Generalizations of Gumbel Softmax Trick

Based on “Gradient Estimation with Stochastic Softmax Tricks” by Paulus M. B. et al.

Kirill Struminsky, October 9th 2020

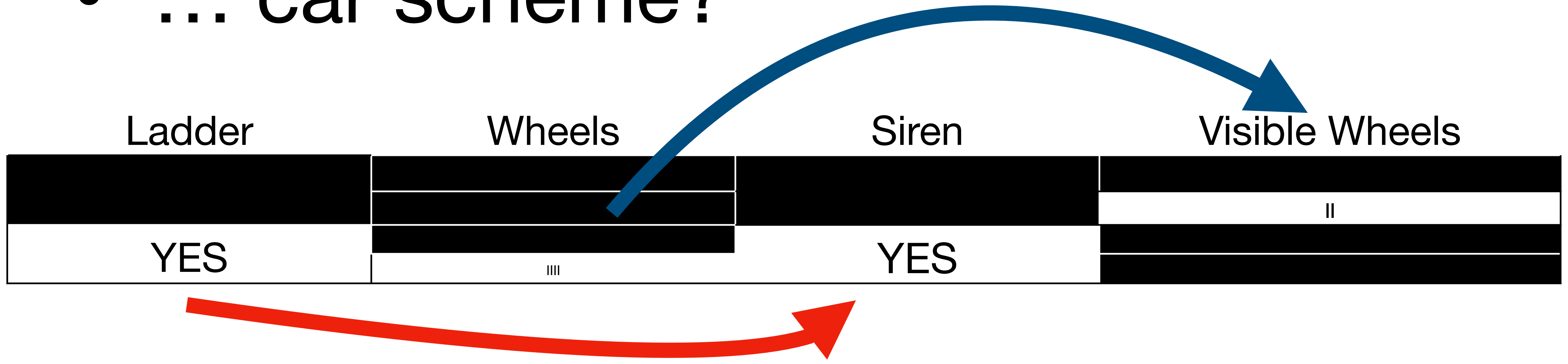
Motivation

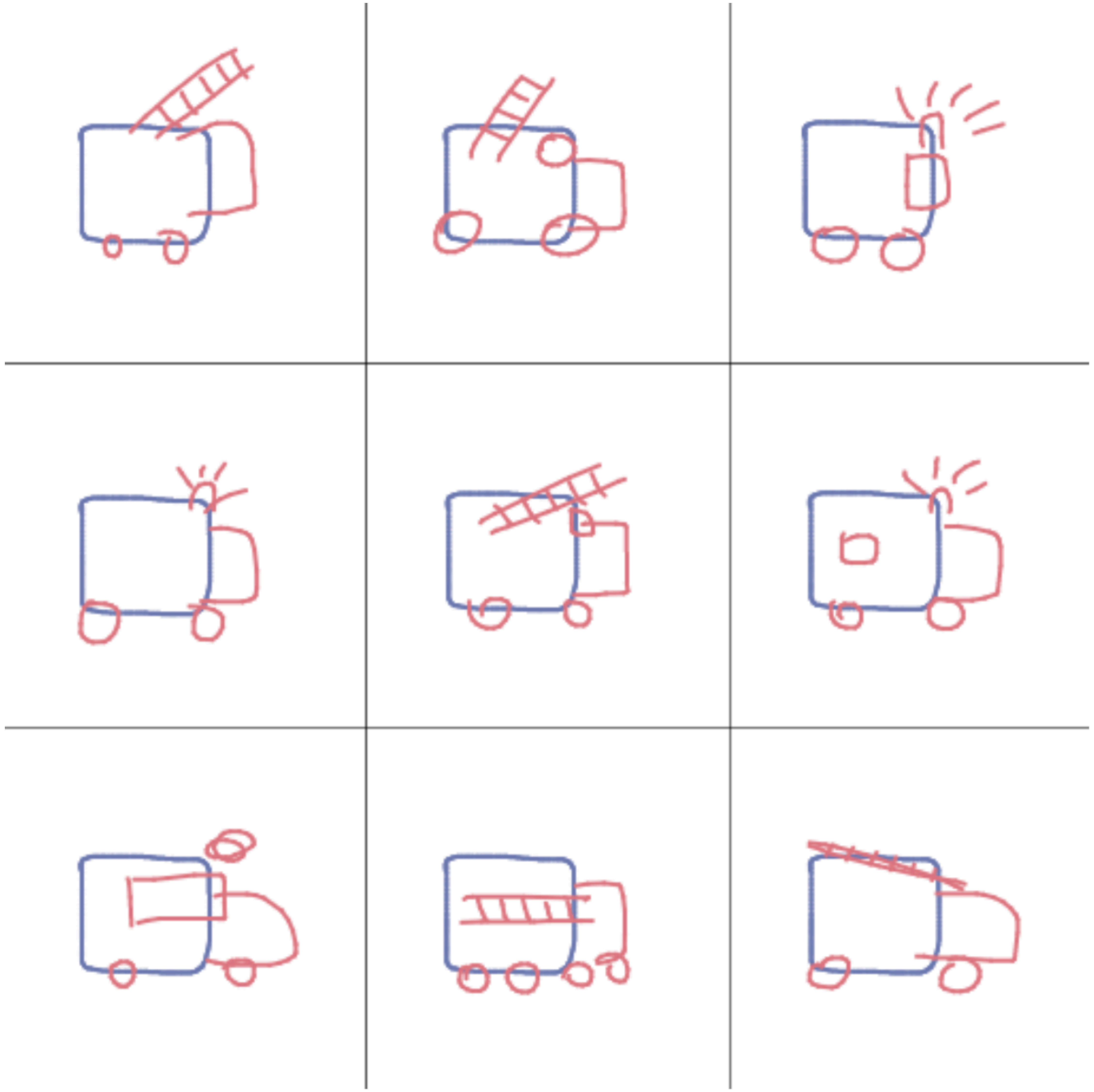
Low-level Motivation

- x is a car, the latent variable z is
  - Real vector

  - Independent categorical

  - … car scheme?





