A^* Sampling

Chris J. Maddison
University of Toronto

Daniel Tarlow
Microsoft Research

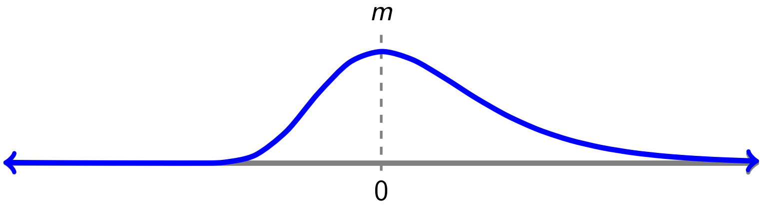
Tom Minka
Microsoft Research

Goal: Given an unnormalized log density $\phi(x)$, produce independent samples $x_1, \ldots, x_n$ from the Gibbs distribution $p(x) \propto \exp(\phi(x))$.

The Gumbel Distribution

$G \sim \text{Gumbel}(m)$ is Gumbel distributed with location m,
if its density is

$$p(g) = \exp(-g + m) \exp(-\exp(-g + m))$$



The Gumbel Distribution

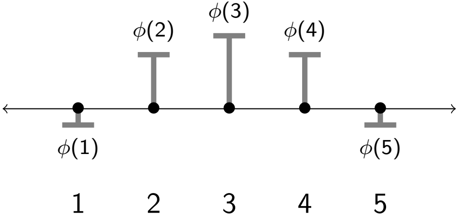
The Gumbel distribution is max-stable.
If $G_i \sim \text{Gumbel}(0)$ IID, then

$$\max\{G_1, G_2\} \sim \text{Gumbel}(\log 2)$$

The Gumbel-Max Trick (well-known, see Yellott 1977)

Suppose we want to sample from a finite distribution

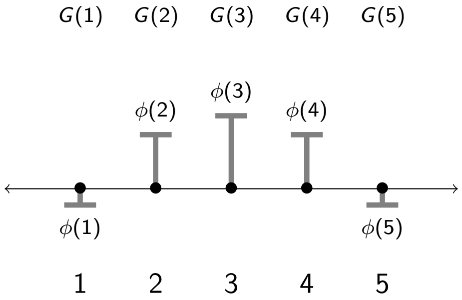
$$p(i) \propto \exp(\phi(i)) \text{ for } i \in \{1, 2, 3, 4, 5\}$$



The Gumbel-Max Trick (well-known, see Yellott 1977)

Suppose we want to sample from a finite distribution

$$p(i) \propto \exp(\phi(i)) \text{ for } i \in \{1, 2, 3, 4, 5\}$$

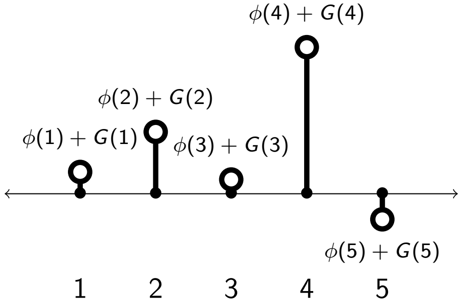


$G(i) \sim$ Gumbel(0) IID

The Gumbel-Max Trick (well-known, see Yellott 1977)

Suppose we want to sample from a finite distribution

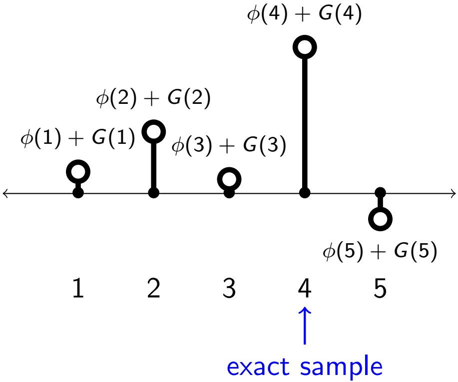
$$p(i) \propto \exp(\phi(i)) \text{ for } i \in \{1, 2, 3, 4, 5\}$$



The Gumbel-Max Trick (well-known, see Yellott 1977)

Suppose we want to sample from a finite distribution

$$p(i) \propto \exp(\phi(i)) \text{ for } i \in \{1, 2, 3, 4, 5\}$$



The Gumbel-Max Trick (well-known, see Yellott 1977)

More formally for any subset B of the indices.

$$\operatorname*{argmax}_{i \in B} G(i) + \phi(i) \sim \frac{\exp(\phi(i))\mathbf{1}(i \in B)}{\sum_{i \in B} \exp(\phi(i))}$$

$$\max_{i \in B} G(i) + \phi(i) \sim \text{Gumbel}(\log \sum_{i \in B} \exp(\phi(i)))$$

What about continuous space?

