# Metric Learning for Value Alignment

**Andrea Loreggia**[1], **Nicholas Mattei**[2], **Francesca Rossi**[3], **K. Brent Venable**[2,4]

[1] University of Padova, Department of Mathematics, Padova, Italy
[2] Tulane University, Department of Computer Science, New Orleans, LA, USA
[3] IBM Research, IBM T.J. Watson Research Center, Yorktown Heights, NY, USA
[4] Institute for Human and Machine Cognition (IHMC), Pensacola, FL, USA



## Abstract

Preference are central to decision making by both machines and humans. Representing, learning, and reasoning with preferences is an important area of study both within computer science and across the social sciences. When we give our preferences to an AI system we expect the system to make decisions or recommendations that are consistent with our preferences but the decisions should also adhere to certain norms, guidelines, and ethical principles. Hence, when working with preferences it is necessary to understand and compute a metric (distance) between preferences – especially if we encode both the user preferences and ethical systems in the same formalism. In this paper we investigate the use of CP-nets as a formalism for representing orderings over actions for AI systems. We leverage a recently proposed metric for CP-nets and a neural network architecture, CPMETRIC, for computing this metric. Using these two tools we look at the how one can build a fast and flexible value alignment system.

## 1 Introduction

Preferences are central to individual and group decision making by both computer systems and humans. Due to this central role in decision making the study of representing [Rossi *et al.*, 2011], learning [Fürnkranz and Hüllermeier, 2010], and reasoning [Domshlak *et al.*, 2011] with preferences is a focus of study within computer science and in many other disciplines including psychology and sociology [Goldsmith and Junker, 2009]. Individuals express their preferences in many different ways: pairwise comparisons, rankings, approvals (likes), positive or negative examples, and many more examples are collected in various libraries and databases [Mattei and Walsh, 2013; Mattei and Walsh, 2017; Bache and Lichman, 2013]. A core task in working with preferences is understanding the relationship *between* preferences. This often takes the form of a dominance query, i.e., which item is more or most preferred, or distance measures, i.e., which object is the closest to my stated preference. These types of reasoning are important in many domains including recommender systems [Fattah *et al.*, 2018], collective decision making [Brandt *et al.*, 2016], and value alignment systems [Russell *et al.*, 2015; Loreggia *et al.*, 2018c; Loreggia *et al.*, 2018b].

Using a formal structure to model preferences, especially one that directly models dependency, can be useful for reasoning. For example, it can support reasoning based on inference and causality, and provide more transparency and explainability since the preferences are explicitly represented and hence scrutable [Kambhampati, 2019]. A number of compact preference representation languages have been developed in the literature for representing and reasoning with preferences; see the work of Amor *et al.* [2016] for a survey of compact graphical models; we specifically focus on conditional preference networks (CP-nets) [Boutilier *et al.*, 2004].

CP-nets are a compact graphical model used to capture qualitative conditional preferences over features (variables) [Boutilier *et al.*, 2004]. Qualitative preferences are important as there is experimental evidence that qualitative preferences may more accurately reflect humans' preferences in uncertain information settings [Popova *et al.*, 2013; Allen *et al.*, 2015]. CP-nets are a popular formalism for specifying preferences in the literature and have been used for a number of applications including recommender systems and product specification [Pu *et al.*, 2011; Fattah *et al.*, 2018]. Consider a car that is described by values for all its possible features: make, model, color, and stereo options. A CP-net consists of a dependency graph and a set of statements of the form, *"all else being equal, I prefer x to y."* For example, in a CP-net one could say *"Given that the car is a Honda Civic, I prefer red to yellow."*, the condition sets the context for the preference over alternatives. These preferences are qualitative, i.e., there is no quantity expressing the degree of preference.

