Order Learning Using Partially Ordered Data via Chainization

Seon-Ho Lee[0000−0002−3844−7081] and Chang-Su Kim[0000−0002−4276−1831]

School of Electrical Engineering, Korea University, Korea
seonholee@mcl.korea.ac.kr, changsukim@korea.ac.kr

S-1 Algorithm Details

S-1.1 Networks

The pairwise comparator (in Fig. 3 in the main paper) consists of two components: a feature extractor and a classifier. The network structures of these components are detailed in Table S-1 and Table S-2, where '$k_h \times k_w$-s-c Conv' denotes the 2D convolution with kernel size $k_h \times k_w$, stride s, and c output channels. Similarly, '$k_h \times k_w$-s MaxPool' represents the 2D max pooling with a $k_h \times k_w$ kernel at stride s. Also, 'BN' means batch normalization [3], and 'c Dense' is a dense layer with c output channels. Note that the feature extractor is implemented based on the VGG16 network, and it takes a $224 \times 224 \times 3$ image as input.

S-1.2 MAP Estimation

Let us describe the MAP estimation rule for rank estimation in Section 3.5. Without loss of generality, we assume that the ranks (or classes) are the first m natural numbers, $\Theta = \{1, 2, \dots m\}$. We estimate the rank $\theta(x)$ of a test instance x by comparing it with the references y_i of known ranks $\theta(y_i) = i$, $1 \le i \le m$. Then, by comparing x with y_i, the comparator yields the probability vector $p^{xy_i} = (p^{xy_i}_{\succcurlyeq}, p^{xy_i}_{\preccurlyeq})$. Thus, given y_i, the probability of $\theta(x) = r$ can be written as

$$P_{\theta(x)}(r \,|\, y_i) = p^{xy_i}_{\succcurlyeq} \cdot P_{\theta(x)}(r \,|\, x \succcurlyeq y_i) + p^{xy_i}_{\preccurlyeq} \cdot P_{\theta(x)}(r \,|\, x \preccurlyeq y_i). \tag{1}$$

Suppose that $x \succcurlyeq y_i$. Then, $i \le \theta(x) \le m$, where m is the maximum possible rank. In other words, there are $m - i + 1$ possible ranks for $\theta(x)$, which are assumed to be equally likely. In other words, we assume that the conditional probability distribution of $P_{\theta(x)}(r \,|\, x \succcurlyeq y_i)$ is a uniform distribution over $[i, m]$. However, $\theta(x) = i$ belongs to both cases of $x \succcurlyeq y_i$ and $x \preccurlyeq y_i$. Therefore, we assume that the conditional probability $P_{\theta(x)}(r \,|\, x \succcurlyeq y_i)$ is given by

$$P_{\theta(x)}(r \,|\, x \succcurlyeq y_i) = \begin{cases} \frac{1}{m-i+0.5}, & \text{if } r > i, \\ \frac{1}{2(m-i+0.5)}, & \text{if } r = i, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Table S-1: The feature extractor in the pairwise comparator.

Layers	Output
3×3-1-64 Conv BN ReLU	224×224×64
3×3-1-64 Conv BN ReLU	224×224×64
3×3-2 MaxPool	112×112×64
3×3-1-128 Conv BN ReLU	112×112×128
3×3-1-128 Conv BN ReLU	112×112×128
3×3-2 MaxPool	56×56×128
3×3-1-256 Conv BN ReLU	56×56×256
3×3-1-256 Conv BN ReLU	56×56×256
3×3-1-256 Conv BN ReLU	56×56×256
3×3-2 MaxPool	28×28×256
3×3-1-512 Conv BN ReLU	28×28×512
3×3-1-512 Conv BN ReLU	28×28×512
3×3-1-512 Conv BN ReLU	28×28×512
3×3-2 MaxPool	14×14×512
3×3-1-512 Conv BN ReLU	14×14×512
3×3-1-512 Conv BN ReLU	14×14×512
3×3-1-512 Conv BN ReLU	14×14×512
14×14-1 MaxPool	1×1×512
512 Dense BN ReLU	512

Table S-2: The classifier in the pairwise comparator.