https://magenta.tensorflow.org/sketch-rnn-demo

High Level Motivation

- What do we want?
  - Structured latent variables
  - Learn computational layers with discrete outputs
- Why do we want it?
  - Interpretability
  - Computational efficiency
  - Better generalization

Gumbel Softmax Trick Recap

Jang E., Gu S., Poole B. Categorical reparameterization with gumbel-softmax
Maddison C. J., Mnih A., Teh Y. W. The concrete distribution: A continuous relaxation of discrete random variables

$$\frac{\mathrm{d}}{\mathrm{d}\theta}\mathbb{E}_{q(z|\theta)}f(z)\,?$$

REINFORCE

$$f(z)\frac{\mathrm{d}}{\mathrm{d}\theta}\log q(z \mid \theta)$$

Reparametrization trick

$$\frac{\mathrm{d}}{\mathrm{d}\theta}f(g_\theta(\varepsilon))$$

Williams R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning
Kingma D. P., Welling M. Auto-encoding variational bayes

Gumbel Argmax Trick

- $z \sim Cat(\operatorname{soft\,max} \theta), \theta \in \mathbb{R}^d$
- Gumbel trick defines $g_\theta(\varepsilon)$ for z
  - Let $\varepsilon_i = -\log(-\log u_i)$ for $u \sim U[0,1]^d$
  - Let $g_\theta(\varepsilon) = \underset{i=1,\ldots,d}{\operatorname{argmax}}(\theta + \varepsilon)$
  - Then $g_\theta(\varepsilon) \overset{d}{=} z$

