A Mathematical Analysis of Learning Loss for Active Learning in Regression

Megh Shukla , Shuaib Ahmed

Mercedes-Benz Research and Development India

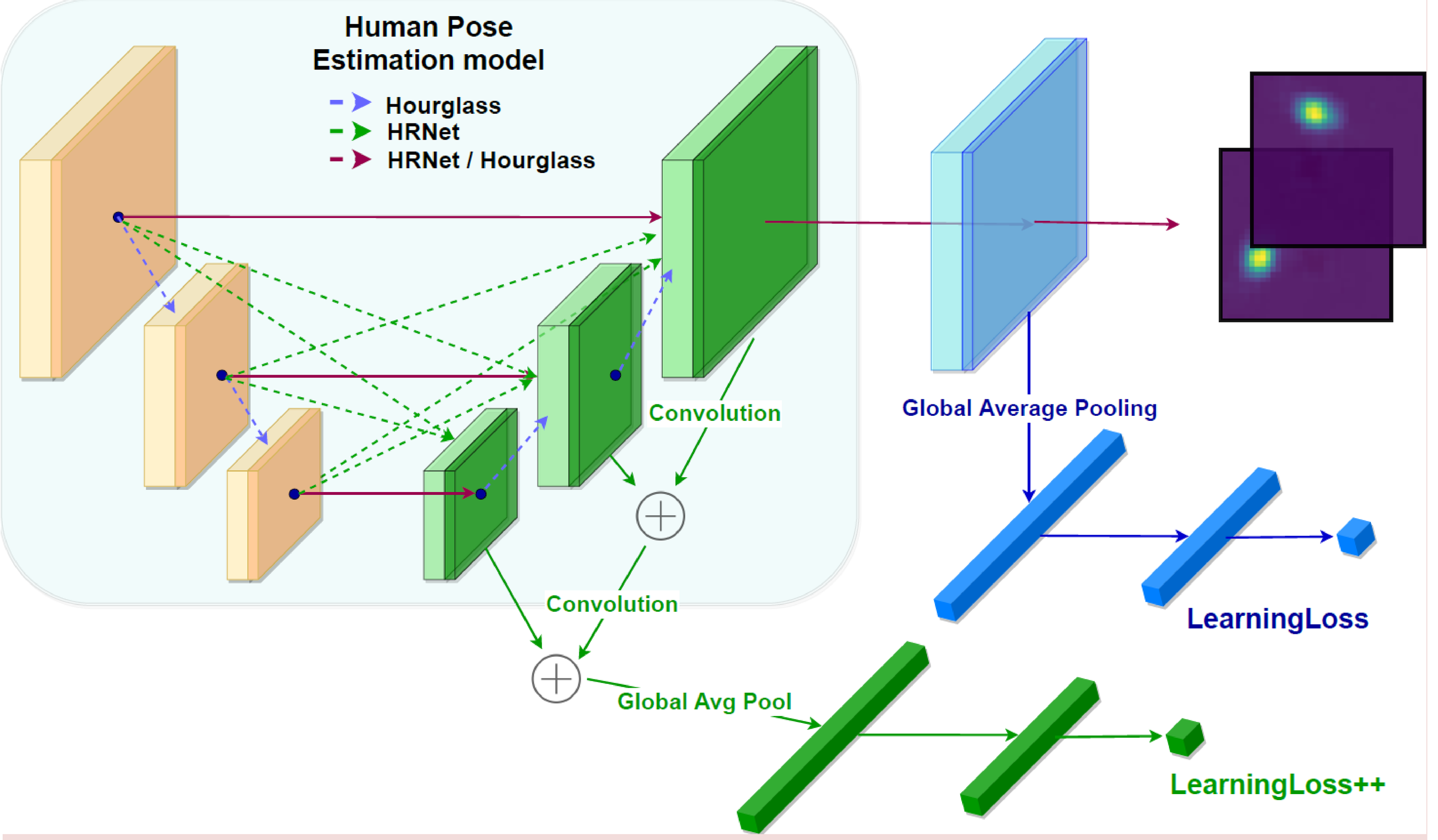
Mercedes-Benz

CVPR VIRTUAL JUNE 19-25

Active Learning for continuous model refinement?