Layers	Output
512 Dense BN ReLU	512
512 Dense BN ReLU	512
Dropout(0.5)	512
2 Dense Softmax	2

We formulate $P_{\theta(x)}(r \,|\, x \preccurlyeq y_i)$ in a similar manner. Then, we approximate the *a posteriori* probability $P_{\theta(x)}(r \,|\, y_1, \ldots, y_m)$ by averaging the single-reference inferences in Eq. (1);

$$P_{\theta(x)}(r \,|\, y_1, \ldots y_m) = \tfrac{1}{m} \textstyle\sum_{i=1}^{m} P_{\theta(x)}(r \,|\, y_i). \tag{3}$$

Finally, we obtain the MAP estimate of the rank of x by

$$\hat{\theta}(x) = \arg\max_{r} P_{\theta(x)}(r \,|\, y_1, \ldots y_m). \tag{4}$$

S-1.3 Threshold τ for Pseudo Pair Sampling

Let us describe how to select the threshold τ in the pseudo pair sampling. By sorting all vertices of $\mathcal{V}$ via σ in Eq. (3) in the main paper, we obtain a chain of vertices. Then, we merge some vertices, which likely come from the same

underlying class, to obtain a shortened chain $(w_1, w_2, \ldots, w_c)$. Here, c denotes the number of vertices in the shortened chain.

To build the pseudo pair set $\mathcal{T}$, we sample an ordered pair (x, y) from the shortened chain, where $x \in w_i$, $y \in w_j$, and $j - i > \tau$. We define the sampling threshold τ as

$$\tau = \frac{c}{m} \tag{5}$$

where m denotes the number of underlying classes in $\mathcal{X}$. We assume that m is known *a priori*.

Note that the chainization algorithm sorts all instances in $\mathcal{X}$ and samples pseudo pairs iteratively. In the clique-edgeless case, we use different τ at different iterations t. This is because no ordering information within $\mathcal{X}_\mathrm{e}$ is known. Hence, at early iterations $t \leq 2$, we do not shorten the chain. Then, we use the sampling threshold τ, given by

$$\tau = \frac{|\mathcal{X}_\mathrm{e}|}{2^t}, \tag{6}$$

so that the threshold decreases as the iteration t goes on. Similarly, in the bipartite case, we set the sampling threshold τ_i for $\mathcal{X}_i, i = 0, 1$, as

$$\tau_i = \frac{|\mathcal{X}_i|}{2^t}. \tag{7}$$

S-1.4 Implementation Details

To initialize the feature extractor, we adopt the parameters pre-trained on the ILSVRC2012 dataset. We initialize the other layers using the Glorot normal method. We update the network parameters using the Adam optimizer with a minibatch size of 16. We start with a learning rate of 10^{-4} and shrink it by a factor of 0.8 every 10,000 steps. Training images are augmented by random horizontal flipping and random cropping. Also, during the chainization process, the comparator is fine-tuned on the augmented training set $\mathcal{P} \cup \mathcal{T}$ until it converges. We then update the linear ordering $\mathcal{L}$ and the pseudo pair set $\mathcal{T}$. For rank estimation, we use randomly selected training instances as the references.

S-2 More Experimental Results

S-2.1 Analysis

Ablation study: We analyze the efficacy of the proposed chainization algorithm on MORPH II at $\gamma = 0.1\%$. Table S-3 compares the linear extension (ρ) and age estimation (MAE) results by varying the configurations of the chainization algorithm. In method I, we assess the proposed algorithm without fine-tuning the comparator using pseudo pairs. Note that both ρ and MAE degrade. Method II performs the chainization without the iterative refinement of the comparator, and method III does it without the chain shortening. Both methods worsen the performances, indicating that the iterative refinement and the shortening are essential components.

Table S-3: Ablation study on the MORPH II dataset.

