d(x, y) = g(y) - g(x)$ that can be computed efficiently when $x - y$ is **sparse.**

(Chakrabarty, Lee, Sidford, Wong 2017)

Our construction: (T = parameter to be optimized)

$$
\begin{array}{lll}
x^0 & \longrightarrow & \hat{g}(x^0) \\
x^1 & \longrightarrow & \hat{g}(x^0) + \tilde{d}(x^0, x^1) \\
x^2 & \longrightarrow & \hat{g}(x^0) + \tilde{d}(x^0, x^2) \\
& \vdots & \\
x^{T-1} & \longrightarrow & \hat{g}(x^0) + \tilde{d}(x^0, x^{T-1})
\end{array}
\qquad
\begin{array}{lll}
x^T & \longrightarrow & \hat{g}(x^T) \\
x^{T+1} & \longrightarrow & \hat{g}(x^T) + \tilde{d}(x^T, x^{T+1}) \\
x^{T+2} & \longrightarrow & \hat{g}(x^T) + \tilde{d}(x^T, x^{T+2}) \\
& \vdots & \\
x^{2T-1} & \longrightarrow & \hat{g}(x^T) + \tilde{d}(x^T, x^{2T-1})
\end{array}
$$

Stochastic Subgradient for the Lovász extension

Second attempt:

Tool: there is an unbiased estimate $\tilde{d}(x, y)$ of $d(x, y) = g(y) - g(x)$ that can be computed efficiently when $x - y$ is **sparse.**

(Chakrabarty, Lee, Sidford, Wong 2017)

Our construction: (T = parameter to be optimized)

$$
\begin{array}{lll}
x^0 \longrightarrow \hat{g}(x^0) & x^T \longrightarrow \hat{g}(x^T) & x^{2T} \longrightarrow \hat{g}(x^{2T}) \\
x^1 \longrightarrow \hat{g}(x^0) + \tilde{d}(x^0, x^1) & x^{T+1} \longrightarrow \hat{g}(x^T) + \tilde{d}(x^T, x^{T+1}) & \\
x^2 \longrightarrow \hat{g}(x^0) + \tilde{d}(x^0, x^2) & x^{T+2} \longrightarrow \hat{g}(x^T) + \tilde{d}(x^T, x^{T+2}) & \cdots \\
\vdots & \vdots & \\
x^{T-1} \longrightarrow \hat{g}(x^0) + \tilde{d}(x^0, x^{T-1}) & x^{2T-1} \longrightarrow \hat{g}(x^T) + \tilde{d}(x^T, x^{2T-1}) &
\end{array}
$$

Stochastic Subgradient for the Lovász extension

Second attempt:

Tool: there is an unbiased estimate $\tilde{d}(x, y)$ of $d(x, y) = g(y) - g(x)$ that can be computed efficiently when $x - y$ is **sparse.**

(Chakrabarty, Lee, Sidford, Wong 2017)

Our construction: (T = parameter to be optimized)

$$
\begin{array}{lcl}
x^0 & \longrightarrow & \hat{g}(x^0) \\
x^1 & \longrightarrow & \hat{g}(x^0) + \tilde{d}(x^0, x^1) \\
x^2 & \longrightarrow & \hat{g}(x^0) + \tilde{d}(x^0, x^2) \\
 & \vdots & \\
x^{T-1} & \longrightarrow & \hat{g}(x^0) + \tilde{d}(x^0, x^{T-1})
\end{array}
\qquad
\begin{array}{lcl}
x^T & \longrightarrow & \hat{g}(x^T) \\
x^{T+1} & \longrightarrow & \hat{g}(x^T) + \tilde{d}(x^T, x^{T+1}) \\
x^{T+2} & \longrightarrow & \hat{g}(x^T) + \tilde{d}(x^T, x^{T+2}) \\
 & \vdots & \\
x^{2T-1} & \longrightarrow & \hat{g}(x^T) + \tilde{d}(x^T, x^{2T-1})
\end{array}
\qquad
\begin{array}{lcl}
x^{2T} & \longrightarrow & \hat{g}(x^{2T}) \\
 & \ldots & \\
\end{array}
$$

T independent samples

T independent samples

...

2

Quantum speed-up for Importance Sampling

Input: discrete probability distribution **D = (p_1,…,p_n)** on [n].

Output: **T** independent samples $i_1,\ldots,i_T \sim D$.