What about continuous space?

1. Is there an analogous process for perturbing infinite spaces?
2. Can we define practical algorithms for optimizing it?

Perturbing Continuous Space

Now we are interested in

$$p(x) \propto \exp(\phi(x)) \text{ for } x \in \mathbb{R}^d$$
$$\mu(B) = \int_{x \in B} \exp(\phi(x)) \text{ for } B \subseteq \mathbb{R}^d$$

For this talk just look at $\mathbb{R}$.

A Quick Re-Frame

We produced a sequence of Gumbels and locations

$$(G(1) + \phi(1), 1) \quad \ldots \quad (G(5) + \phi(5), 5)$$

such that

$$\max\{G(i) + \phi(i) \mid i \in B\} \sim \text{Gumbel}(\log \sum_{i \in B} \exp(\phi(i)))$$

$$\text{argmax}\{G(i) + \phi(i) \mid i \in B\} \sim \frac{\exp(\phi(i))\mathbf{1}(i \in B)}{\sum_{i \in B} \exp(\phi(i))}$$

Perturbing Continuous Space

By analogy, we want a sequence (G_k, X_k) for $k \to \infty$ such that

$$\max\{G_k \mid X_k \in B\} \sim \text{Gumbel}(\log \mu(B))$$
$$\text{argmax}\{G_k \mid X_k \in B\} \sim \frac{\exp(\phi(x))\mathbf{1}(x \in B)}{\int_{i \in B} \exp(\phi(x))}$$

Perturbing Continuous Space

bottom-up: instantiate noise $\rightarrow$ find maxes

- Generating infinitely many random variables, *then* finding maxes is a non-starter.

Perturbing Continuous Space

bottom-up: instantiate noise $\rightarrow$ find maxes

- Generating infinitely many random variables, *then* finding maxes is a non-starter.

top-down: pick max $\rightarrow$ generate the rest

- Generate maxes over increasingly refined subsets of space.

Perturbing Continuous Space

bottom-up: instantiate noise $\rightarrow$ find maxes

- Generating infinitely many random variables, *then* finding maxes is a non-starter.

top-down: pick max $\rightarrow$ generate the rest

- Generate maxes over increasingly refined subsets of space.

With Gumbel noise, these two directions are equivalent.

Top-Down Construction

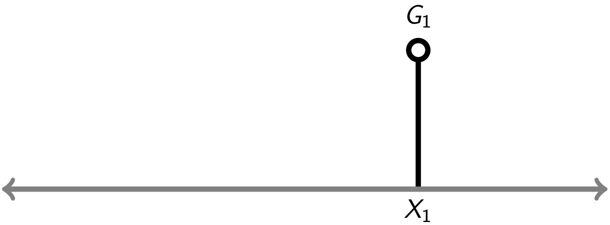
A stream (G_k, X_k) for $k = 1, \ldots, \infty$

G_k bounds the noise in its subset

$X_1 \sim \exp(\phi(x))/\mu(\mathbb{R})$

Top-Down Construction

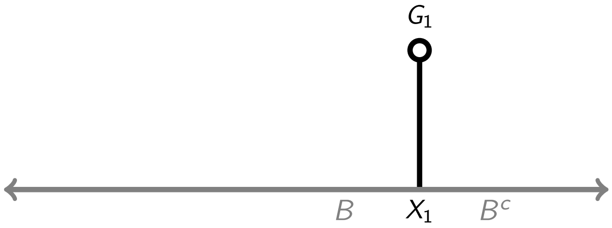
A stream (G_k, X_k) for $k = 1, \ldots, \infty$
G_k bounds the noise in its subset



$G_1 \sim \text{Gumbel}(\log \mu(\mathbb{R}))$

Top-Down Construction

A stream (G_k, X_k) for $k = 1, \ldots, \infty$
G_k bounds the noise in its subset