Method	Pseudo Pairs	Iteration	Chain Shortening	ρ	MAE
I				0.923	3.66
II	✓		✓	0.929	3.44
III	✓	✓		0.930	3.43
IV (Proposed)	✓	✓	✓	0.936	3.35

Iterative refinement of linear ordering: We analyze the efficacy of the iterative refinement in the chainization. Fig. S-1 plots how the linear ordering performances on MORPH II improve as the iterative fine-tuning goes on. At every γ, the performances are greatly improved at the first iteration. During this period, the comparator is firstly fine-tuned on the augmented training set $\mathcal{P} \cup \mathcal{T}$ including pseudo pairs. This confirms the effectiveness of the pseudo pairs. In general, the performances converge after four iterations.

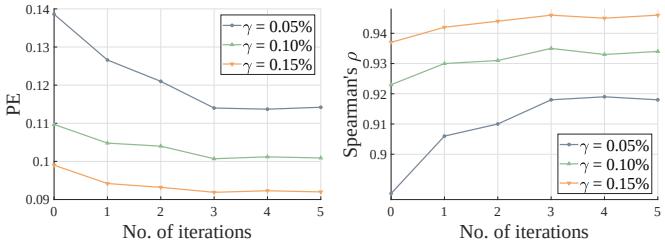


Fig. S-1: Linear extension results on MORPH II according to iterations.

Fig S-2 shows the performances on MORPH II in the bipartite case, by plotting the results for $\mathcal{X}_0$ and $\mathcal{X}_1$ separately. They exhibit similar convergence trends to Fig. S-1.

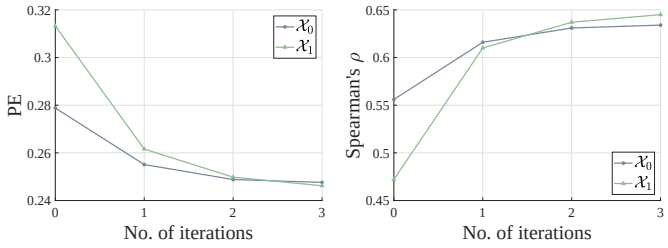
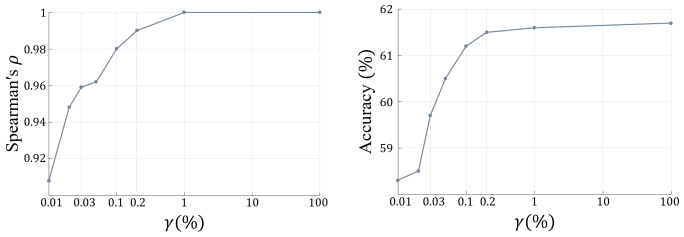


Fig. S-2: Linear extension results in the bipartite case on MORPH II according to iterations.

Linear extension and rank estimation according to γ: We analyze linear extension and rank (age group) estimation results on the Adience dataset according to γ. Fig. S-3 plots how Spearman's ρ and the rank estimation accuracy improve as γ increases. At $\gamma = 0.2\%$, the proposed algorithm achieves a high ρ of 0.99, indicating that $\mathcal{L}$ is estimated almost perfectly. Hence, the rank estimation accuracy also saturates when $\gamma > 0.2\%$. In other words, it is required to annotate only 0.2% of all pairs for the proposed algorithm to perform as effectively as the original order learning with the full annotations does.



(a) (b)

Fig. S-3: (a) Linear extension and (b) rank estimation performances at different γ's. Note that the x-axis is in a logarithmic scale.

Impacts of τ on linear extension: We analyze the impacts of the sampling threshold τ on linear extension of a partial ordering. Table S-4 compares the linear extension results on MORPH II at different τ settings, where $\gamma = 0.1\%$. In this test, we multiply the default τ in Eq. (5) by three factors. Both smaller ($\times 0.5$) and larger ($\times 2, \times 4$) factors degrade the performances. Therefore, τ in Eq. (5) is used as the default setting in this work.

S-2.2 More Results in Random Edge Case

Fig. S-4 compares the rank estimation performances at various γ on the Aesthetics dataset. Compared to OL [6], the proposed algorithm achieves higher

Table S-4: Linear extension results on MORPH II at different τ settings.