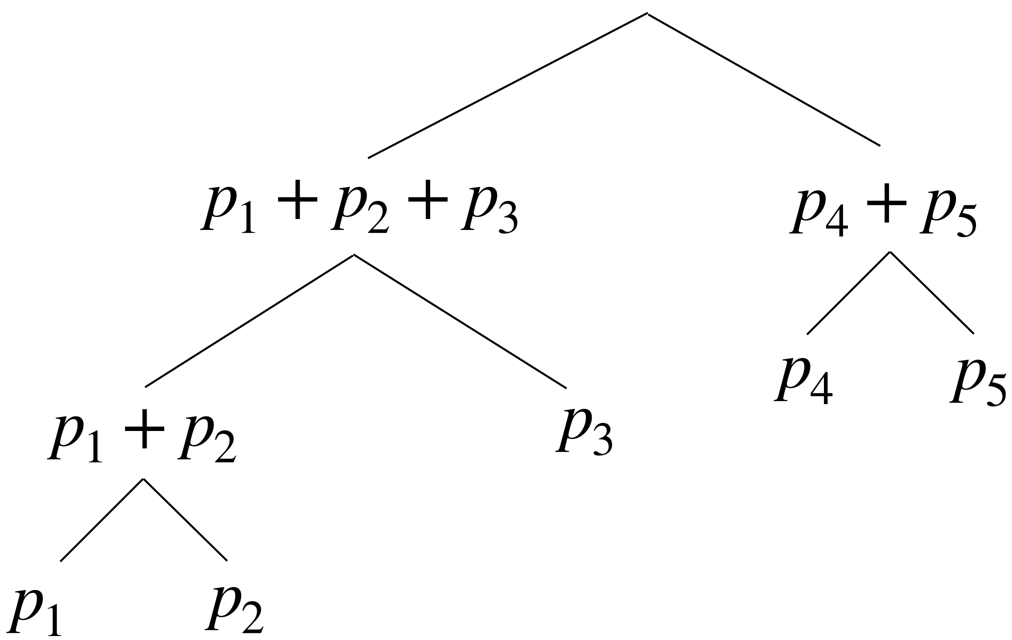
Evaluation oracle access

Classical	Quantum
$i \mapsto p_i$	$U(\lvert i\rangle \lvert 0\rangle) = \lvert i\rangle \lvert p_i\rangle$

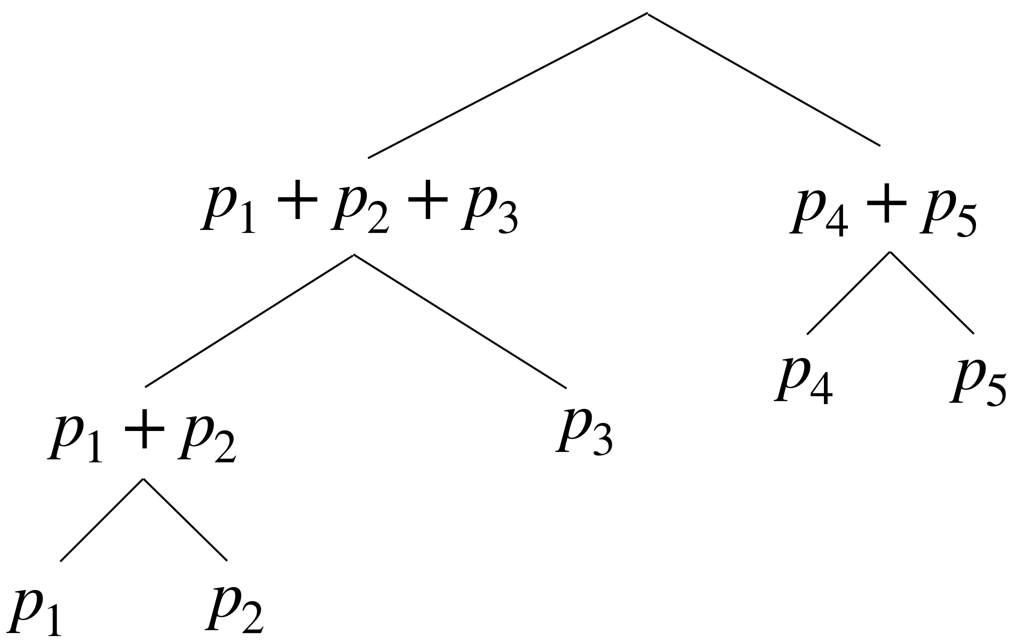
Cost = # queries to the evaluation oracle