A CP-net induces an ordering over all possible *outcomes*, i.e., all complete assignments to the set of features. This is a partial order if the dependency graph of the CP-net is acyclic, i.e., the conditionality of the statements does not create a cycle, as is often assumed in work with CP-nets [Goldsmith *et al.*, 2008]. The size of the description of the CP-net may be exponentially smaller than the partial order it describes. Hence, CP-nets are called a *compact* representation and reasoning and learning on the compact structure, instead of the full order, is an important topic of research. Recent work proposes the first formal metric to describe the distance between CP-nets [Loreggia *et al.*, 2018a] and the related formalism of LP-trees [Li and Kazimipour, 2018] in a rigorous way. What is im-

portant is not the differences in the surface features of the CP-nets, e.g., a single statement or dependency, but rather the distance between their induced partial orders. Even a small difference in a CP-net could generate a very different partial order, depending on which feature is involved in the modification. While the metrics proposed by Loreggia *et al.* [2018a] are well grounded, they are computationally hard to compute, in general, and approximations must be used.

We envision the use of CP-nets to solve one part of the *value alignment* problem [Russell *et al.*, 2015; Loreggia *et al.*, 2018c; Loreggia *et al.*, 2018b]. This is part of a broader research program we call *Ethically Bounded AI*, that seeks to understand how to harness the power of AI yet prevent these systems from making choices we do not consider ethical [Rossi and Mattei, 2019]. In this work we envision using a distance metric to measure the difference between an individual agent's preferences over actions and another ordering given by, e.g., ethics, norms, or business values [Loreggia *et al.*, 2018c; Loreggia *et al.*, 2018b].

Following work in metric learning over structured representations [Bellet *et al.*, 2015], we wish to learn the distance between partial orders represented compactly as CP-nets. We do not want to work with the partial orders directly as they may be exponentially larger than the CP-net representation. Informally, given two CP-nets, we wish to estimate the distance between their induced partial orders using a neural network. The number of possible CP-nets grows extremely fast, from 481,776 for 4 binary features to over $5.24 \times 10^{40}$ with 7 binary features [Allen *et al.*, 2017]. However, the computation time of the approximation algorithm proposed by Loreggia *et al.* [2018a] scales linearly with the number of features, hence, new methods must be explored. Therefore, leveraging the inferential properties of neural networks may help us make CP-nets more useful as a preference reasoning formalism.

**Contributions.** We propose using metric learning as a tool to practically solve aspects of the value alignment problem. We propose to model both user preferences and ethical priorities over actions using the CP-net formalism and we demonstrate how one can use a state of the art neural network formulation, CPMETRIC, to quickly and accurately judge distances between preferred actions and ethical actions. We evaluate our models and metrics on generated CP-nets and show that CPMETRIC leads to a speed up in computation while still being accurate.

## 2 CP-nets

Conditional Preference networks (CP-nets) are a graphical model for compactly representing conditional and qualitative preference relations [Boutilier *et al.*, 2004]. CP-nets are comprised of sets of *ceteris paribus* preference statements (cp-statements). For instance, the cp-statement, *"I prefer red wine to white wine if meat is served,"* asserts that, given two meals that differ *only* in the kind of wine served *and* both containing meat, the meal with red wine is preferable to the meal with white wine. CP-nets have been extensively used in the preference reasoning preference learning and social choice literature as a formalism for working with qualitative preferences [Domshlak *et al.*, 2011; Rossi *et al.*, 2011; Brandt *et al.*, 2016]. CP-nets have even been used to compose web services [Wang *et al.*, 2009] and other decision aid systems [Pu *et al.*, 2011].