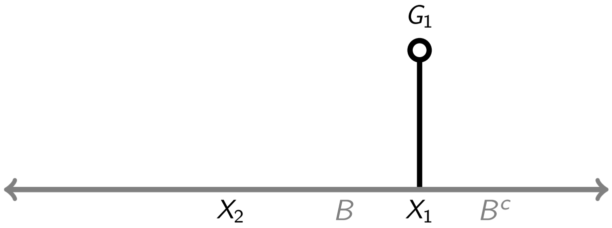


split space on X_1

Top-Down Construction

A stream (G_k, X_k) for $k = 1, \ldots, \infty$

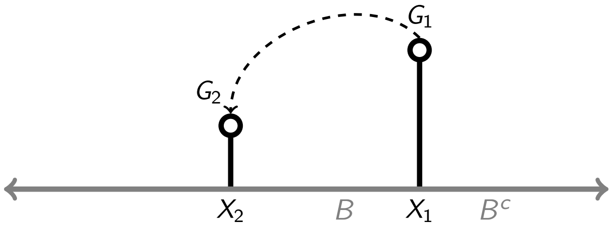
G_k bounds the noise in its subset



$$X_2 \sim \exp(\phi(x))\mathbf{1}(x \in B)\,/\mu(B)$$

Top-Down Construction

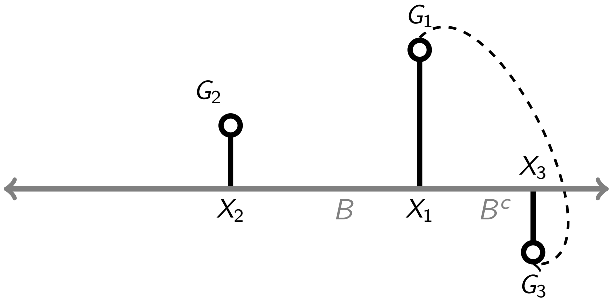
A stream (G_k, X_k) for $k = 1, \ldots, \infty$
G_k bounds the noise in its subset



$G_2 \sim \text{TruncGumbel}(\log \mu(B), G_1)$

Top-Down Construction

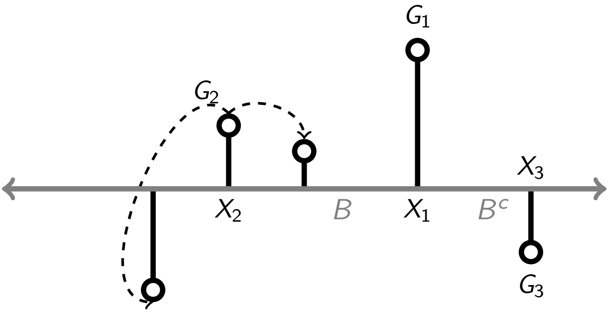
A stream (G_k, X_k) for $k = 1, \ldots, \infty$
G_k bounds the noise in its subset



Top-Down Construction

A stream (G_k, X_k) for $k = 1, \ldots, \infty$

G_k bounds the noise in its subset



recursively subdivide space and generate regional maxes

Perturbing Continuous Space

For $B \subseteq \mathbb{R}$

$$\max\{G_k \mid X_k \in B\} \sim \text{Gumbel}(\log \mu(B))$$

$$\operatorname{argmax}\{G_k \mid X_k \in B\} \sim \frac{\exp(\phi(x))\mathbf{1}(x \in B)}{\mu(\mathbb{R})}$$

Call $\{\max\{G_k \mid X_k \in B\} \mid B \subseteq \mathbb{R}\}$ a *Gumbel Process*.

Recap

1. We want to draw independent samples
2. We found a process whose optima are samples
3. But the procedure for generating it assumes we can draw independent samples

A^* Sampling

How to practically optimize a Gumbel process without assuming you can tractably sample from $p(x)$ and compute $\mu(B)$.

A* Sampling

Like in rejection sampling, decompose $\phi(x)$ into a tractable and boundable component

$$\phi(x) = i(x) + o(x)$$

where for region B we can tractably sample and compute volumes from $q(x) \propto \exp(i(x))$ and bound $o(x) \leq M_B$.

We can also decompose the Gumbel Process

A* Sampling

We can take (G_k^q, X_k^q), a stream of values from the Gumbel process for $q(x)$ and transform it into a realization of a Gumbel process for $p(x)$ by adding $o(x)$.