Binary Tree

Binary Tree



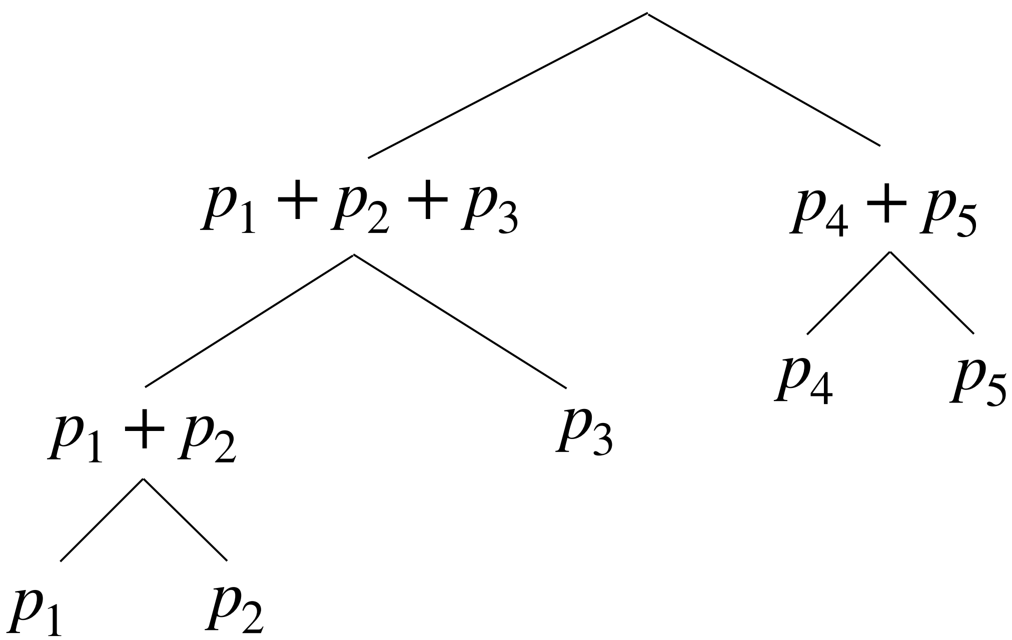
Binary Tree



Preprocessing time: $O(n)$

Cost per sample: $O(\log n)$

Binary Tree

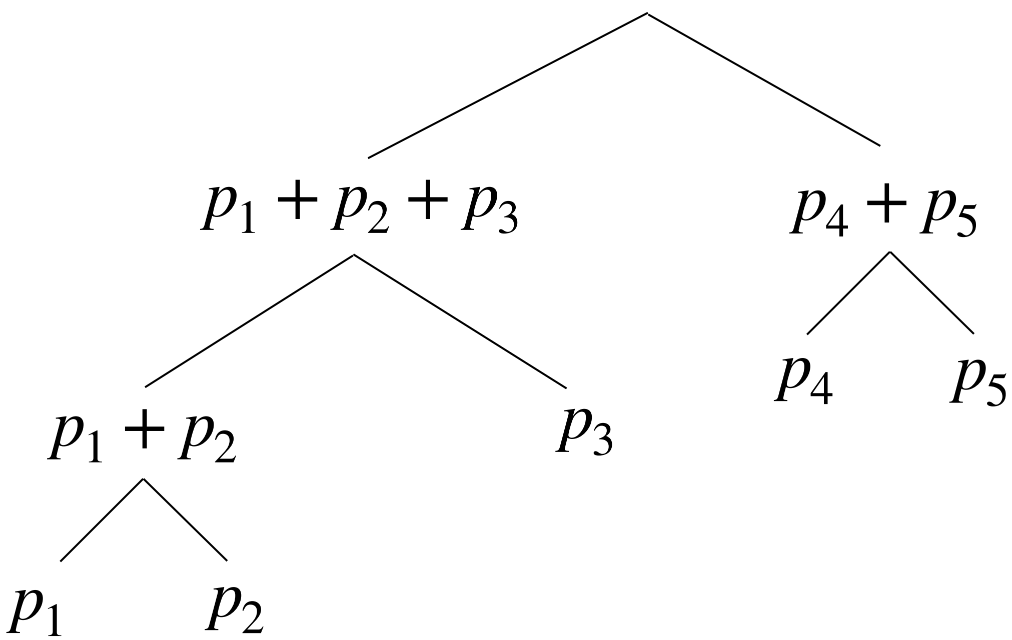


Preprocessing time: $O(n)$

Cost per sample: $O(\log n)$

Cost for T samples: $O(n + T \log n)$

Binary Tree



Preprocessing time: $O(n)$

Cost per sample: $O(\log n)$

Cost for T samples: $O(n + T \log n)$

Alias Method

(Walker 1974, Vose 1991)

Preprocessing time: $O(n)$

Cost per sample: $O(1)$

Cost for T samples: $O(n + T)$

Importance Sampling with Quantum State preparation

(Grover 2000)

Preprocessing:

Sampling **(repeat T times):**

Importance Sampling with Quantum State preparation

(Grover 2000)

Preprocessing:

1. Compute $p_{\max} = \max\{p_1, \ldots, p_n\}$ with quantum **Maximum Finding**

Sampling **(repeat T times)**:

Importance Sampling with Quantum State preparation

(Grover 2000)

Preprocessing:

1. Compute $p_{\max} = \max\{p_1, \ldots, p_n\}$ with quantum **Maximum Finding**

2. Construct the unitary $V(|0\rangle|0\rangle) \longmapsto \frac{1}{\sqrt{n}} \sum_{i \in [n]} |i\rangle|0\rangle$

Sampling **(repeat T times):**

Importance Sampling with Quantum State preparation

(Grover 2000)

Preprocessing:

1. Compute $p_{\max} = \max\{p_1, \ldots, p_n\}$ with quantum **Maximum Finding**

2. Construct the unitary $V(|0\rangle|0\rangle) \longmapsto \frac{1}{\sqrt{n}} \sum_{i\in[n]} |i\rangle|0\rangle$

$$\longmapsto \frac{1}{\sqrt{n}} \sum_{i\in[n]} |i\rangle\left(\sqrt{\frac{p_i}{p_{\max}}}|0\rangle + \sqrt{1 - \frac{p_i}{p_{\max}}}|1\rangle\right)$$

Sampling (repeat T times):

Importance Sampling with Quantum State preparation

(Grover 2000)

Preprocessing:

1. Compute $p_{\max} = \max\{p_1, \ldots, p_n\}$ with quantum **Maximum Finding**

2. Construct the unitary $V(|0\rangle|0\rangle) \longmapsto \frac{1}{\sqrt{n}} \sum_{i\in[n]} |i\rangle|0\rangle$

$$\longmapsto \frac{1}{\sqrt{n}} \sum_{i\in[n]} |i\rangle\left(\sqrt{\frac{p_i}{p_{\max}}}|0\rangle + \sqrt{1-\frac{p_i}{p_{\max}}}|1\rangle\right)$$

$$= \frac{1}{\sqrt{np_{\max}}}\left(\sum_i \sqrt{p_i}|i\rangle\right)|0\rangle + \ldots|1\rangle$$

Sampling (repeat T times):

Importance Sampling with Quantum State preparation

(Grover 2000)

Preprocessing:

1. Compute $p_{\max} = \max\{p_1, \ldots, p_n\}$ with quantum **Maximum Finding**

2. Construct the unitary $V(|0\rangle|0\rangle) \longmapsto \frac{1}{\sqrt{n}} \sum_{i \in [n]} |i\rangle|0\rangle$

$$\longmapsto \frac{1}{\sqrt{n}} \sum_{i \in [n]} |i\rangle \left(\sqrt{\frac{p_i}{p_{\max}}} |0\rangle + \sqrt{1 - \frac{p_i}{p_{\max}}} |1\rangle \right)$$

$$= \frac{1}{\sqrt{np_{\max}}} \left(\sum_i \sqrt{p_i} |i\rangle \right) |0\rangle + \ldots |1\rangle$$

Sampling **(repeat T times)**:

1. Prepare $\sum_i \sqrt{p_i} |i\rangle$ with **Amplitude Amplification** on V, and measure it.

Importance Sampling with Quantum State preparation

(Grover 2000)

Preprocessing:

1. Compute $p_{\max} = \max\{p_1, \ldots, p_n\}$ with quantum **Maximum Finding**

2. Construct the unitary $V(|0\rangle|0\rangle) \longmapsto \frac{1}{\sqrt{n}} \sum_{i \in [n]} |i\rangle|0\rangle$

$$\longmapsto \frac{1}{\sqrt{n}} \sum_{i \in [n]} |i\rangle \left(\sqrt{\frac{p_i}{p_{\max}}} |0\rangle + \sqrt{1 - \frac{p_i}{p_{\max}}} |1\rangle \right)$$

$$= \frac{1}{\sqrt{n p_{\max}}} \left(\sum_i \sqrt{p_i} |i\rangle \right) |0\rangle + \ldots |1\rangle$$

Sampling (repeat T times):

1. Prepare $\sum_i \sqrt{p_i} |i\rangle$ with **Amplitude Amplification** on V, and measure it.

Preprocessing time: $O(\sqrt{n})$

Cost per sample: $O\left(\sqrt{n p_{\max}}\right)$

Importance Sampling with Quantum State preparation

(Grover 2000)

Preprocessing:

1. Compute $p_{\max} = \max\{p_1, \ldots, p_n\}$ with quantum **Maximum Finding**

2. Construct the unitary $V(|0\rangle|0\rangle) \longmapsto \frac{1}{\sqrt{n}} \sum_{i \in [n]} |i\rangle|0\rangle$

$$\longmapsto \frac{1}{\sqrt{n}} \sum_{i \in [n]} |i\rangle \left(\sqrt{\frac{p_i}{p_{\max}}} |0\rangle + \sqrt{1 - \frac{p_i}{p_{\max}}} |1\rangle \right)$$

$$= \frac{1}{\sqrt{n p_{\max}}} \left(\sum_i \sqrt{p_i} |i\rangle \right) |0\rangle + \ldots |1\rangle$$

Sampling **(repeat T times):**

1. Prepare $\sum_i \sqrt{p_i} |i\rangle$ with **Amplitude Amplification** on V, and measure it.

Preprocessing time: $O(\sqrt{n})$

Cost per sample: $O(\sqrt{n p_{\max}})$

Cost for T samples: $O(\sqrt{n} + T\sqrt{n p_{\max}})$

Importance Sampling with Quantum State preparation

(Grover 2000)

Preprocessing:

1. Compute $p_{\max} = \max\{p_1, \ldots, p_n\}$ with quantum **Maximum Finding**

2. Construct the unitary $V(|0\rangle|0\rangle) \longmapsto \frac{1}{\sqrt{n}} \sum_{i \in [n]} |i\rangle|0\rangle$

$$\longmapsto \frac{1}{\sqrt{n}} \sum_{i \in [n]} |i\rangle \left(\sqrt{\frac{p_i}{p_{\max}}} |0\rangle + \sqrt{1 - \frac{p_i}{p_{\max}}} |1\rangle \right)$$

$$= \frac{1}{\sqrt{n p_{\max}}} \left(\sum_i \sqrt{p_i} |i\rangle \right) |0\rangle + \ldots |1\rangle$$

Sampling **(repeat T times):**

1. Prepare $\sum_i \sqrt{p_i} |i\rangle$ with **Amplitude Amplification** on V, and measure it.

Preprocessing time: $O(\sqrt{n})$

Cost per sample: $O(\sqrt{n p_{\max}})$

Cost for T samples: $O(\sqrt{n} + T\sqrt{n p_{\max}}) = O(T\sqrt{n})$

Importance Sampling with Quantum State preparation

(Grover 2000)

Preprocessing:

1. Compute $p_{\max} = \max\{p_1, \ldots, p_n\}$ with quantum **Maximum Finding**

2. Construct the unitary $V(|0\rangle|0\rangle) \longmapsto \frac{1}{\sqrt{n}} \sum_{i \in [n]} |i\rangle|0\rangle$

$$\longmapsto \frac{1}{\sqrt{n}} \sum_{i \in [n]} |i\rangle\left(\sqrt{\frac{p_i}{p_{\max}}}|0\rangle + \sqrt{1 - \frac{p_i}{p_{\max}}}|1\rangle\right)$$

$$= \frac{1}{\sqrt{np_{\max}}}\left(\sum_i \sqrt{p_i}|i\rangle\right)|0\rangle + \ldots|1\rangle$$

Sampling **(repeat T times):**

1. Prepare $\sum_i \sqrt{p_i}|i\rangle$ with **Amplitude Amplification** on V, and measure it.

Preprocessing time: $O(\sqrt{n})$

Cost per sample: $O(\sqrt{np_{\max}})$

Cost for T samples: $O(\sqrt{n} + T\sqrt{np_{\max}}) = O(T\sqrt{n})$

Our result: $\boxed{O(\sqrt{Tn})}$

Importance Sampling with a Quantum Oracle

Our result: $O\left(\sqrt{Tn}\right)$ for obtaining T independent samples from D = (p_1,…,p_n).

Importance Sampling with a Quantum Oracle

Our result: $O(\sqrt{Tn})$ for obtaining T independent samples from D = (p_1,…,p_n).

Element	1	2	3	4	5	6	7
Probability	p_1	p_2	p_3	p_4	p_5	p_6	p_7

Distribution D

Importance Sampling with a Quantum Oracle

Our result: $O\left(\sqrt{Tn}\right)$ for obtaining T independent samples from D = $(p_1,\ldots,p_n)$.

Element	1	2	3	4	5	6	7
Probability	p_1	p_2	p_3	p_4	p_5	p_6	p_7

Distribution D

$$D \xrightarrow[P_{\text{Heavy}} = \sum_{i \in \text{Heavy}} p_i]{p_i \geq 1/T} D_{\text{Heavy}}$$

Element	1	3	4
Probability	$\frac{p_1}{P_{\text{Heavy}}}$	$\frac{p_3}{P_{\text{Heavy}}}$	$\frac{p_4}{P_{\text{Heavy}}}$

Distribution D_{Heavy}

Importance Sampling with a Quantum Oracle

Our result: $O(\sqrt{Tn})$ for obtaining T independent samples from D = ($p_1,\ldots,p_n$).

Element	1	2	3	4	5	6	7
Probability	p_1	p_2	p_3	p_4	p_5	p_6	p_7

Distribution D

$p_i \geq 1/T$

$P_{\text{Heavy}} = \sum_{i \in \text{Heavy}} p_i$

$p_i < 1/T$

$P_{\text{Light}} = \sum_{i \in \text{Light}} p_i$

Element	1	3	4
Probability	$\frac{p_1}{P_{\text{Heavy}}}$	$\frac{p_3}{P_{\text{Heavy}}}$	$\frac{p_4}{P_{\text{Heavy}}}$

Distribution D_{Heavy}

Element	2	5	6	7
Probability	$\frac{p_2}{P_{\text{Light}}}$	$\frac{p_5}{P_{\text{Light}}}$	$\frac{p_6}{P_{\text{Light}}}$	$\frac{p_7}{P_{\text{Light}}}$

Distribution D_{Light}

Importance Sampling with a Quantum Oracle

Our result: $O(\sqrt{Tn})$ for obtaining T independent samples from D = ($p_1,\ldots,p_n$).