$$\operatorname{soft\,max}(\theta) = (0.1,0.5,0.4)$$

$$u = (0.4,0.3,0.7)$$

$$g_\theta(\varepsilon) = \arg\max(-2.2, -0.9, +0.1)$$

$$= (0,0,1)$$

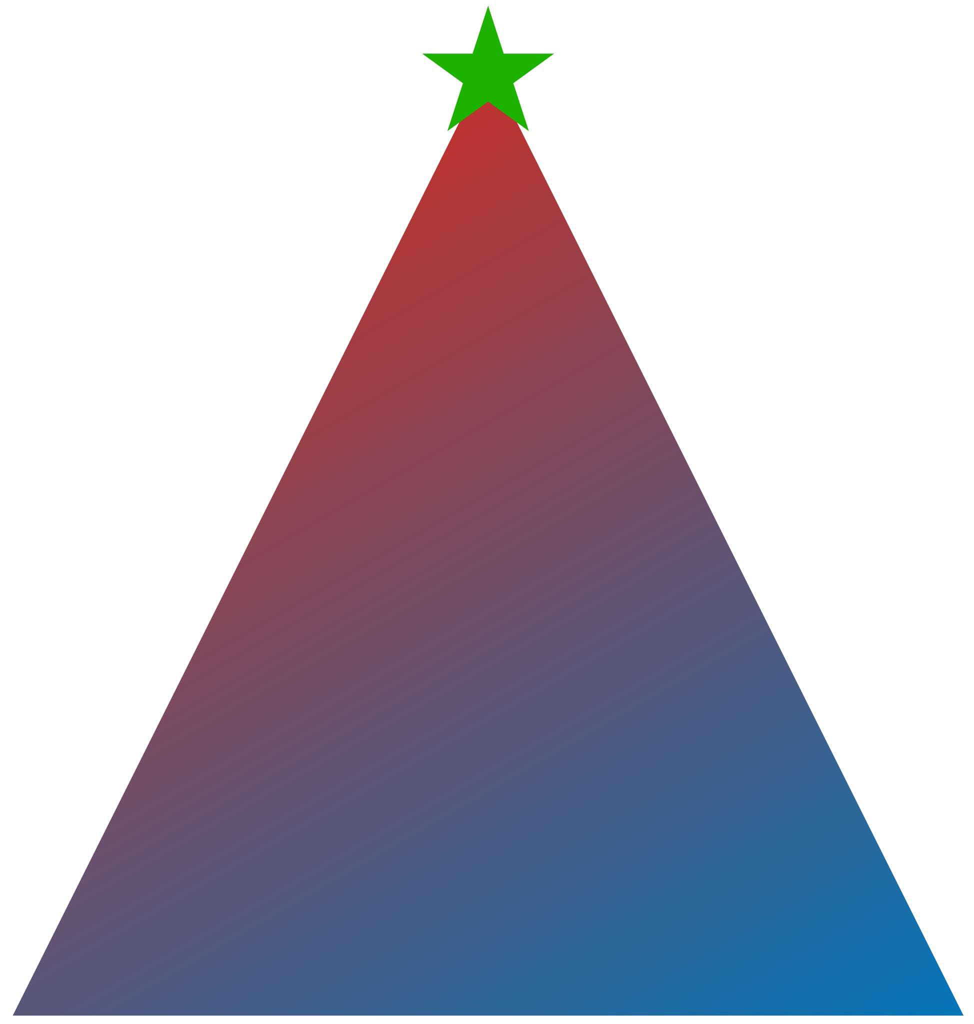
*Soft*max

- But $g_\theta(\varepsilon) = \arg\max(\theta + \varepsilon)$ is a piece-wise constant function of θ
- So $\mathbb{E}_{q(\varepsilon)} \frac{\mathrm{d}}{\mathrm{d}\theta} f(g_\theta(\varepsilon)) = 0 \neq \frac{\mathrm{d}}{\mathrm{d}\theta} \mathbb{E}_{q(z|\theta)} f(z)$
- **Idea:** replace $\arg\max(\,\cdot\,)$ with $\mathrm{soft}\max(\,\cdot\,)$
    - $\mathrm{soft}\max(\frac{\theta + \varepsilon}{T}) \stackrel{T \to 0}{\to} \arg\max(\theta + \varepsilon)$

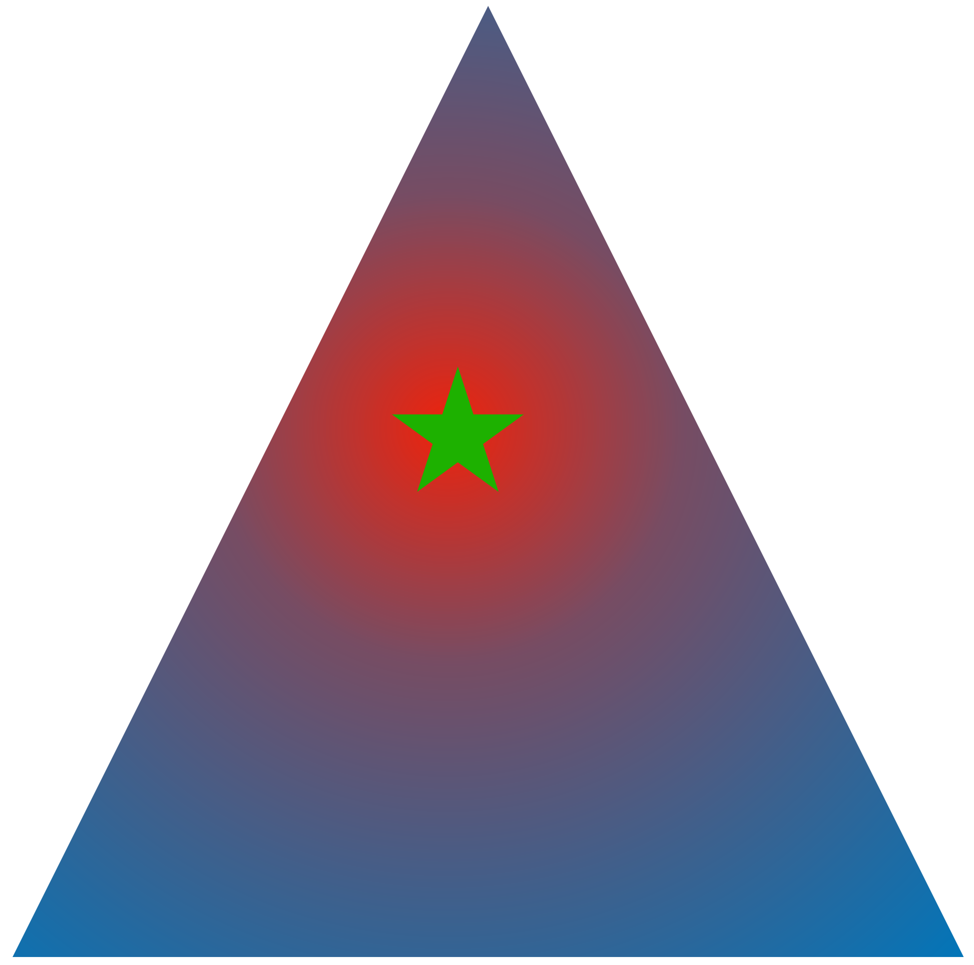
A Different View on Argmax and Softmax

- Rewrite $\underset{i=1,\ldots,d}{\operatorname{argmax}}\, w = \underset{z\in\Delta^d}{\operatorname{argmax}}\, w^T z$
  - Δ^d is a convex hull of one-hots for z
- Then $\operatorname{soft\,max}(\frac{w}{T}) = \underset{z\in\Delta^d}{\operatorname{argmax}}(w^T z + TH(z))$
  - $H(z) = -\sum_i z_i \log z_i$

$\arg\max$:



$\operatorname{soft\,max}$:



A Different View on Gumbel Softmax Trick

- Relaxed categorical sample is $z = \text{soft max}(\frac{\theta + \varepsilon}{T})$
- Equivalently $z = \underset{z \in \Delta^d}{\text{argmax}}((\theta + \varepsilon)^T z + TH(z))$