τ	PE (↓)	ρ (↑)
$\times 0.5$	0.111	0.923
$\times 1$ (Proposed)	**0.100**	**0.936**
$\times 2$	0.110	0.925
$\times 4$	0.115	0.922

accuracies at all γ's without exception. Notably, at $\gamma = 1\%$, the proposed algorithm yields 69.2% accuracy, which is comparable to 69.9% accuracy of OL at $\gamma = 100\%$. This indicates that the proposed algorithm can reduce the amount of annotated pairs required for order learning significantly.

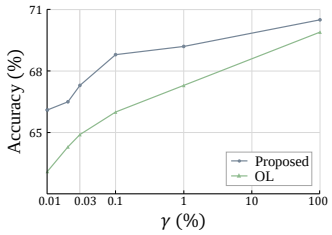


Fig. S-4: Comparison of the proposed chainization with OL [6] on Aesthetics. The x-axis is in a logarithmic scale.

Table S-5 compares the proposed algorithm with conventional ordinal regressors [2, 5, 7, 8, 8, 9] on Aesthetics. We provide the results of the proposed algorithm at $\gamma = 100\%$ as the performance upper bounds. Due to the subjectivity and ambiguity of aesthetic criteria, the pairwise comparison is challenging in the Aesthetics dataset. Nevertheless, the proposed algorithm yields comparable performances to the conventional regressors, even when $\gamma \leq 0.03\%$.

Table S-5: Comparison of rank estimation results on Aesthetics.

Algorithm	Accuracy (%)	MAE
RED-SVM [7]	64.59	0.330
OR-CNN [9]	68.96	0.326
CNNm [8]	69.45	0.376
CNNPOR [8]	70.05	0.316
SORD [2]	72.03	0.290
POE [5]	72.44	0.287
Proposed ($\gamma = 100\%$)	70.54	0.312
Proposed ($\gamma = 0.03\%$)	67.34	0.329
Proposed ($\gamma = 0.02\%$)	66.52	0.335
Proposed ($\gamma = 0.01\%$)	66.05	0.344

S-2.3 More Results in Clique-Edgeless Case

We provide more rank estimation results in the clique-edgeless case, which includes typical semi-supervised learning and unsupervised domain adaptation scenarios. First, Fig. S-5 shows the semi-supervised learning results on Aesthetics. In this test, we compare the proposed algorithm with the state-of-the-art ordinal regression technique POE [5] at various supervision levels s. Also, we use two semi-supervised learning algorithms FlexMatch [17] and FixMatch [14] for POE to exploit unlabeled instances for training. When $s \leq 50\%$, the proposed algorithm achieves the best accuracies by employing pseudo pairs as auxiliary information for training. This demonstrates that a reliable rank estimator can be obtained via the chainization when annotations are available only for a subset of training data.

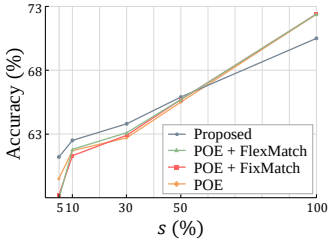


Fig. S-5: Semi-supervised learning performances on Aesthetics at various supervision levels s.

Next, Table S-6 compares the proposed algorithm with conventional unsupervised domain adaptation techniques, SAFN and HAFN [16]. Adience and MORPH II are used as the source and target domains, respectively. Note that SAFN and HAFN are optimized for object classification data, whose classes are clearly distinct from one another. However, in ordered data, inter-class differences are relatively small, so the domain adaptation is more challenging. Thus, the conventional algorithms yield poor results. In contrast, the proposed algorithm provides decent performances.

Table S-6: Domain adaptation results from Adience to MORPH II.

Algorithm	ρ	MAE	Accuracy (%)
SAFN [16]	0.525	0.93	34.0
HAFN [16]	0.493	1.05	32.1
Proposed	**0.798**	**0.54**	**51.5**

S-2.4 More Results in Bipartite Case

Feature space visualization: In Fig. S-6, we visualize the feature spaces of MORPH II in the bipartite case after the chainization, together with bipartite subset and age labels, respectively, using t-SNE. It is observed that the proposed chainization algorithm sorts the features well according to ages, even though no supervision is provided within $\mathcal{X}_0$ and $\mathcal{X}_1$.