1. **Can we recognize real world failures on-the-fly, allowing for continuous model refinement?**
2. ***LearningLoss++*:** A mathematical evolution of Learning Loss to better identify failure use cases

Learning Loss



$$\mathbb{L}_{loss} = \max\Big(0, -\,\mathrm{sign}(\underbrace{l_i - l_j}_{\text{True Loss}})(\underbrace{\hat{l}_i - \hat{l}_j}_{\text{Predicted}}) + \underbrace{\xi}_{\text{Margin}}\Big)$$

Training: Compares the true and predicted losses for a pair of images; predicted loss margin tries to ensure minimum separation between the predicted loss pair

Drawbacks:

1. The objective lacks rigorous theoretical analysis
2. The margin hyperparameter leads to exploding weights

LearningLoss++

We show that a KL divergence based objective is equivalent to the original empirical formulation:

$$\mathbb{L}_{loss}(w, \theta_i, \theta_j) = \mathrm{KL}(p||q) = p_i \log\frac{p_i}{q_i} + p_j \log\frac{p_j}{q_j}$$

1. Learning Loss gradient:

$$\nabla_w \mathbb{L}_{loss} \in \{0, \pm(\theta_i - \theta_j)\}$$
$$\nabla_\theta \mathbb{L}_{loss} \in \{0, \pm w\}$$

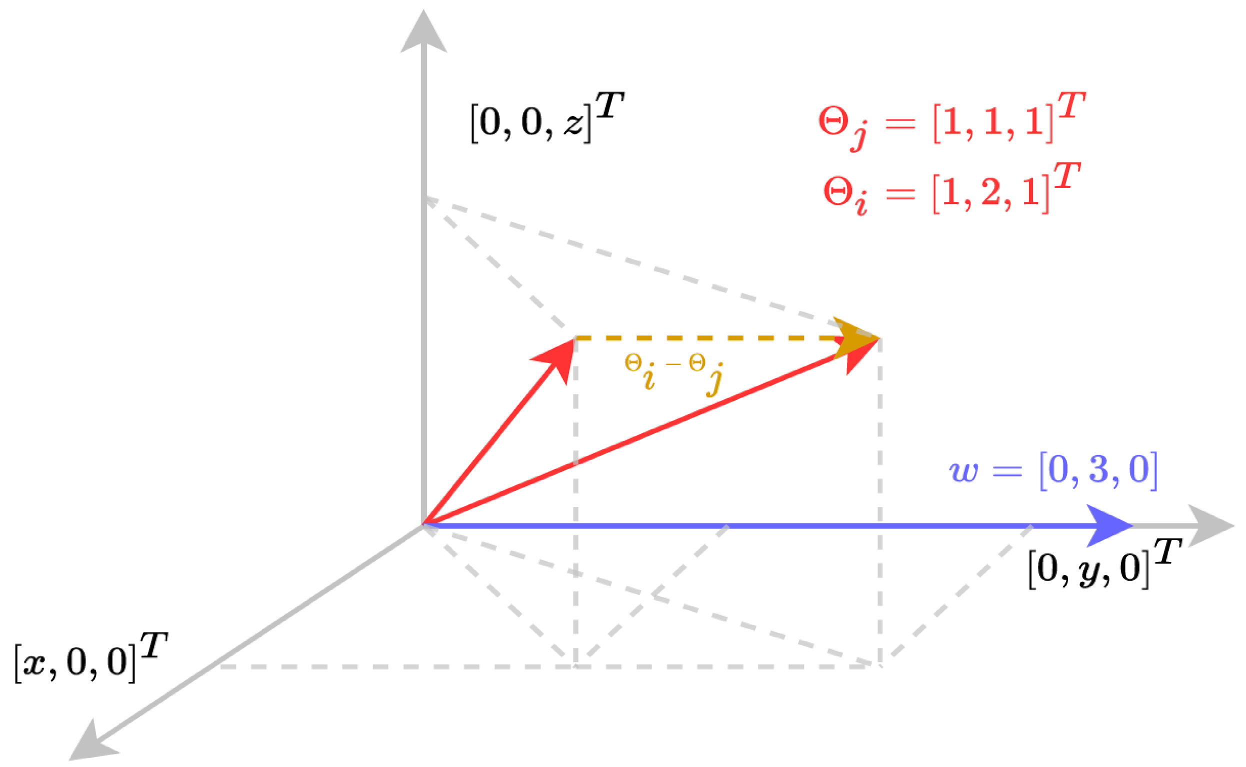
2. LearningLoss++ gradient:

$$\nabla_w \mathbb{L}_{loss}(w, \theta_i, \theta_j) = (q_i - p_i)\underbrace{(\theta_i - \theta_j)}_{\text{Embedding for images } i \text{ and } j}$$
$$\nabla_\theta \mathbb{L}_{loss}(w, \theta_i, \theta_j) = (q_i - p_i) w$$

LearningLoss++ introduces a smoothness to the objective, absorbing the predicted loss margin!

Analysis

How does Learning Loss learn to predict losses?



1. The **weight vector aligns along discriminatory components** to explain the predicted losses for an embedding pair
2. Role of the predicted loss margin: If the predicted losses are not sufficiently separated, the **margin causes repeated updates to the weight vector** increasing its norm.

Analysis

Regression analysis allows us to model the true loss with a Gamma distribution

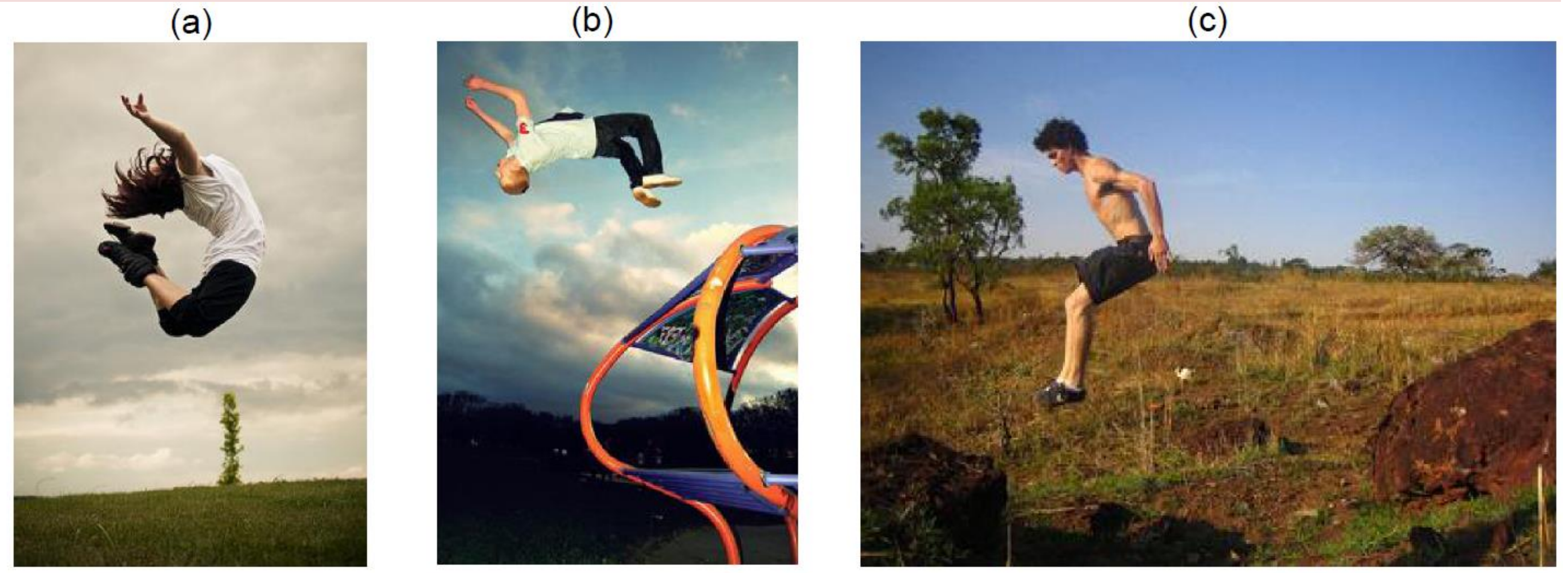
What is the probability of sampling two images with similar true losses? *(Eq: 1)*