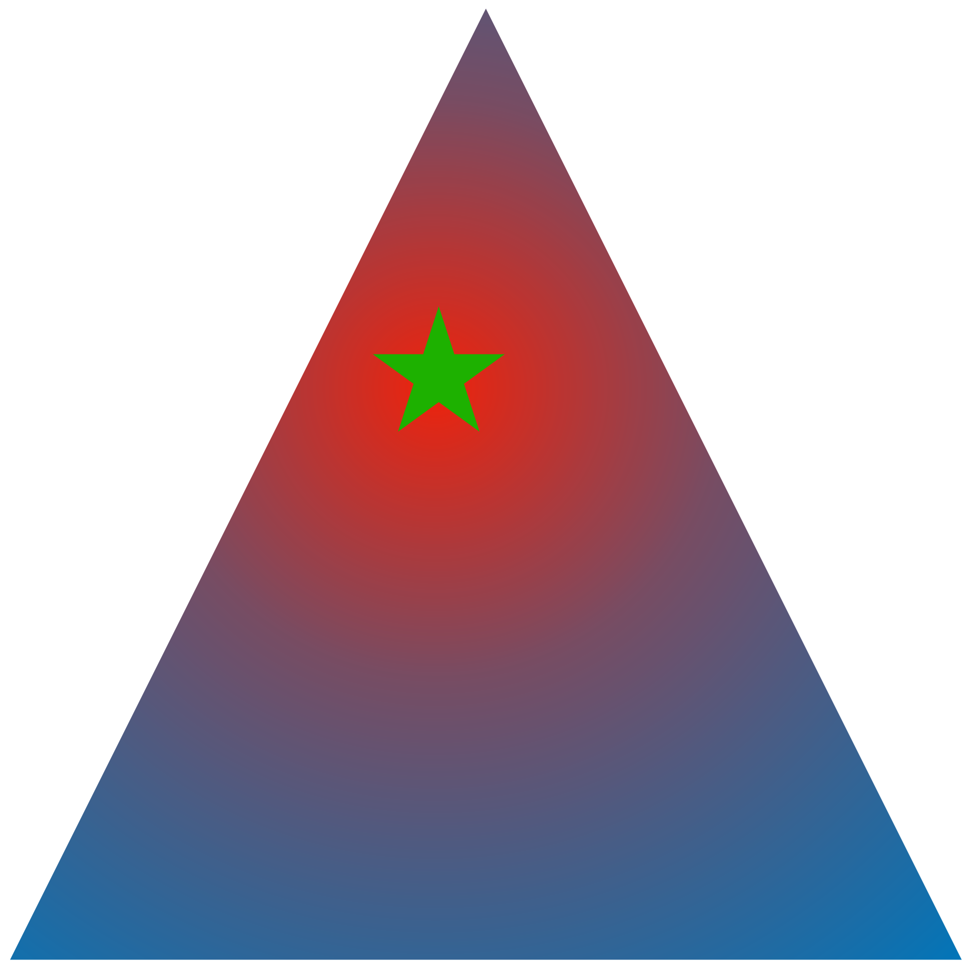
1. Perturb θ with ε
2. Find $\arg\max$

Beyond Softmax

- We can consider any strongly convex function $H(\,\cdot\,)$

$$\text{sparse max}(w) = \underset{z \in \Delta^d}{\text{argmax}}\left(w^T z - \frac{\|z\|^2}{2}\right)$$

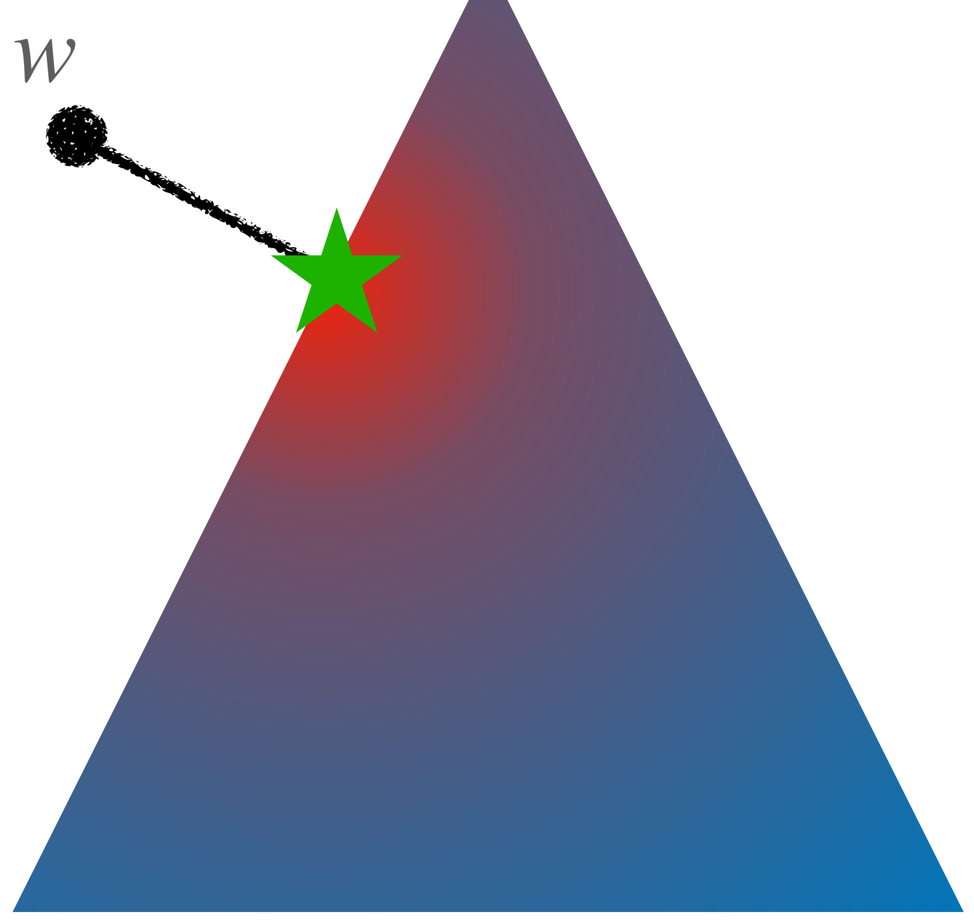
soft max :