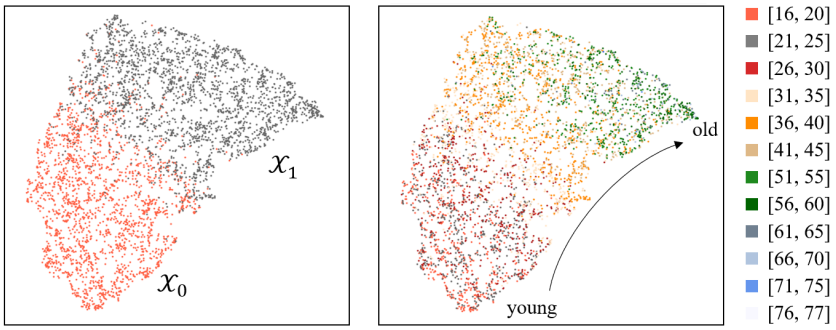
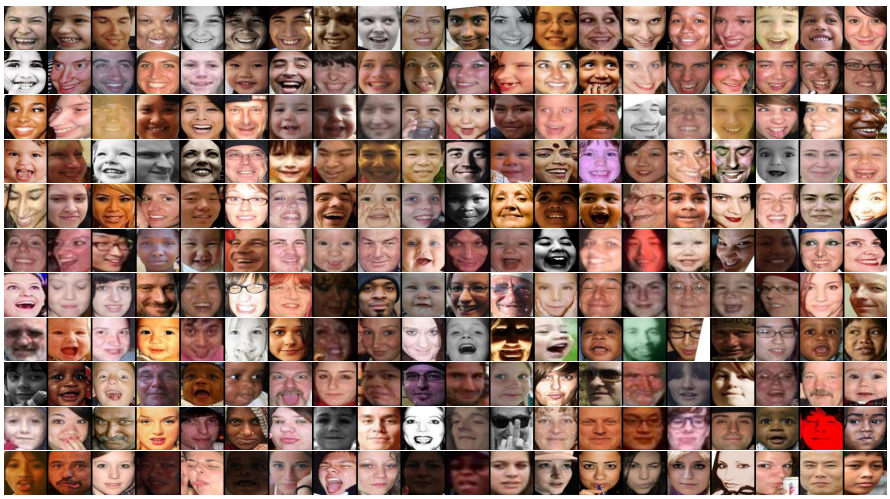


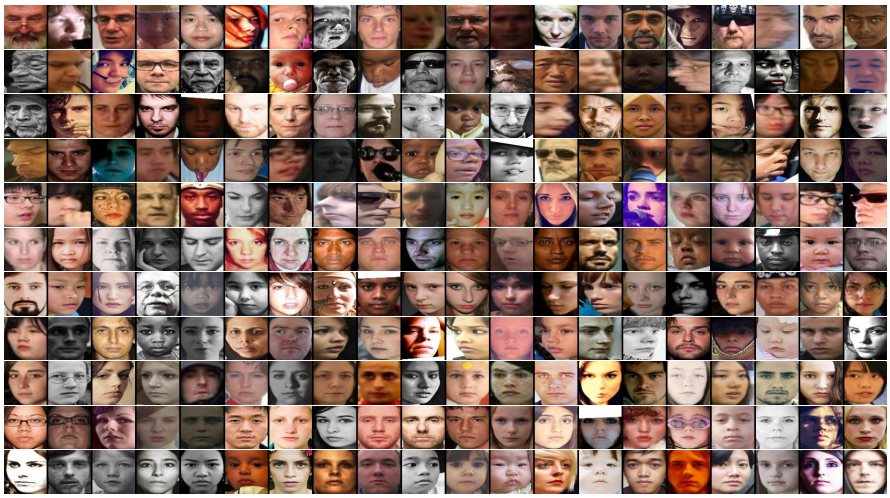
Fig. S-6: t-SNE visualization of the feature space of MORPH II in the bipartite case with (a) bipartite subset labels and (b) age labels.

More examples: We provide more sorting results in the bipartite case similarly to Fig. 8 in the main paper. In this test, we use the FER+ and RAF-DB [4] datasets. RAF-DB is a dataset for facial expression recognition, including 15,339 images in seven emotion classes.