Formally, a CP-net has a set of features (or variables) $F = \{X_1, \ldots, X_n\}$ with finite domains $\mathcal{D}(\mathcal{X}_\infty), \ldots, \mathcal{D}(\mathcal{X}_\setminus)$. For each feature $X_i$, we are given a set of *parent* features $Pa(X_i)$ that can affect the preferences over the values of $X_i$. This defines a *dependency graph* in which each node $X_i$ has $Pa(X_i)$ as its immediate predecessors. An *acyclic* CP-net is one in which the dependency graph is acyclic. Given this structural information, one needs to specify the preference over the values of each variable $X_i$ for *each complete assignment* to the parent variables, $Pa(X_i)$. This preference is assumed to take the form of a total or partial order over $\mathcal{D}(\mathcal{X}_\rangle)$. A cp-statement for some feature $X_i$ that has parents $Pa(X_i) = \{x_1, \ldots, x_n\}$ and domain $D(X_i) = \{a_1, \ldots, a_m\}$ is a total ordering over $D(X_i)$ and has general form: $x_1 = v_1, x_2 = v_2, \ldots, x_n = v_n : a_1 \succ \ldots \succ a_m$, where for each $X_i \in Pa(X_1) : x_i = v_i$ is an assignment to a parent of $X_i$ with $v_i \in \mathcal{D}(\mathcal{X}_\rangle)$. The set of cp-statements regarding a certain variable $X_i$ is called the cp-table for $X_i$.

Consider the CP-net depicted graphically in Figure 1 (left) with features are $A$, $B$, $C$, and $D$. Figure 1 (right) gives the full induced preference order for the CP-net. Each variable has binary domain containing $f$ and $\overline{f}$ if $F$ is the name of the feature. All cp-statements in the CP-net are: $a \succ \overline{a}$, $b \succ \overline{b}$, $(a \wedge b) : c \succ \overline{c}$, $(\overline{a} \wedge \overline{b}) : c \succ \overline{c}$, $(a \wedge \overline{b}) : \overline{c} \succ c$, $(\overline{a} \wedge b) : \overline{c} \succ c$, $c : d \succ \overline{d}$, $\overline{c} : \overline{d} \succ d$. Here, statement $a \succ \overline{a}$ represents the unconditional preference for $A = a$ over $A = \overline{a}$, while statement $c : d \succ \overline{d}$ states that $D = d$ is preferred to $D = \overline{d}$, given that $C = c$. The semantics of CP-nets depends on the notion of a *worsening flip*: a change in the value of a variable to a less preferred value according to the cp-statement for that variable. For example, in the CP-net above, passing from $abcd$ to $\overline{a}bcd$ is a worsening flip since $c$ is better than $\overline{c}$ given $a$ and $b$. One outcome $\alpha$ is *preferred to* or *dominates* another outcome $\beta$ (written $\alpha \succ \beta$) if and only if there is a chain of worsening flips from $\alpha$ to $\beta$. This definition induces a preorder over the outcomes, which is a partial order if the CP-net is acyclic [Boutilier *et al.*, 2004], as depicted in Figure 1 (right).

The complexity of dominance and consistency testing in CP-nets is an area of active study in preference reasoning [Goldsmith *et al.*, 2008; Rossi *et al.*, 2011]. Finding the optimal outcome of a CP-net is NP-hard [Boutilier *et al.*, 2004] in general but can be found in polynomial time for acyclic CP-nets by assigning the most preferred value for each cp-table. Indeed, acyclic CP-nets induce a lattice over the outcomes as (partially) depicted in Figure 1 (right). The induced preference ordering, Figure 1 (right), can be exponentially larger than the CP-net Figure 1 (left), which motivates learning a metric using only the (more compact) CP-net.

## 3 Preferences and Ethical Priorities

In what follow we assume that we can describe a user's behavior through her preferences over the features of the domain. This can be modeled as a CP-net and it gives us an ordering over the actions the user would like to take. The example given in Loreggia *et al.* [2018c] concerns a driver of a vehicle: they

