Local Identifiability Problem

Motivation

The "flat tire" problem: Imagine you're trying to optimize a function, but you're driving on a perfectly flat parking lot. No matter which direction you turn the steering wheel, you don't move—there's no gradient to follow. This is exactly what happens with hard box embeddings when boxes are disjoint or contained: the loss function is flat, and gradient descent has nowhere to go.

The thought experiment: Consider two boxes that are completely separate—say, one box represents "cats" and another represents "dogs". With hard boxes, if they're disjoint, the intersection volume is exactly zero. Now, if you try to move the "cats" box slightly closer to the "dogs" box, but they're still disjoint, the intersection volume remains zero. The gradient is zero—no learning signal! The optimizer is stuck, unable to distinguish between "cats box at position A" and "cats box at position B" as long as both keep the boxes disjoint.

Gumbel boxes solve this elegantly by replacing deterministic boundaries with probabilistic ones. Even when boxes are "disjoint" in expectation, there is always some probability of overlap (the boundaries can "wiggle" into each other). This ensures that every parameter configuration produces a distinct expected loss—the loss landscape is never completely flat. The optimizer always has a gradient to follow, even if it's small. It's like replacing the flat parking lot with a gentle slope: you can always tell which way is "downhill".

Historical development: The evolution from hard boxes to Gumbel boxes was motivated by this exact problem. Initial work on box embeddings used hard boxes (deterministic intervals), which suffered from zero gradients when boxes were disjoint or contained. The first solution was "smoothed boxes" using Gaussian noise, but this broke the max-stability property—intersections of Gaussian boxes are not Gaussian. Gumbel boxes were chosen specifically because they combine smoothness (probabilistic boundaries) with max-stability (intersections remain Gumbel), providing both differentiability and mathematical tractability. This is why Gumbel distributions, rather than other smooth distributions, are used in box embeddings.

Connection to other smoothing techniques: The local identifiability problem is a fundamental challenge in machine learning, appearing in many contexts beyond box embeddings. Common solutions include: (1) **Gaussian smoothing** (add Gaussian noise), which breaks mathematical structure; (2) **temperature annealing** (as in Gumbel-Softmax), which interpolates between hard and soft; (3) **entropy regularization**, which encourages exploration; and (4) **reparameterization tricks** (as in variational autoencoders), which make discrete variables differentiable. Gumbel boxes use a combination of these: Gumbel noise provides smoothing, the scale parameter β acts as a temperature, and the max-stability property ensures mathematical consistency. This makes Gumbel boxes a particularly elegant solution that preserves both differentiability and mathematical structure.

Definition

The **local identifiability problem** arises when multiple distinct parameter configurations produce identical loss values, creating flat regions in the loss landscape where the gradient $\nabla_\theta L(\theta) = 0$. This prevents learning because gradient descent has no direction to follow—all parameter values in the flat region yield the same loss, so the optimizer cannot distinguish between them.

Statement

Theorem (Local Identifiability Problem). For hard boxes, the loss landscape has flat regions with zero gradients:

1. **Disjoint boxes**: When boxes A and B are completely disjoint, $\text{Vol}(A \cap B) = 0$ regardless of their separation distance. Any local perturbation preserving disjointness yields zero gradient.
2. **Contained boxes**: When box B is fully contained in box A, small perturbations preserving containment produce identical loss values, creating zero-gradient regions.

Theorem (Gumbel Solution). By modeling coordinates as Gumbel random variables, the expected volume computation involves all parameters continuously. Let θ_A and θ_B denote the location parameters of boxes A and B, and let ε_A and ε_B be the Gumbel random variables (the "noise" terms). Then:

$$E[\text{Vol}(A \cap B)] = \int \int \text{Vol}(A(\theta_A, \varepsilon_A) \cap B(\theta_B, \varepsilon_B)) \, \mathrm{d}P(\varepsilon_A) \, \mathrm{d}P(\varepsilon_B)$$

This ensemble perspective (averaging over all possible realizations of the Gumbel noise) ensures that different parameter configurations θ_A, θ_B produce different expected loss values, restoring local identifiability. The gradient $\nabla_{\theta_A, \theta_B} E[\text{Vol}(A \cap B)]$ is non-zero for all parameter values.