Element	1	2	3	4	5	6	7
Probability	p_1	p_2	p_3	p_4	p_5	p_6	p_7

Distribution D

$p_i \geq 1/T$

$P_{\text{Heavy}} = \sum_{i \in \text{Heavy}} p_i$

$p_i < 1/T$

$P_{\text{Light}} = \sum_{i \in \text{Light}} p_i$

Element	1	3	4
Probability	$\frac{p_1}{P_{\text{Heavy}}}$	$\frac{p_3}{P_{\text{Heavy}}}$	$\frac{p_4}{P_{\text{Heavy}}}$

Distribution D_{Heavy}

Element	2	5	6	7
Probability	$\frac{p_2}{P_{\text{Light}}}$	$\frac{p_5}{P_{\text{Light}}}$	$\frac{p_6}{P_{\text{Light}}}$	$\frac{p_7}{P_{\text{Light}}}$

Distribution D_{Light}

Use the Alias method

Importance Sampling with a Quantum Oracle

Our result: $O(\sqrt{Tn})$ for obtaining T independent samples from D = ($p_1,\ldots,p_n$).

Element	1	2	3	4	5	6	7
Probability	p_1	p_2	p_3	p_4	p_5	p_6	p_7

Distribution D

$p_i \geq 1/T$

$P_{\text{Heavy}} = \sum_{i \in \text{Heavy}} p_i$

$p_i < 1/T$

$P_{\text{Light}} = \sum_{i \in \text{Light}} p_i$

Element	1	3	4
Probability	$\frac{p_1}{P_{\text{Heavy}}}$	$\frac{p_3}{P_{\text{Heavy}}}$	$\frac{p_4}{P_{\text{Heavy}}}$

Distribution D_{Heavy}

Use the Alias method

Element	2	5	6	7
Probability	$\frac{p_2}{P_{\text{Light}}}$	$\frac{p_5}{P_{\text{Light}}}$	$\frac{p_6}{P_{\text{Light}}}$	$\frac{p_7}{P_{\text{Light}}}$

Distribution D_{Light}

Use Quantum State Preparation

Importance Sampling with a Quantum Oracle

Preprocessing:

Importance Sampling with a Quantum Oracle

Preprocessing:

1. Compute the set **Heavy** $\subset [n]$ of indices i such that $\mathbf{p_i \geq 1/T}$, using **Grover Search**.

Importance Sampling with a Quantum Oracle

Preprocessing:

1. Compute the set **Heavy** $\subset [n]$ of indices i such that $\mathbf{p_i \geq 1/T}$, using **Grover Search**.

2. Compute $\mathbf{P_{Heavy}} = \sum_{i \in \text{Heavy}} p_i$

Importance Sampling with a Quantum Oracle

Preprocessing:

1. Compute the set **Heavy** $\subset [n]$ of indices i such that $\mathbf{p_i \geq 1/T}$, using **Grover Search**.

2. Compute $\mathbf{P_{Heavy}} = \sum_{i \in \mathrm{Heavy}} p_i$

3. Apply the preprocessing step of the **Alias Method** on $\mathbf{D_{Heavy}}$.

Importance Sampling with a Quantum Oracle

Preprocessing:

1. Compute the set **Heavy** $\subset$ [n] of indices i such that **$p_i \geq 1/T$**, using **Grover Search**.

2. Compute **P_{Heavy}** $= \sum_{i \in \text{Heavy}} p_i$

3. Apply the preprocessing step of the **Alias Method** on **D_{Heavy}**.

4. Apply the preprocessing step of the **Quant. State Preparation** method on **D_{Light}**.

Importance Sampling with a Quantum Oracle

Sampling (repeat T times):

Importance Sampling with a Quantum Oracle

Sampling **(repeat T times)**:

Flip a coin that is head with probability $\mathbf{P_{Heavy}}$:

Importance Sampling with a Quantum Oracle

Sampling **(repeat T times):**

Flip a coin that is head with probability P_{Heavy} :

- **Head**: sample $i \sim D_{Heavy}$ with the **Alias Method.**

Importance Sampling with a Quantum Oracle

Sampling (repeat T times):

Flip a coin that is head with probability P_{Heavy} :

- **Head**: sample $i \sim D_{Heavy}$ with the **Alias Method.**
- **Tail**: sample $i \sim D_{Light}$ with **Quantum State Preparation.**

Importance Sampling with a Quantum Oracle

Preprocessing:

1. Compute the set **Heavy** ⊂ [n] of indices i such that $p_i \geq 1/T$, using **Grover Search**.

   Cost:

2. Compute $P_{\text{Heavy}} = \sum_{i \in \text{Heavy}} p_i$

   Cost:

3. Apply the preprocessing step of the **Alias Method** on D_{Heavy}.

   Cost:

4. Apply the preprocessing step of the **Quant. State Preparation** method on D_{Light}.

   Cost:

Importance Sampling with a Quantum Oracle

Preprocessing:

1. Compute the set **Heavy** ⊂ [n] of indices i such that **$p_i \geq 1/T$**, using **Grover Search**.

   Cost: $O(\sqrt{nT})$ since **|Heavy| ≤ T**

2. Compute **P_{Heavy}** $= \sum_{i \in \text{Heavy}} p_i$

   Cost:

3. Apply the preprocessing step of the **Alias Method** on **D_{Heavy}**.

   Cost:

4. Apply the preprocessing step of the **Quant. State Preparation** method on **D_{Light}**.

   Cost:

Importance Sampling with a Quantum Oracle

Preprocessing:

1. Compute the set **Heavy** $\subset [n]$ of indices i such that $p_i \geq 1/T$, using **Grover Search**.

   Cost: $O(\sqrt{nT})$ since **|Heavy| ≤ T**

2. Compute $P_{\text{Heavy}} = \sum_{i \in \text{Heavy}} p_i$

   Cost: $O(T)$

3. Apply the preprocessing step of the **Alias Method** on D_{Heavy}.

   Cost:

4. Apply the preprocessing step of the **Quant. State Preparation** method on D_{Light}.

   Cost:

Importance Sampling with a Quantum Oracle

Preprocessing:

1. Compute the set **Heavy** ⊂ [n] of indices i such that **$p_i \geq 1/T$**, using **Grover Search**.

   Cost: $O(\sqrt{nT})$ since **|Heavy| ≤ T**

2. Compute **P_{Heavy}** $= \sum_{i \in \mathrm{Heavy}} p_i$

   Cost: $O(T)$

3. Apply the preprocessing step of the **Alias Method** on **D_{Heavy}**.

   Cost: $O(T)$

4. Apply the preprocessing step of the **Quant. State Preparation** method on **D_{Light}**.

   Cost:

Importance Sampling with a Quantum Oracle

Preprocessing:

1. Compute the set **Heavy** ⊂ [n] of indices i such that $p_i \geq 1/T$, using **Grover Search**.

   Cost: $O(\sqrt{nT})$ since $|\text{Heavy}| \leq T$

2. Compute $P_{\text{Heavy}} = \sum_{i \in \text{Heavy}} p_i$

   Cost: $O(T)$

3. Apply the preprocessing step of the **Alias Method** on D_{Heavy}.

   Cost: $O(T)$

4. Apply the preprocessing step of the **Quant. State Preparation** method on D_{Light}.

   Cost: $O(\sqrt{n})$

Importance Sampling with a Quantum Oracle

Sampling **(repeat T times)**:

Flip a coin that is head with probability **P_{Heavy}** :

- **Head**: sample **$i \sim D_{Heavy}$** with the **Alias Method.**

  Cost per sample:

- **Tail**: sample **$i \sim D_{Light}$** with **Quantum State Preparation.**

Importance Sampling with a Quantum Oracle

Sampling (repeat T times):

Flip a coin that is head with probability P_{Heavy} :

- **Head**: sample $i \sim D_{Heavy}$ with the **Alias Method.**

  Cost per sample: $O(1)$

- **Tail**: sample $i \sim D_{Light}$ with **Quantum State Preparation.**

Importance Sampling with a Quantum Oracle

Sampling (repeat T times):

Flip a coin that is head with probability P_{Heavy} :

- **Head**: sample $i \sim D_{Heavy}$ with the **Alias Method.**

  Cost per sample: $O(1)$ **Total cost:** $O(T)$

- **Tail**: sample $i \sim D_{Light}$ with **Quantum State Preparation.**

Importance Sampling with a Quantum Oracle

Sampling (repeat T times):

Flip a coin that is head with probability **P_{Heavy}** :

- **Head**: sample **i ~ D_{Heavy}** with the **Alias Method.**

  Cost per sample: $O(1)$ **Total cost:** $O(T)$

- **Tail**: sample **i ~ D_{Light}** with **Quantum State Preparation.**

  Cost per sample:

Importance Sampling with a Quantum Oracle

Sampling **(repeat T times):**

Flip a coin that is head with probability **P_{Heavy}** :

- **Head**: sample **i ~ D_{Heavy}** with the **Alias Method.**

  Cost per sample: $O(1)$ **Total cost:** $O(T)$

- **Tail**: sample **i ~ D_{Light}** with **Quantum State Preparation.**

  Cost per sample: $O\left(\sqrt{np_{\max}}\right)$ where $p_{\max} = \max\left\{\frac{p_i}{P_{\text{Light}}} : i \in \text{Light}\right\}$

Importance Sampling with a Quantum Oracle

Sampling **(repeat T times):**

Flip a coin that is head with probability **P_{Heavy}** :

- **Head**: sample **i ~ D_{Heavy}** with the **Alias Method.**

  Cost per sample: $O(1)$ **Total cost:** $O(T)$

- **Tail**: sample **i ~ D_{Light}** with **Quantum State Preparation.**

  Cost per sample: $O\left(\sqrt{np_{\max}}\right)$ where $p_{\max} = \max\left\{\frac{p_i}{P_{\text{Light}}} : i \in \text{Light}\right\}$

$$\leq \frac{1}{T \cdot P_{\text{Light}}}$$

Importance Sampling with a Quantum Oracle

Sampling **(repeat T times):**

Flip a coin that is head with probability **P_{Heavy}** :