$$P(|X - Y| \le \delta) = \int_0^\delta \gamma(x,k,\Theta)\int_0^{x+\delta}\gamma(y,k,\Theta)\,\mathrm{d}y\mathrm{d}x + \int_\delta^\infty \gamma(x,k,\Theta)\int_{x-\delta}^{x+\delta}\gamma(y,k,\Theta)\,\mathrm{d}y\mathrm{d}x$$

What is the expected gradient for a given true loss separation? *(Eq: 2)*

$$\mathbb{E}_{x,y|\delta_2}\Big[\nabla_w \mathbb{L}(w,\theta_i,\theta_j)\Big] = \lim_{\delta_1 \to \delta_2}\int_{x=0}^{x=\infty}\int_{y=x+\delta_1}^{y=x+\delta_2}\Big(q_i - \frac{x}{2x+\delta_2}\Big)(\theta_i - \theta_j)\frac{\gamma(x,k,\Theta)\gamma(y,k,\Theta)}{p(y - x = \delta_2)}$$

Do the above solutions have a closed form solution? Yes, assuming $k \in \mathbb{Z}^+$!
Can we use these closed form solutions to gain better insights?



	(a)	(b)	(c)
True Loss: l	0.45	0.47	2.87
Pred Loss: $\hat{l}$	1.48	1.51	1.63
Image Pairs(i,j)		$(i=(a), j=(b))$	$(i=(a), j=(c))$
$\|\|\nabla_w(l_i, l_j, \hat{l}_i, \hat{l}_j)\|\|$: (LearningLoss)		$\|\|\theta_i - \theta_j\|\|$	$\|\|\theta_i - \theta_j\|\|$
$\|\|\nabla_w(l_i, l_j, \hat{l}_i, \hat{l}_j)\|\|$: (LearningLoss++)		≈ 0	$0.32\|\|\theta_i - \theta_j\|\|$

Case 1: The true and predicted losses are similar for a pair of images

A sufficiently trained Learning Loss network goes against intuition and imposes a penalty for a non trivial number of image pairs with true loss margin <= δ (Tab: 1)

Case 2: The predicted losses are similar, but the true losses are vastly different.

While the gradient response for Learning Loss is constant, LearningLoss++ gradient reflects the mismatch in true and predicted loss separation (Tab: 2)

Result

LearningLoss++ (right) has a higher degree of correlation with the true loss, consistently identifying *lossy* images in comparison to Learning Loss (left).

Why use LearningLoss++?

1. Rigorous analysis for better explainability
2. Recognize real world failures on-the-fly!
3. Eliminates the margin hyperparameter!
4. The revised objective results in a smoother gradient to identify *lossy* images

(Table: 1)

δ	0.02	0.04	0.06	0.08	0.1	0.125	0.15
$P_{X,Y,\gamma}$	0.094	0.185	0.274	0.358	0.437	0.527	0.607

(Table: 2)

$\delta \rightarrow$	0.0	0.1	0.2	0.3	0.4	0.5
LL++	q_i-0.5	q_i-0.39	q_i-0.3	q_i-0.25	q_i-0.21	q_i-0.18
LL	$\leftarrow$ constant c_1 $\rightarrow$					