- Finds sparse vectors:

    - $w^T z - \dfrac{\|z\|^2}{2} = \|w - z\|^2 + const$

soft max :



Martins A., Astudillo R. From softmax to sparsemax: A sparse model of attention and multi-label classification

The Limitation of GST

- Time is $O(d)$
  - Perturb each θ_i and find max
- For combinatorial z the support is $d \gg 1$
  - Gumbel softmax trick is too slow

Stochastic Softmax Tricks

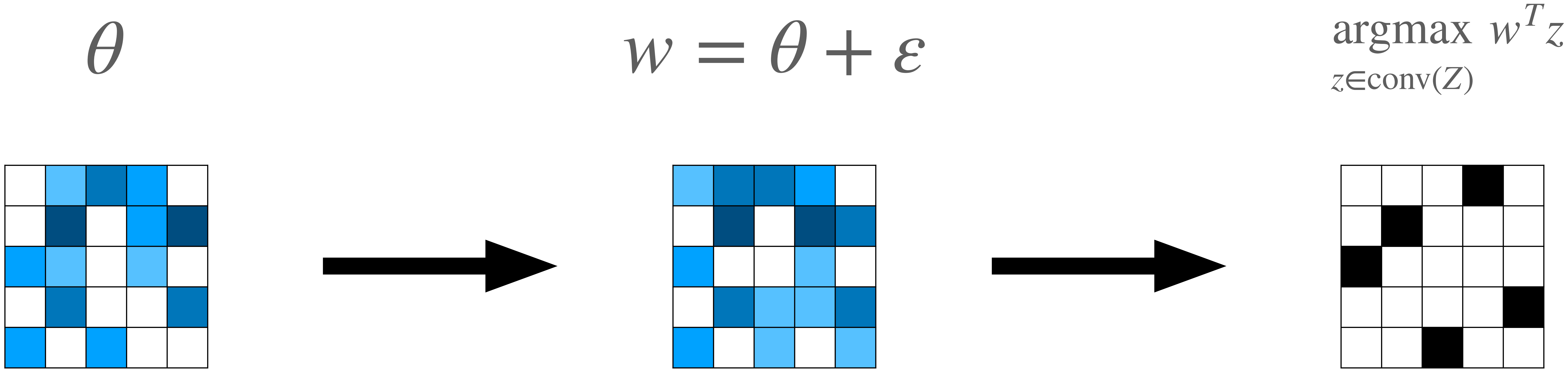
Paulus, M. B., Choi, D., Tarlow, D., Krause, A., & Maddison, C. J. Gradient Estimation with Stochastic Softmax Tricks.

Stochastic Max Trick

	Gumbel Argmax Trick	Stochastic argmax Trick
Support	$Z = \{e_1, \ldots, e_d\} \subset \mathbb{R}^d$	$Z = \{z_1, \ldots, z_m\} \subset \mathbb{R}^d$
Perturbation	$w = \theta_i - \log(-\log(u_i)), u \sim U[0,1]^d$	$w = r_\theta(\varepsilon)$
Forward pass	$z = \operatorname{argmax}_{z' \in \operatorname{conv} Z} w^T z'$	$z = \operatorname{argmax}_{z' \in \operatorname{conv} Z} w^T z'$

Example

- $Z = \{z_1, \dots, z_m\} \subset R^{n \times n}$ is a set of permutation matrices on n elements
- $m = n!$



Regularization & Relaxation

Stochastic *Arg*max → Stochastic *Soft*max

- Add a strongly convex regularizer $f : \mathbb{R}^d \to \{\mathbb{R}, \inf\}$

$$z = \underset{z' \in \mathrm{conv}(Z)}{\mathrm{argmax}}\ w^T z' \quad \to \quad z_T = \underset{z' \in \mathrm{conv}(Z)}{\mathrm{argmax}}\ (w^T z' - Tf(z'))$$

Prop 1. If z is a.s. unique, then $\lim_{T \to 0} z_T = z$

Prop 2. z_T exists, is unique and differentiable in w

Backpropagation

- Unroll the optimizer
- Write custom backward pass
- Use finite difference approximation

$$\frac{\mathrm{d}\mathscr{L}(z_t)}{\mathrm{d}w} \approx \frac{z_t(w + \epsilon\partial\mathscr{L}(z_t)/\partial z_t) - z_t(w - \epsilon\partial\mathscr{L}(z_t)/\partial z_t)}{2\epsilon}$$

- $z_t(w + \epsilon\partial\mathscr{L}(z_t)/\partial z_t) = z_t + \epsilon\frac{\partial\mathscr{L}}{\partial z_t}\frac{\mathrm{d}z_t}{\mathrm{d}w} + o(\epsilon)$

Domke J. Implicit differentiation by perturbation

Framework Requirements

- Inference
  - Reparametrized r.v. $w = r_\theta(\varepsilon)$
  - Solver for $\underset{z \in \operatorname{conv} Z}{\operatorname{argmax}}\, w^T z$
- Training
  - Strongly convex regularizer $f(z)$
  - Solver for $\underset{z \in \operatorname{conv} Z}{\operatorname{argmax}}(w^T z - t f(z))$

How to choose regularizer f?