- **Head**: sample **i ~ D_{Heavy}** with the **Alias Method.**

  Cost per sample: $O(1)$ **Total cost:** $O(T)$

- **Tail**: sample **i ~ D_{Light}** with **Quantum State Preparation.**

  Cost per sample: $O\left(\sqrt{np_{\max}}\right)$ where $p_{\max} = \max\left\{\frac{p_i}{P_{\text{Light}}} : i \in \text{Light}\right\}$

  $$\leq \frac{1}{T \cdot P_{\text{Light}}}$$

Total expected cost:

Importance Sampling with a Quantum Oracle

Sampling **(repeat T times):**

Flip a coin that is head with probability **P_{Heavy}** :

- **Head**: sample **i ~ D_{Heavy}** with the **Alias Method.**

  Cost per sample: $O(1)$ **Total cost:** $O(T)$

- **Tail**: sample **i ~ D_{Light}** with **Quantum State Preparation.**

  Cost per sample: $O\left(\sqrt{np_{\max}}\right)$ where $p_{\max} = \max\left\{\frac{p_i}{P_{\text{Light}}} : i \in \text{Light}\right\}$

  $$\leq \frac{1}{T \cdot P_{\text{Light}}}$$

Total expected cost: $O\left(T \cdot P_{\text{Light}} \cdot \sqrt{np_{\max}}\right)$

Importance Sampling with a Quantum Oracle

Sampling **(repeat T times):**

Flip a coin that is head with probability P_{Heavy} :

- **Head**: sample **i ~ D_{Heavy}** with the **Alias Method.**

  Cost per sample: $O(1)$ **Total cost:** $O(T)$

- **Tail**: sample **i ~ D_{Light}** with **Quantum State Preparation.**

  Cost per sample: $O\left(\sqrt{np_{\max}}\right)$ where $p_{\max} = \max\left\{\frac{p_i}{P_{\text{Light}}} : i \in \text{Light}\right\}$

  $$\leq \frac{1}{T \cdot P_{\text{Light}}}$$

Total expected cost: $O\left(T \cdot P_{\text{Light}} \cdot \sqrt{np_{\max}}\right) = O\left(T \cdot P_{\text{Light}} \cdot \sqrt{\frac{n}{T \cdot P_{\text{Light}}}}\right)$

Importance Sampling with a Quantum Oracle

Sampling **(repeat T times):**

Flip a coin that is head with probability **P_{Heavy}** :

- **Head**: sample **$i \sim D_{Heavy}$** with the **Alias Method.**

  Cost per sample: $O(1)$ **Total cost:** $O(T)$

- **Tail**: sample **$i \sim D_{Light}$** with **Quantum State Preparation.**

  Cost per sample: $O\left(\sqrt{np_{\max}}\right)$ where $p_{\max} = \max\left\{\frac{p_i}{P_{\text{Light}}} : i \in \text{Light}\right\} \le \frac{1}{T \cdot P_{\text{Light}}}$

  Total expected cost: $O\left(T \cdot P_{\text{Light}} \cdot \sqrt{np_{\max}}\right) = O\left(T \cdot P_{\text{Light}} \cdot \sqrt{\frac{n}{T \cdot P_{\text{Light}}}}\right) = O\left(\sqrt{nTP_{\text{Light}}}\right)$

Importance Sampling with a Quantum Oracle

Sampling **(repeat T times):**

Flip a coin that is head with probability **P_{Heavy}** :

- **Head**: sample **$i \sim D_{Heavy}$** with the **Alias Method.**

  Cost per sample: $O(1)$ **Total cost:** $O(T)$

- **Tail**: sample **$i \sim D_{Light}$** with **Quantum State Preparation.**

  Cost per sample: $O\left(\sqrt{np_{\max}}\right)$ where $p_{\max} = \max\left\{\frac{p_i}{P_{\text{Light}}} : i \in \text{Light}\right\} \leq \frac{1}{T \cdot P_{\text{Light}}}$

  Total expected cost: $O\left(T \cdot P_{\text{Light}} \cdot \sqrt{np_{\max}}\right) = O\left(T \cdot P_{\text{Light}} \cdot \sqrt{\frac{n}{T \cdot P_{\text{Light}}}}\right) = O\left(\sqrt{nTP_{\text{Light}}}\right) = O\left(\sqrt{nT}\right)$

Conclusion

Open questions:

- Can we obtain a quantum speedup for exact/approximate submodular function minimization?

- Can we improve the lower bound for exact/approximate submodular function minimization?

- Can we prepare T copies of the state $\sum_{i \in [n]} \sqrt{p_i} \, | \, i \rangle$ in time $O(\sqrt{nT})$.

arXiv: 1907.05378