Do as your neighbors: Invariant learning through non-parametric neighbourhood matching

Andrei Nicolicioiu[1,2] Jerry Huang[1,2] Dhanya Sridhar[1,2] Aaron Courville[1,2]
[1]Mila - Quebec AI Institute [2]Université de Montréal

Introduction

- Deep learning models can be accurate on standard in-domain data while generalizing poorly **out of distribution (OOD)**.
- **Invariance** has been proven to be an effective signal for learning **robust** predictors.
- Most methods are still *hard to train on real-world* data and are sensible to hyperparameter selection.

Contributions

- Propose a **method for invariance learning (iKNN)** based on the idea of domain invariance of non-parametric methods.
- iKNN is able to recover the **robust predictor** in *simple* standard settings.
- iKNN is somewhat **less sensitive** to hyperparameters - advantageous since hyperparameter search and model selection are very challenging in out-of-distribution settings.

Proposed method

Invariant KNN

Proposed method: Each environment has an associated KNN predictor:

$$\mu_A(X) = \frac{1}{k} \sum_{X_i \in \mathcal{N}_k^A(X)} \langle \phi(X), \phi(X_i) \rangle Y_i$$

$$\mu_B(X) = \frac{1}{k} \sum_{X_i \in \mathcal{N}_k^B(X)} \langle \phi(X), \phi(X_i) \rangle Y_i$$

Constrain the non-parametric (KNN) predictors associated with training domains to be similar:

$$\mathcal{L}_{\text{iKNN}} = \sum_X ||\mu_A(X) - \mu_B(X)||$$

Q: What assumptions do we need for?

$$\mathbb{P}_{\mathcal{E}_A}[\phi(X)|Y] = \mathbb{P}_{\mathcal{E}_B}[\phi(X)|Y]$$

Conditional Mean Embeddings

Using all points results in matching of mean-embeddings.

$$w_A^c = \frac{1}{|\mathcal{E}_A^c|} \sum_{X_i \in \mathcal{E}_A^c} \phi(X_i)$$

$$w_B^c = \frac{1}{|\mathcal{E}_B^c|} \sum_{X_i \in \mathcal{E}_B^c} \phi(X_i)$$

Constrain mean-embeddings of training domains to be similar:

$$\mathcal{L}_{\text{mean}} = \sum_c ||w_A^c - w_B^c||$$

Which has its minimum when:

$$\mathbb{E}_{\mathcal{E}_A}[\phi(X)|Y] = \mathbb{E}_{\mathcal{E}_B}[\phi(X)|Y]$$

Results

Table 1: iKNN performs comparably against other invariant methods on ColoredMNIST.

Algorithm	Test Spurious correlation		
	Strong (+) (ID)	Zero (OOD)	Strong (-) (OOD)
ERM	**88.4** ± 0.4	52.6 ± 0.2	17.5 ± 1.1
IRM	71.6 ± 2.3	70.0 ± 0.5	67.1 ± 2.1
V-REx	70.1 ± 1.3	**70.3** ± 0.7	69.0 ± 0.7
iKNN	68.7 ± 0.2	**70.1** ± 1.0	**70.1** ± 1.2