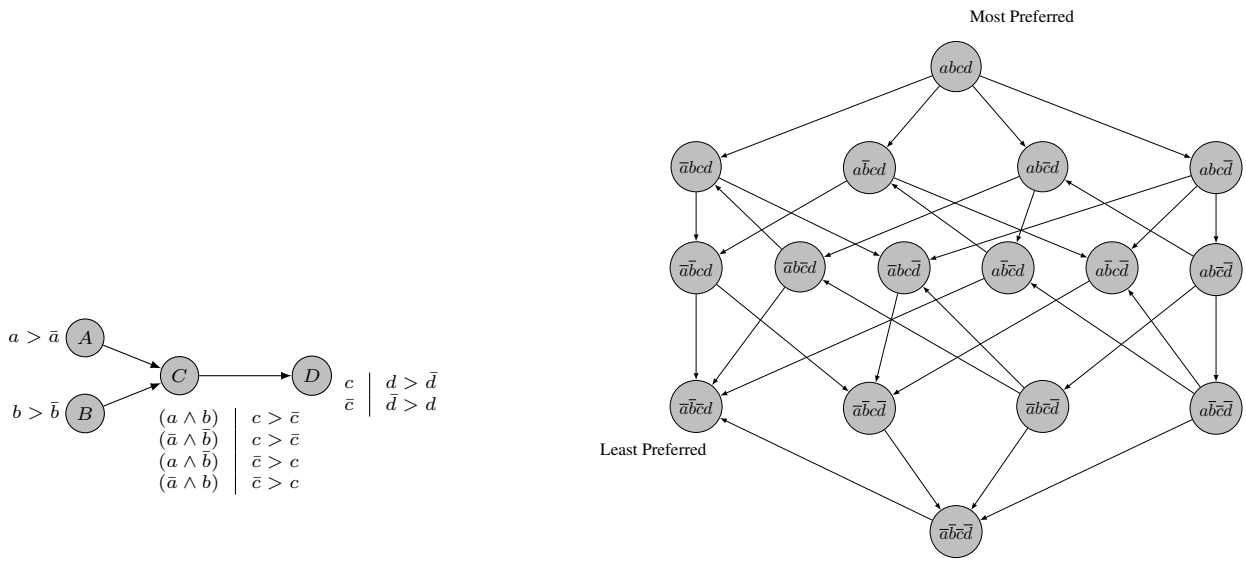


Figure 1: A CP-net with $n = 4$ features (left) and part of in the induced partial order (right). Note that the partial order is over all $2^n = 16$ possible combinations and arrows denote the dominance relation. We have arranged the nodes so that each is one flip between the levels.

may want to go as fast as possible and run over certain small animals. However, there may be some overall ethical or moral guidelines (priorities) the system must follow. In this case we want to evaluate the difference, or distance, between the individual user and the society. This idea of morality as ordering over actions was first proposed by Sen [1974].

To operationalize this system we wish to describe both preferences and priorities using the CP-net formalism and using a notion of distance in the metric space of CP-nets. This enables us and the system to understand whether users' preferences are close enough to the moral principles or not. Eventually, when preferences deviate from the desired behavior, we can use CP-nets, since they induce an ordering, to find a trade-off so that the quality of the decision with respect to the subjective preferences does not significantly degrade when conforming to the ethical principles [Loreggia *et al.*, 2018c; Loreggia *et al.*, 2018b]. In this way we have bounded the behavior of the system to be *ethical* while still being responsive to the user preferences [Rossi and Mattei, 2019].

Traditional reasoning and learning approaches in AI provide different and complementary capabilities to an autonomous agent. Symbolic and logical reasoning allow these agents to manipulate symbols and perform inference, while machine learning techniques can learn and optimize many ill-defined problems from large amounts of data. As ongoing work, we intend to study the use of both kinds of approaches to model, learn, and reason with both preferences and ethical principles. Inspired by the System 1 and System 2 theory of Daniel Kahneman [Kahneman, 2011], we will define a dual-agent architecture that will provide autonomous agents with the ability to combine symbolic and accurate reasoning with data interpretation and learning, for both preferences and ethical principles. This will allow machines to be flexible and context-dependent in how they handle and combine these two sources of information for decision making [Rossi and Loreggia, 2019; Rossi and Mattei, 2019].