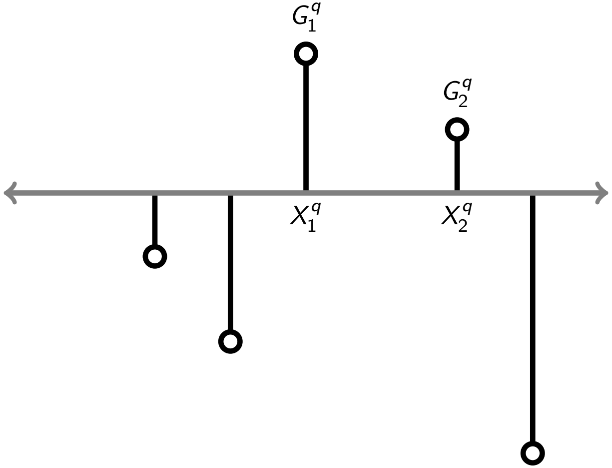
$$G_k^q + o(X_k^q) = G_k$$

A* Sampling

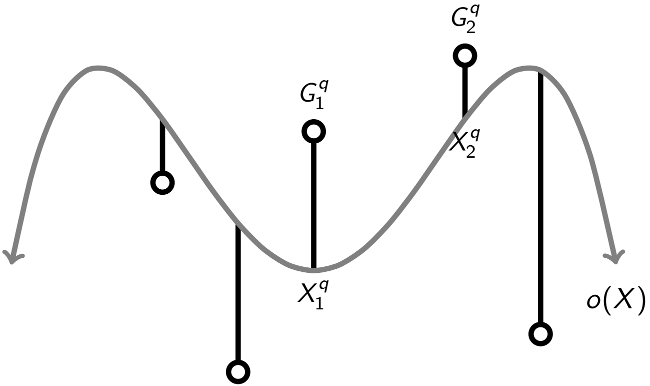
Take stream (G_k^q, X_k^q) for $q(x)$, then

$$\max\{G_k^q + o(X_k^q) \,|\, X_k^q \in B\} \sim \text{Gumbel}(\log \mu(B))$$
$$\text{argmax}\{G_k^q + o(X_k^q) \,|\, X_k^q \in B\} \sim \frac{\exp(\phi(x))\mathbf{1}(x \in B)}{\mu(B)}$$

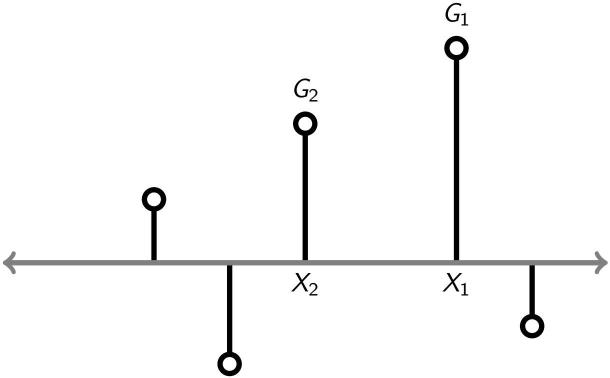
A* Sampling



A* Sampling



A* Sampling



A* Sampling

To draw a sample we want to find

$$\text{argmax}\{G_k^q + o(X_k^q)\}$$

This decomposition is useful because we can bound

- contribution from the noise of q Gumbel process
- contribution of $o(x)$ — this community is good at bounding these functions

$$\max\{G_k^q + o(X_k^q) \mid X_k^q \in B\} \leq \max\{G_k^q \mid X_k^q \in B\} + M_B$$

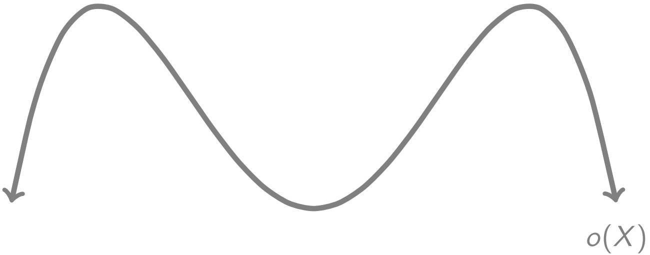
Core Idea: Use A* search to find the optimum.

A* Sampling — Ingredients

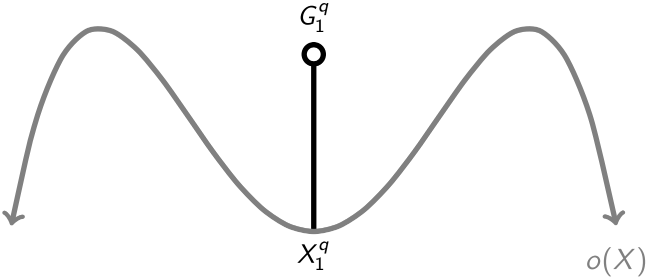
- The stream of values (G_k^q, X_k^q)
  - G_k^q bounds the noise in its subset.
- Upper bounds on a subset B, $G_k^q + M_B$
- Lower bounds on a subset B, $G_k^q + o(X_k^q)$