- Need to solve $\underset{z \in \operatorname{conv} Z}{\operatorname{argmax}}(w^T z - tf(z))$
  - Euclidian projection $f(z) = \frac{\|z\|^2}{2}$
  - Neg. entropy $f(z) = \sum z_i \log z_i$ or $f(z) = \sum_i (z_i \log z_i - (1 - z_i)\log(1 - z_i))$
  - Exponential Families (next slides)

Exponential Family Reminder

- Consider $q(z \mid \theta) \propto I(z \in Z) \cdot \exp(z^t \theta)$
- Typical tasks:
  - MAP Inference: $\text{argmax}_{z \in Z} \theta^T z$
  - Find log-partition: $A(\theta) = \log \sum_{z \in Z} exp(\theta^T z)$
- Nice trick: $\frac{\partial A(\theta)}{\partial \theta} = \mathbb{E}_{q(z|\theta)} z$

- Ex.: Categorical
  - $Z = \{e_1, \ldots, e_d\}$
  - $A(\theta) = \log \sum \exp \theta_i$
  - $\frac{\partial A(\theta)}{\partial \theta} = \text{soft}\max(\theta)$

Wainwright M. J., Jordan M. I. Graphical models, exponential families, and variational inference.

Exponential Family Regularizer

- Consider $A(\theta) = \log \sum_{z \in Z} exp(\theta^T z)$
- Take $f(z) = A^*(z) = \sup_{\theta \in \Omega}(\theta^T z - A(\theta))$
- Softmax Trick:
    - $\underset{z \in \mathrm{conv}(Z)}{\mathrm{argmax}} \, (w^T z - f(z)) = \nabla_w A(w) = \mathbb{E}_{q(z|w)} z$

- Ex.: Gumbel Softmax
    - $A^*(z) = \sum_i z_i \log z_i$
    - $w = \theta - \log(-\log u))$
    - $\frac{\partial A}{\partial \theta} = \mathrm{soft\,max}(w)$

Wainwright M. J., Jordan M. I. Graphical models, exponential families, and variational inference.

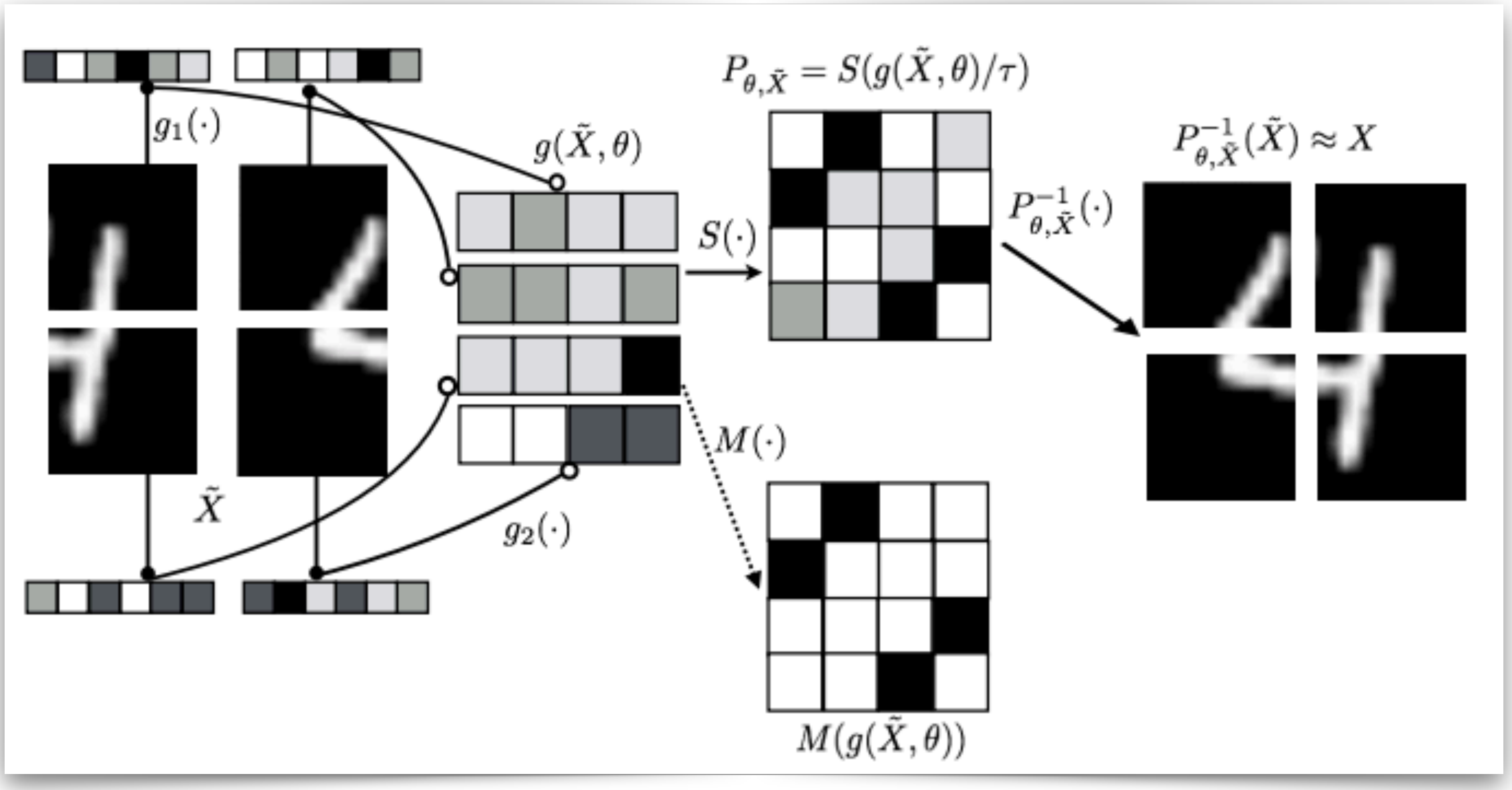
Applications

Gumbel Sinkhorn (1/4)