The combined use of deep learning techniques and logical reasoning formalisms is an exciting research direction to find a principled ways to develop AI systems that are both accountable and able to explain themselves. We hope that these approaches will be able to overcome limitations of the "black-box paradigm" in the machine learning discipline [Rossi and Loreggia, 2019].

## 4 Metric Learning on CP-nets

Metric learning algorithms aim to learn a metric (or distance function) over a set of training points or samples [Sohn, 2016]. The importance of metrics has grown in recent years with the use of these functions in many different domains: from clustering to information retrieval and from recommender systems to preference aggregation. For instance, many clustering algorithms like the $k$-Means or classification algorithm including $k$-Nearest Neighbor use a distance value between points.

Formally, a metric space is a pair $(M, d)$ where $M$ is a set of elements and $d$ is a function $d : M \times M \to \mathbb{R}$ where $d$ satisfies four criteria. Given any three elements $A, B, C \in M$, $d$ must satisfy

1. (1) $d(A, B) \geq 0$, there must be a value for all pairs;
2. (2) $d(A, B) = d(B, A)$, $d$ must be symmetric;
3. (3) $d(A, B) \leq d(A, C) + d(C, B)$; $d$ must satisfy the triangle inequality; and
4. (4) $d(A, B) = 0$ if and only if $A = B$; $d$ can be zero if and only if the two elements are the same.

Xing *et al.* [2002] first formalized the problem of metric learning, i.e., learning the metric directly from samples rather

than formally specifying the function $d$. This approach requires training data, meaning that we have some oracle that is able to give the value of the metric for each pair. The success of deep learning in many different domains [Krizhevsky *et al.*, 2012] has lead many researchers to apply these approaches to the field of metric learning, resulting in a number of important results [Bellet *et al.*, 2015].

In this work we focus on metric spaces ($M$, $d$) where $M$ is a set of CP-nets. Given this, we want to learn the distance $d$ which best approximates the Kendall tau distance (KTD) [Kendall, 1938] between the induced partial orders. Informally, the Kendall tau distance between two orderings is the number of pairs that are *discordant*, i.e., not ordered in the same way, in both orderings. This distance metric extended to partial orders (Definition 1) was shown to be a metric on the space of CP-nets by Loreggia *et al.* [2018a]. To extend the classic KTD to CP-nets, a penalty parameter $p$ defined for partial rankings [Fagin *et al.*, 2006] was extended to the case of partial orders. Loreggia *et al.* [2018a] assume that all CP-nets are acyclic and in minimal (non-degenerate) form, i.e., all arcs in the dependency graph have a real dependency expressed in the cp-statements, a standard assumption in the CP-net literature (see e.g., [Allen *et al.*, 2017; Boutilier *et al.*, 2004]).

**Definition 1.** *Given two CP-nets $A$ and $B$ inducing partial orders $P$ and $Q$ over the same unordered set of outcomes $U$: $KTD(A,B) = KT(P,Q) = \sum_{\forall i,j \in U, i \neq j} K^p_{i,j}(P,Q)$ where $i$ and $j$ are two outcomes with $i \neq j$ (i.e., iterate over all unique pairs), we have:*

1. *$K^p_{i,j}(P,Q) = 0$ if $i,j$ are ordered in the same way or are incomparable in $P$ and $Q$;*
2. *$K^p_{i,j}(P,Q) = 1$ if $i,j$ are ordered inversely in $P$ and $Q$;*
3. *$K^p_{i,j}(P,Q) = p$, $0.5 \leq p < 1$ if $i,j$ are ordered in $P$ and incomparable in $Q$ (resp. $Q,P$).*

*To make this distance scale invariant, i.e., a value in $[0,1]$, it is divided by $|U|$.*

CP-nets present two important challenges when used for metric learning. The first is that we are attempting to learn a metric via a compact representation of a partial order. We are not learning over the partial orders induced by the CP-nets directly, as they could be exponentially larger than the CP-nets. The second challenge is the encoding of the graphical structure itself. Graph learning with neural networks is still a active and open area of research; Goyal and Ferrara [2017] give a complete survey of recent work as well as a Python library of implementations for many of these techniques. Most of these works focus on finding good embeddings for the nodes of the network and then using collections of these learned embeddings to represent the graph for, e.g., particular segmentation or link prediction tasks. None of these techniques have been applied to embedding graphs for metric learning.