Generally, the two expensive operations are computing M_B and $o(X_k^q)$

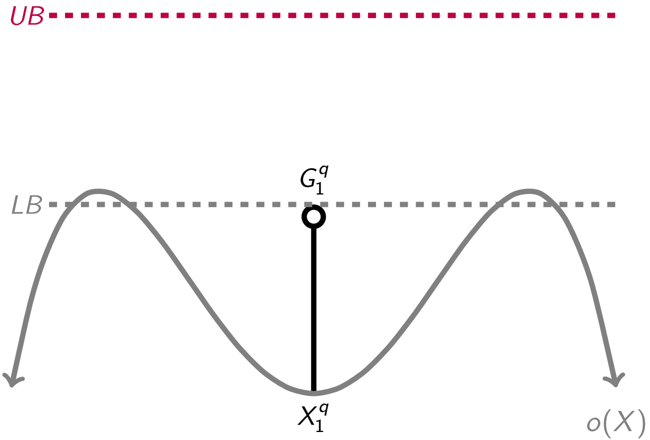
A* Sampling



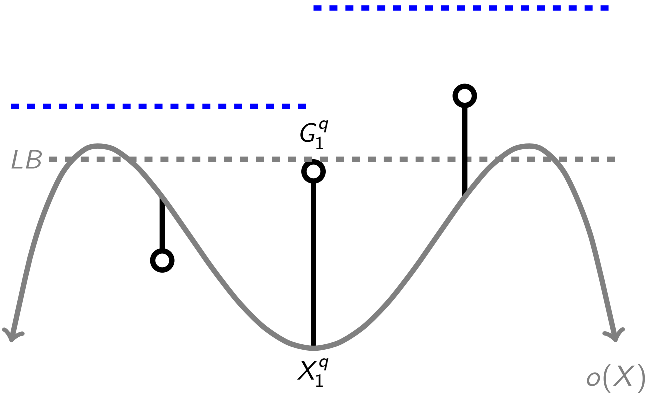
A* Sampling



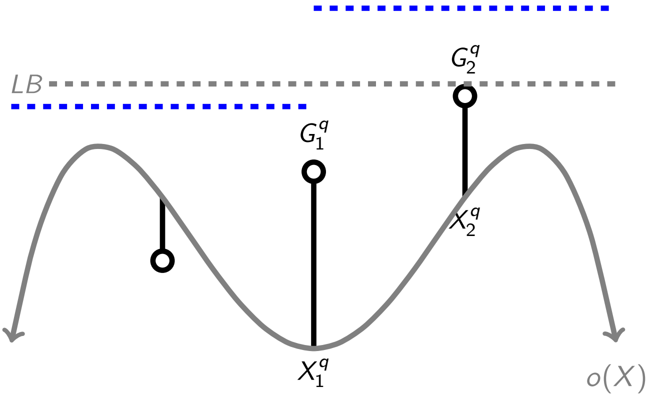
A* Sampling



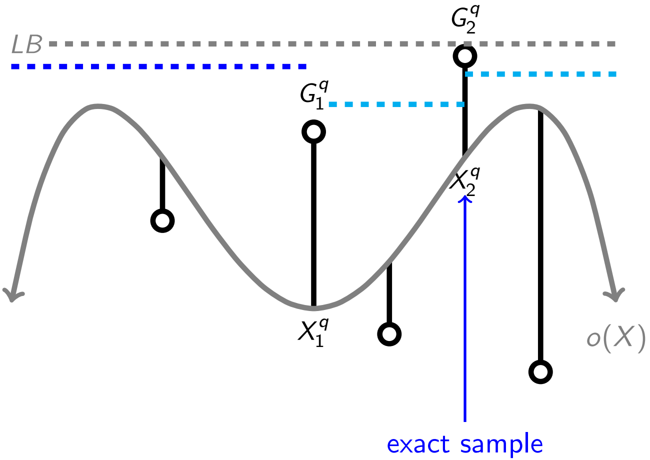
A* Sampling



A* Sampling



A* Sampling



Come see us at the poster

- Experiments relating A* sampling to other samplers
- Analysis relating A* sampling to adaptive rejection type samplers
  - A* sampling couples which regions are refined and where the sample is — more efficient use of bounds and likelihood.

Use Case

- Whenever you might sit down to implement slice sampling or rejection sampling for low dimensional non-trivial distributions consider A* sampling.
  - e.g. for the conditionals of a Gibbs sampler
  - In many cases more efficient that alternatives
- We do not solve the problem of high dimensions — scales poorly in the worst case.
  - Not surprising, because general & exact.

Conclusions

- Extended the Gumbel-Max trick to continuous spaces.
- Defined A* Sampling, a practical algorithm that optimizes a Gumbel process with A*.
- Result is new generic sampling algorithm and a useful perspective on the sampling problem.

Acknowledgments

Special thanks to:
James Martens
Radford Neal
Elad Mezuman
Roger Grosse