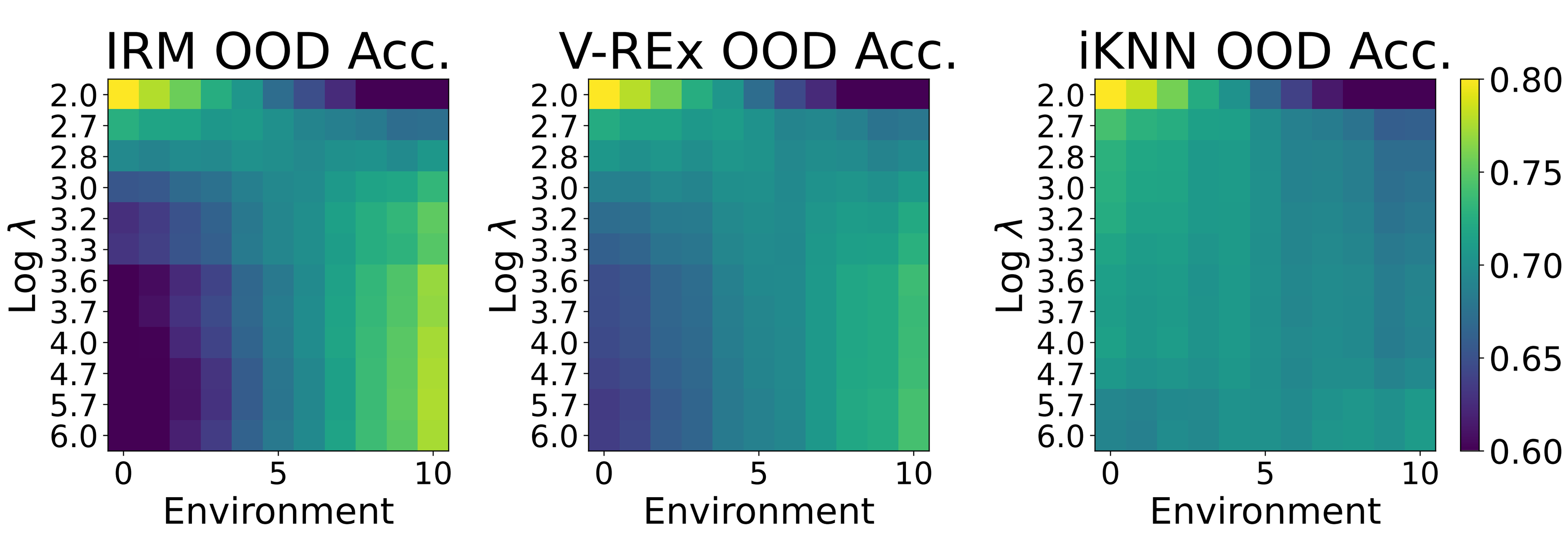

Figure 1: Performance of algorithms on cMNIST as the λ parameter increases on differing levels of OOD testing environments. Environments go from 0 (color-label correlation = 1.0) to 10 (color-label correlation=-1.0) with models trained on environments 1 and 2. A good invariant model has stable performance across all environments. We see that iKNN is generally more stable for a larger range of λ values.

Local vs Local constraint

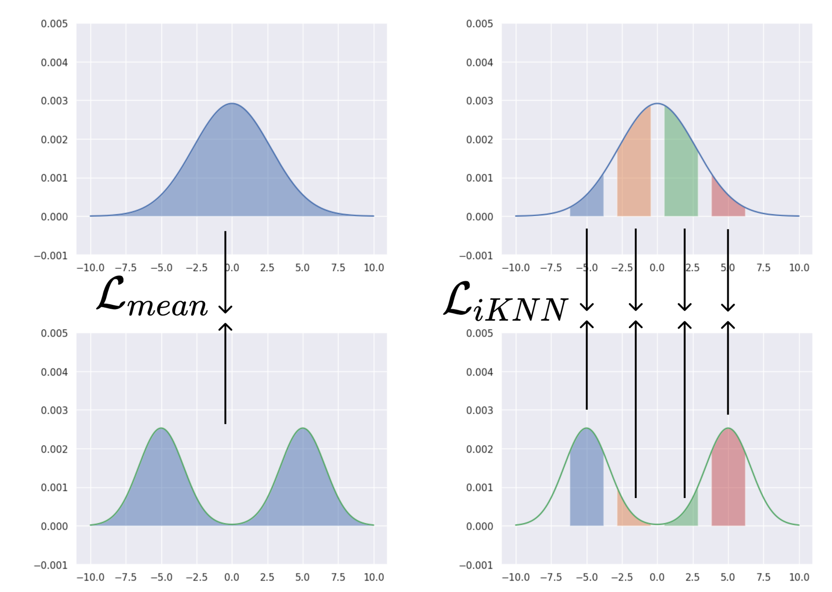

Figure 2: Our proposed $\mathcal{L}_{\text{iKNN}}$ vs the simplified version $\mathcal{L}_{\text{mean}}$. $\mathcal{L}_{\text{mean}}$ puts a constraint on the mean embeddings of features across the complete distributions, while $\mathcal{L}_{\text{iKNN}}$ constraints *local* means across all points in the two distributions. For the given distributions $\mathcal{L}_{\text{mean}}$ is already zero while $\mathcal{L}_{\text{iKNN}}$ is not, thus the distributions can be constrained further using $\mathcal{L}_{\text{iKNN}}$.

Invariance Penalty Sensitivity

In Figure 1 shows the performance across environments for different invariance penalty (λ) values. Methods not sensitive to λ have uniform performance across environments for a large range of λ. For this, we measure the standard deviation of the accuracy across the environments and average them across λ values.

Model	IRM	V-REx	iKNN
Avg std ↓	4.7	2.7	**1.8**

This shows that iKNN is less sensitive to choices of λ, and can work well under a larger range of λ values. Since choosing the right hyperparameters is extremely important in an out-of-distribution setting, this larger range is an important advantage.

Enforcing the same non-parametric predictor (KNN) across multiple domains leads to robust learning.

Take-aways

- **KNNs on multiple domains can be used to obtain robust predictions.**
- **Method is somewhat less sensitive to hyperparameter selection.**
- **More investigations are needed to fully understand its assumptions and to apply the method to real-world settings.**

Contact

E-mail: andrei.nicolicioiu@mila.quebec