## 5 Structure of CPMETRIC

In our task the metric space is $(M,d)$ where $M$ is a set of compact, graphical preferences that induce a partial order and our goal is to learn the metric $d$ *only* from the compact, graphical representation. The key challenge is the need to find a vector representation of not only the graph but the cp-statement. We briefly define the representations used for CPMETRIC here, for a complete overview, see Loreggia *et al.* [2019].

We represent a CP-net $I$ over $m$ using two matrices. First is the adjacency matrix $adj_I$ which represents the dependency graph of the CP-net and is a $m \times m$ matrix of 0s and 1s. The second matrix represents the list of cp-statements $cpt_I$, which is a $m \times 2^{m-1}$ matrix, where each row represents a variable $X_i \in F$ and each column represents a complete assignment for each of the variables in $F \setminus X_i$. The list is built following a topological ordering of variables in the CP-net. Each cell $cpt_I(i,j)$ stores the preference value for the $i$th variable given the $j$th assignment to variables in $F \setminus X_i$.

The set of training examples $X = \{x_1, \ldots, x_n\}$ is made up of pairs of CP-nets represented through their normalized Laplacians and the cp-statements. The set of corresponding labels $Y = \{y_1, \ldots, y_n\}^T$, where each $y_i \in Y, y_i \in [0,1]$ is the normalized value of KTD between the CP-nets in $x_i$. Each $x_i \in X$ is then a tuple $(\mathcal{L}_A, cpt_A, \mathcal{L}_B, cpt_B)$ representing a pair of CP-net $(A,B)$ by their Laplacian, $\mathcal{L}_A$, and the encoding of their cp-statements, $cpt_A$.

## 6 Experiment

CPMETRIC is trained to learn the KTD metric by varying the number of features of the CP-nets $n \in \{3, \ldots, 7\}$ and using two different autoencoders. For a complete discussion of the performance of CPMETRIC on the standard classification and regression tasks in comparison with *I*-CPD, including a discussion of tuning hyperparameters see Loreggia *et al.* [2019]. In this paper we focus on what we call the *comparison task* that is necessary for value alignment systems. Informally, this is the task where given two CP-nets, say one representing the preferences of the user and one representing a set of ethical values or constraints, we want to decide if some third CP-net, say the course of action to be taken, is *closer* to the preferences or closer to the constraints.

### 6.1 Data Generation and Training

For each number of features $n \in \{3, \ldots, 7\}$ we generate 1000 CP-nets uniformly at random using the generators from Allen et al. [Allen *et al.*, 2017]. This set of CP-nets is split into a training-generative-set (900 CP-nets) and test-generative-set (100 CP-nets) 10 different ways to give us 10 fold cross validation. For each fold we compute the training and test dataset comprised of all, e.g., $\binom{900}{2}$, possible pairs of CP-nets from the training-generative-set and test-generative-set, respectively, along with the value of KTD for that pair. While we generate the CP-nets themselves uniformly at random observe that this creates an unbalanced set of distances – it induces a normal distribution – and hence our sets are unbalanced. Figure 2 shows the distribution of of CP-net pairs over 20 intervals for all CP-nets generated for $n \in \{3, \ldots, 7\}$. While our classification experiments are for $m = 10$ classes, dividing the interval into 20 classes provides a better visualization of the challenge of obtaining training samples at the edges of the distribution.