Fig. S-7 and Fig. S-8 show examples of linear ordering on RAF-DB, while Fig. S-9 and Fig. S-10 do so on FER+.



(a) Happiness



(b) Neutral

Fig. S-7: Sorting of the instances in two selected classes 'happiness' – 'neutral' in the RAF-DB dataset.

(a) Neutral

(b) Sadness

Fig. S-8: Sorting of the instances in two selected classes 'neutral' – 'sadness' in the RAF-DB dataset.

(a) Sadness

(b) Happiness

Fig. S-9: Sorting of the instances in two selected classes 'sadness' – 'happiness' in the FER+ dataset.

(a) Neutral

(b) Anger

Fig. S-10: Sorting of the instances in two selected classes 'neutral' – 'anger' in the FER+ dataset.

S-3 Ethics and Bias Statement

Recently, ethical concerns about the fairness of deep-learning-based systems have been raised [1,10,13]. Especially, due to the intrinsic imbalance of facial datasets [9,12,18], most deep learning methods on facial analysis [11,15] have unwanted gender or racial bias. The proposed algorithm is not free from this bias either when it is trained on such datasets. Hence, the bias should be resolved before any practical usage. Also, even though the proposed algorithm discovers some subclasses from the sorting results of instances, these results should never be misinterpreted in such a way as to encourage any kind of discrimination. We recommend using the proposed algorithm for research only.

References

1. Castelvecchi, D.: Is facial recognition too biased to be let loose? Nature **587**(7834), 347–349 (2020) 13
2. Diaz, R., Marathe, A.: Soft labels for ordinal regression. In: CVPR (2019) 6
3. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: ICML (2015) 1
4. Li, S., Deng, W., Du, J.: Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild. In: CVPR (2017) 9
5. Li, W., Huang, X., Lu, J., Feng, J., Zhou, J.: Learning probabilistic ordinal embeddings for uncertainty-aware regression. In: CVPR (2021) 6, 7
6. Lim, K., Shin, N.H., Lee, Y.Y., Kim, C.S.: Order learning and its application to age estimation. In: ICLR (2020) 5, 6
7. Lin, H.T., Li, L.: Reduction from cost-sensitive ordinal ranking to weighted binary classification. Neural Computation **24**(5), 1329–1367 (2012) 6
8. Liu, Y., Kong, A.W.K., Goh, C.K.: A constrained deep neural network for ordinal regression. In: CVPR (2018) 6
9. Niu, Z., Zhou, M., Wang, L., Gao, X., Hua, G.: Ordinal regression with multiple output CNN for age estimation. In: CVPR (2016) 6, 13
10. Noorden, R.V.: The ethical questions that haunt facial-recognition research. Nature **587**(7834), 354–358 (2020) 13
11. Or-El, R., Sengupta, S., Fried, O., Shechtman, E., Kemelmacher-Shlizerman, I.: Lifespan age transformation synthesis. In: ECCV (2020) 13
12. Ricanek, K., Tesafaye, T.: MORPH: A longitudinal image database of normal adult age-progression. In: FGR (2006) 13
13. Roussi, A.: Resisting the rise of facial recognition. Nature **587**(7834), 350–353 (2020) 13
14. Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C.A., Cubuk, E.D., Kurakin, A., Li, C.L.: FixMatch: Simplifying semi-supervised learning with consistency and confidence. In: NIPS (2020) 7
15. Wen, X., Li, B., Guo, H., Liu, Z., Hu, G., Tang, M., Wang, J.: Adaptive variance based label distribution learning for facial age estimation. In: ECCV (2020) 13
16. Xu, R., Li, G., Yang, J., Lin, L.: Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In: ICCV (2019) 7
17. Zhang, B., Wang, Y., Hou, W., Wu, H., Wang, J., Okumura, M., Shinozaki, T.: FlexMatch: Boosting semi-supervised learning with curriculum pseudo labeling. In: NIPS (2021) 7
18. Zhang, Z., Song, Y., Qi, H.: Age progression/regression by conditional adversarial autoencoder. In: CVPR (2017) 13