Proof

Hard boxes produce zero gradients in two critical cases:

1. **Disjoint boxes**: When boxes A and B are completely disjoint, $\text{Vol}(A \cap B) = 0$ regardless of their separation distance. Any local perturbation that preserves disjointness yields zero gradient, creating a flat region in parameter space.
2. **Contained boxes**: When box B is fully contained in box A, small perturbations that preserve containment produce identical loss values. The optimizer cannot distinguish between different parameter configurations that all satisfy the containment constraint.

Gumbel boxes solve this fundamental problem through three mechanisms:

1. **Expected volumes are always positive**: Even when boxes are "disjoint" in the sense that their expected boundaries don't overlap, $E[\text{Vol}(A \cap B)] > 0$ due to the probabilistic nature of the boundaries. The tails of the Gumbel distributions ensure some probability of overlap.

   Quantitative bound: For disjoint boxes with separation distance d (measured between expected boundaries), we have $E[\text{Vol}(A \cap B)] \geq Ce^{-\frac{d}{\beta}}$ for some constant $C > 0$. This exponential decay ensures that the expected volume is always positive, guaranteeing a non-zero gradient signal. The exponential decay rate $\frac{1}{\beta}$ means that as β decreases (boxes become "sharper"), the overlap probability decreases exponentially, but never reaches zero. This is a fundamental property of Gumbel distributions: they have infinite support, so there's always some probability mass in any region.
2. **Gradients are dense**: All parameters contribute to the expected volume through the Bessel function formula (see the Gumbel-Box Volume document). The smooth dependence on

parameters via $K_0\left(2e^{-\frac{\mu_y-\mu_x}{2\beta}}\right)$ ensures that the gradient $\nabla_\theta E[\mathrm{Vol}(A\cap B)]$ is non-zero for all parameter values θ. Specifically, $\frac{\partial}{\partial\mu}E[\mathrm{Vol}]\neq 0$ for all μ and β. The Bessel function K_0 is smooth and differentiable everywhere, and its derivative with respect to its argument is non-zero (except at infinity), ensuring that changes in location parameters always produce changes in expected volume.

3. **Smooth loss landscape**: The probabilistic formulation eliminates flat regions entirely. The loss function $L(\theta) = -\log E[\mathrm{Vol}(A\cap B)]$ becomes a smooth function of the parameters, enabling effective gradient-based optimization. The Bessel function provides smooth, differentiable gradients everywhere. Unlike hard boxes where the loss function has discontinuities (sudden jumps from zero to positive volume), Gumbel boxes produce a loss landscape that is infinitely smooth (smooth of all orders), allowing gradient descent to navigate the entire parameter space without getting stuck in flat regions.

The "stuck optimizer" problem and its solution: Let's see exactly what happens with hard boxes versus Gumbel boxes when boxes are far apart.

Hard boxes (disjoint) — the problem:
- Box A: $[0.0, 0.0]$ to $[0.3, 0.3]$ (a box in the lower-left corner)
- Box B: $[0.7, 0.7]$ to $[1.0, 1.0]$ (a box in the upper-right corner)
- Separation distance: $d = 0.4$ (they're well separated)
- Intersection volume: 0 (they don't overlap at all)
- Loss gradient: $\nabla_\theta L = 0$ (flat region—the optimizer is stuck!)

The puzzle: What if you want to move Box A slightly to the right? The intersection volume stays at zero (they're still disjoint), so the gradient is still zero. The optimizer can't tell the difference between Box A at position $(0.0, 0.0)$ and Box A at position $(0.01, 0.0)$—both give the same loss. This is the local identifiability problem in action.

Gumbel boxes (disjoint, with $\beta = 0.1$) — the solution:
- Expected intersection volume: $E[\mathrm{Vol}(A\cap B)] > 0$ (small but positive!)
- The magic number: approximately $Ce^{-\frac{0.4}{0.1}} = Ce^{-4} \approx 0.018C$ for some constant C
- Even though the boxes are far apart, the probabilistic boundaries create a tiny overlap probability
- Loss: $L = -\log E[\mathrm{Vol}(A\cap B)]$ is finite and differentiable (not infinite, not zero)
- Gradient: $\nabla_\theta L \neq 0$ (non-zero everywhere—the optimizer can learn!)

The "aha!" moment: The gradient points in the direction that increases intersection volume. Even when boxes are far apart, moving them closer together increases the (tiny) expected overlap, creating a learning signal. The probabilistic formulation ensures the loss landscape has no flat regions—it's like replacing a flat desert with rolling hills. You can always find a direction to go.