We ran a preliminary experiment on balancing our dataset by sub-sampling the training and test datasets. In these small experiments, performance was much worse than per-

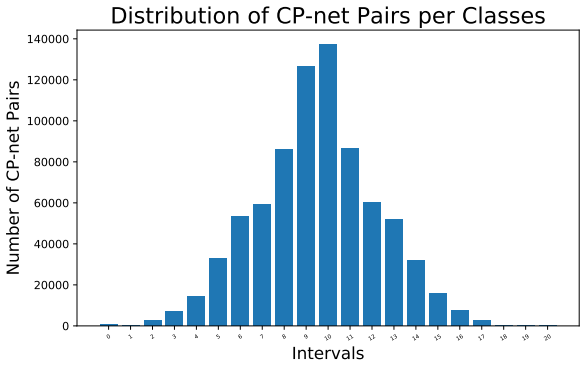


Figure 2: Histogram of the number of CP-net pairs per interval across all experimental datasets. CP-nets pairs are not distributed uniformly in the class intervals.

formance on the unbalanced dataset. Because we are learning a metric, for each CP-net $A$, there is only one CP-net $B$ such $KTD(A, B) = 1$ and only one CP-net $C$ such $KTD(A, C) = 0$. Consequently, attempting to balance or hold out CP-nets from test or train can lead to poor performance. We conjecture that in order to improve this task we should perform some kind of data augmentation, but this would introduce more subjective assumptions on how and where data should be augmented [Wong *et al.*, 2016].

All training was done on a machine with 2 x Intel(R) Xeon(R) CPU E5-2670 @ 2.60GHz and one NVidia K20 128GB GPU. We train CPMETRIC for 70 epochs using the Adam optimizer [Kingma and Ba, 2014]. For each number of features of the CP-net $n$ we use all $\binom{900}{2}$ pairs in the training-set. There are only 488 binary CP-nets with 3 features [Allen *et al.*, 2017], hence, for $n = 3$ the training-set is 17K samples while for $n > 3$ the number of samples in the training-set is 800K. Both the *Autoencoder* and *Siamese Autoencoder* models are trained for 100 epochs using the Adam optimizer [Kingma and Ba, 2014] using the same training-set. Model weights from the best performing epoch are saved and subsequently transferred to the deep neural network used to learn the distance function.

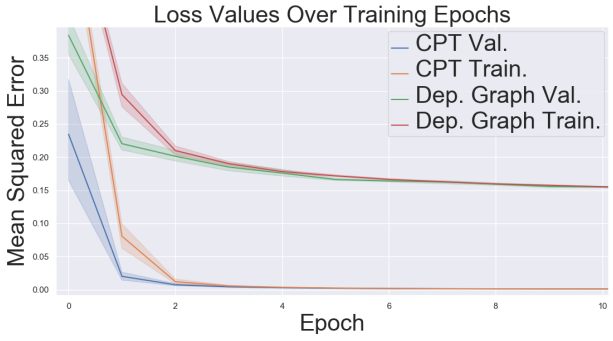


Figure 3: Performance of the autoencoder on the validation and training set for 10 epochs.

The training and validation loss for the autoencoder is shown in Figure 3. Observe that the loss for the CPT representation approaches zero after only 3 epochs for both the training and validation phases. The same trend is true for the adjacency matrix, though the loss converges to $\approx 0.15$.

## 6.2 Comparison Task Performance

For many applications we are not concerned with the true value of the distance but rather deciding which of two preferences is closer to a reference point. For example, in product recommendation we may want to display the closer of two objects and not care about computing the values [Pu *et al.*, 2011]. Formally, the qualitative comparison takes takes a set of CP-nets triples $(A, B, C)$, where $A$ is a reference CP-net and the task is to decide which other CP-net $B$ or $C$ is *closer* to $A$. We generate uniformly at random 1000 triples of CP-nets for each $n \in \{3, \ldots, 7\}$. For each triple $(A, B, C)$ we compute both KTD$(A, B)$ and KTD$(A, C)$ to establish ground truth and use our regression networks to predict the distance between $(A, B)$ and $(A, C)$.