- Take permutation matrices as Z
  - Then $\mathrm{conv}(Z)$ consists of doubly-stochastic
- **Hungarian algorithm** solves $\arg\max w^T z$
- Entropy $f(z) = \sum_{i,j} z_{i,j} \log z_{i,j}$
- **Sinkhorn algorithm** finds $\arg\max(w^T z - T \cdot f(z))$

Mena G. et al. Learning latent permutations with gumbel-sinkhorn networks

Finding Latent Permutations: Jigsaw



Finding Latent Permutations: Connectomes

- Synthetic data $Y_t = PWP^T Y_{t-1} + \nu_t$
- Recover W

Prop. known neurons	40.%		30.%		20.%		10.%	
Difficulty	Easy	Hard	Easy	Hard	Easy	Hard	Easy	Hard
MCMC	.85	.82	.51	.44	.29	.27	.16	.12
(Linderman et al., 2017)	**.97**	.95	.90	**.85**	**.77**	**.59**	.39	.21
Gumbel-Sinkhorn	**.97**	**.96**	**.92**	.84	.76	**.59**	**.44**	**.26**
Gumbel-Sinkhorn, no regularization	.96	.93	.89	.78	.71	.52	.4	.23

k-subset Selection (2 / 4)

- $Z = \{z \in \{0,1\}^d \mid \sum z_i = k\}$

- Sort to solve $\arg\max w^T z$

- $Z = \{z \in \{0,1\}^{2d-1} \mid \sum_{i=1}^{n} z_i = k, z_i = z_{i-d} z_{i-d+1}$ **for** $d < i < 2d-1\}$

- Dynamic programming for $\arg\max$

- Exponential family relaxation

BeerAdvocate Interpretability

Pours a slight tangerine orange and straw yellow. The head is nice and bubbly but fades very quickly with a little lacing. Smells like Wheat and European hops, a little yeast in there too. There is some fruit in there too, but you have to take a good whiff to get it. The taste is of wheat, a bit of malt, and a little fruit flavour in there too. Almost feels like drinking Champagne, medium mouthful otherwise. Easy to drink, but not something I'd be trying every night.

Appearance: 3.5 Aroma: 4.0 Palate: 4.5 Taste: 4.0 Overall: 4.0

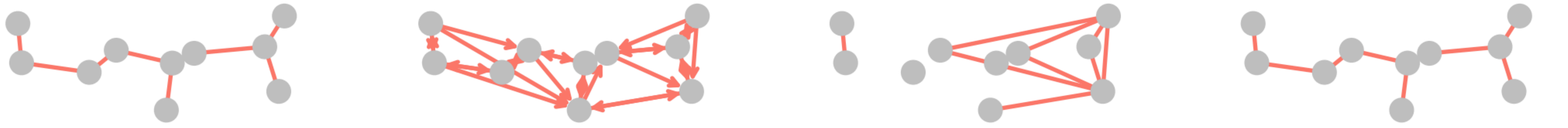
		$k = 5$		$k = 10$		$k = 15$	
Model	Relaxation	MSE	Subs. Prec.	MSE	Subs. Prec.	MSE	Subs. Prec.
Simple	*L2X* [17]	3.6 ± 0.1	28.3 ± 1.7	3.0 ± 0.1	25.5 ± 1.2	2.6 ± 0.1	25.5 ± 0.4
	SoftSub [84]	3.6 ± 0.1	27.2 ± 0.7	3.0 ± 0.1	26.1 ± 1.1	2.6 ± 0.1	25.1 ± 1.0
	Euclid. Top k	3.5 ± 0.1	25.8 ± 0.8	2.8 ± 0.1	32.9 ± 1.2	2.5 ± 0.1	29.0 ± 0.3
	Cat. Ent. Top k	3.5 ± 0.1	26.4 ± 2.0	2.9 ± 0.1	32.1 ± 0.4	2.6 ± 0.1	28.7 ± 0.5
	Bin. Ent. Top k	3.5 ± 0.1	29.2 ± 2.0	2.7 ± 0.1	33.6 ± 0.6	2.6 ± 0.1	28.8 ± 0.4
	E.F. Ent. Top k	3.5 ± 0.1	28.8 ± 1.7	2.7 ± 0.1	32.8 ± 0.5	2.5 ± 0.1	29.2 ± 0.8
	Corr. Top k	$\mathbf{2.9 \pm 0.1}$	$\mathbf{63.1 \pm 5.3}$	$\mathbf{2.5 \pm 0.1}$	$\mathbf{53.1 \pm 0.9}$	$\mathbf{2.4 \pm 0.1}$	$\mathbf{45.5 \pm 2.7}$

Latent Spanning Trees (3 / 4)

- Take adjacency matrices of undirected trees as $Z \subset \mathbb{R}^{d \times d}$
- Consider $q(z \mid \theta) \propto I(z \in Z) \cdot \exp(\theta^T z)$
- **Kruskal algorithm** for $\arg\max w^T z$
- **Kirchhoff's theorem** computes $A(\theta) = \log \sum_{z \in Z} \exp(\theta^t z)$
- Use $\frac{\partial A}{\partial \theta}$ as the relaxed matrix

Koo T. et al. Structured prediction models via the matrix-tree theorem

Dynamic Reconstruction



Ground Truth | *Indep. Directed Edges* | *E.F. Ent. Top* $|V|-1$ | *Spanning Tree*

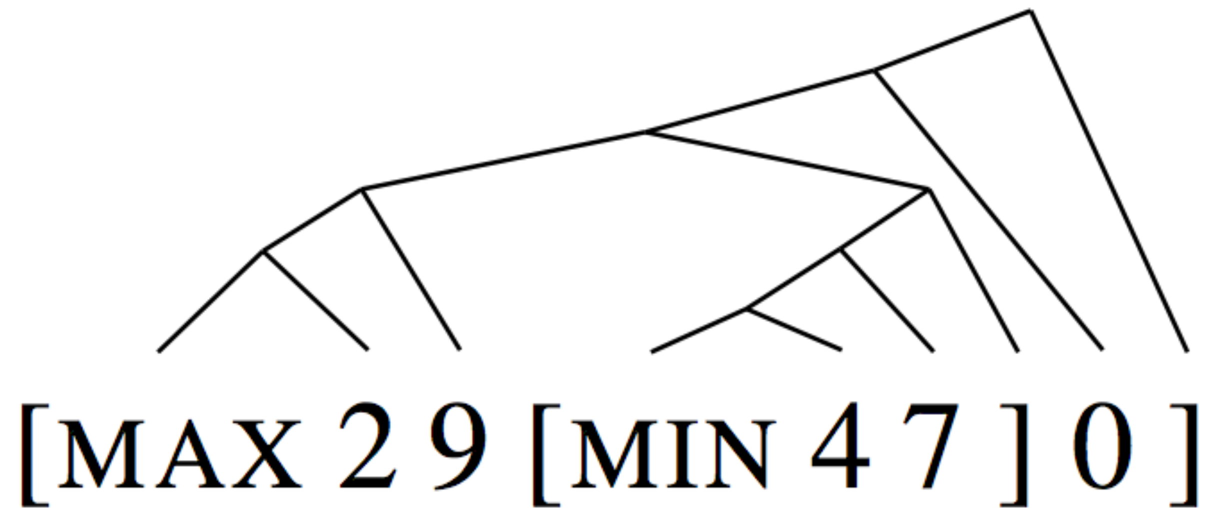
Edge Distribution	$T=10$ ELBO	$T=10$ Edge Prec.	$T=10$ Edge Rec.	$T=20$ ELBO	$T=20$ Edge Prec.	$T=20$ Edge Rec.
Indep. Directed Edges [38]	-1370 ± 20	48 ± 2	$\mathbf{93 \pm 1}$	-1340 ± 160	97 ± 3	$\mathbf{99 \pm 1}$
E.F. Ent. Top $\|V\|-1$	-2100 ± 20	41 ± 1	41 ± 1	-1700 ± 320	98 ± 6	98 ± 6
Spanning Tree	$\mathbf{-1080 \pm 110}$	$\mathbf{91 \pm 3}$	91 ± 3	$\mathbf{-1280 \pm 10}$	$\mathbf{99 \pm 1}$	$\mathbf{99 \pm 1}$

Latent Arborescence (4 / 4)

- Take adjacency matrices of directed rooted trees as $Z \subset \mathbb{R}^{d \times d}$
- Consider $q(z \mid \theta) \propto I(z \in Z) \exp(\theta^T z)$
- **Edmonds algorithm** for $\arg\max w^T z$
- **Kirchhoff's theorem** computes $A(\theta) = \log \sum_{z \in Z} \exp(\theta^t z)$
- Use $\frac{\partial A}{\partial \theta}$ as the relaxed matrix

Koo T. et al. Structured prediction models via the matrix-tree theorem

Unsupervised Parsing



- Corro & Titov relax Einser algorithm by replacing $\arg\max$ with $\operatorname{soft}\max$
- Simplified ListOps dataset

Model	Edge Distribution	Task Acc.	Edge Precision	Edge Recall
LSTM	—	92.1 ± 0.2	—	—
GNN on latent graph	*Indep. Undirected Edges*	89.4 ± 0.6	20.1 ± 2.1	45.4 ± 6.5
	Spanning Tree	91.2 ± 1.8	33.1 ± 2.9	47.9 ± 5.2
GNN on latent digraph	*Indep. Directed Edges*	90.1 ± 0.5	13.0 ± 2.0	56.4 ± 6.7
	Arborescence			
	- Neg. Exp.	71.5 ± 1.4	23.2 ± 10.2	20.0 ± 6.0
	- Gaussian	$\mathbf{95.0 \pm 2.2}$	65.3 ± 3.7	60.8 ± 7.3
	- Gumbel	$\mathbf{95.0 \pm 3.0}$	$\mathbf{75.5 \pm 7.0}$	$\mathbf{71.9 \pm 12.4}$
	Ground Truth Edges	98.1 ± 0.1	100	100

Corro C., Titov I. Differentiable perturb-and-parse: Semi-supervised parsing with a structured variational autoencoder
Nangia N., Bowman S. R. Listops: A diagnostic dataset for latent tree learning

Conclusions

- Learning structured latent variables is an active research direction
- *Stochastic softmax trick* generalize *gumbel softmax trick*
- Latent structure works as an *inductive bias*
- There is more to be done