Table 1 displays the accuracy, as a percentage out of 1000 trials, of our three CPMETRIC architectures versus *I*-CPD for this task; Table 2 gives the average runtime per pair, averaged over all 1000 trials. The standard deviations in Table 1 are across the 10 folds of the training/test set. For all of our networks we obtain an accuracy above $85\%$ and all the networks perform about the same on this task ($\pm 2.0\%$) and the trend for accuracy is flat across the size of the CP-nets. It is interesting to note that on this task neither of the autoencoders were able to significantly improve performance as they did for the quantitative comparison tasks. While the results are inconclusive, as all instances of CPMETRIC performed about the same, it will be interesting to see if there are autoencoder architectures that are more suited to the comparison task.

A positive take away is that, as Table 2 shows, we achieve a sub-linear increase in inference time for our model. *I*-CPD scales linearly with the description size of the CP-net so the neural network does, after training, offer the ability to, in a practical amount of time, compare CP-nets of larger sizes. This gives us hope that while the metric itself is NP-hard to compute in a direct way, we can use the power of deep learning to enable systems that could be practically deployed.

Unfortunately the trend for accuracy is negative when the number of features increases. However, this is also the case for the *I*-CPD approximation and both metrics seem to be losing accuracy at about the same rate. The loss in accuracy for the neural network models could be caused by the unbalanced nature of the training and testing datasets. Again, as shown in Figure 2, generating the CP-nets themselves uniformly at random does not give us a uniform distribution over distances and correcting this may give better performance.

Our conjecture for the slight disadvantage for CPMETRIC over *I*-CPD has to do with the directionality of the errors. When training the network we are optimizing for accuracy on the regression task. However, when using this metric it does not matter if CPMETRIC overestimates by a small amount or underestimates by some small amount. However, when looking at the comparison task, it may matter a lot if the direction of our errors is random. An important direction for

| | No Autoencoder | Autoencoder | Siam. Autoencoder | *I*-CPD |
|---|---|---|---|---|
| N | Accuracy on Triples | Accuracy on Triples | Accuracy on Triples | Accuracy on Triples |
| 3 | 85.01% (2.01%) | 85.76% ( 2.29%) | 85.47% (2.32%) | 91.80% |
| 4 | 91.17% (0.92%) | 91.38% (1.10%) | 91.78% (1.13%) | 92.90% |
| 5 | 88.40% (0.91%) | 89.36% (1.08%) | 89.18% (1.08%) | 90.80% |
| 6 | 87.33% (0.80%) | 87.17% (1.33%) | 86.79% (1.84%) | 90.10% |
| 7 | 84.79% (1.16%) | 84.57% (1.14%) | 85.12% (0.86%) | 89.90% |

Table 1: Performance of the various network architectures on the qualitative comparison task as well as performance of *I*-CPD. While our networks do not achieve the best performance on this task they are competitive with the more costly approximation algorithm *I*-CPD.

| N | *I*-CPD | Autoencoder Neural Network |
|---|---|---|
| 3 | 0.69 (0.48) msec | 0.087 (0.004) msec |
| 4 | 1.09 (0.33) msec | 0.098 (0.004) msec |
| 5 | 1.85 (0.49) msec | 0.100 (0.005) msec |
| 6 | 3.16 (0.74) msec | 0.114 (0.003) msec |
| 7 | 4.65 (0.86) msec | 0.138 (0.001) msec |

Table 2: Comparison of the mean runtime for a single triple over 1000 trials on the qualitative comparison task of the neural network and *I*-CPD.

future work is to try different optimization objectives when training the network to see if this bias is the reason for the underperformance.

## 7 Conclusion

In this paper we have discussed how to use CPMETRIC, a novel neural network model to learn a metric (distance) function between partial orders induced from a CP-net, a compact, structured preference representation, to enable practical value alignment systems. To our knowledge this is the first use of neural networks to learn and measure preferences for the value alignment problem. We feel that this is an interesting and fruitful direction for research in the AI Safety domain as we must develop practical and efficient tools that can be used to effectively harness the power of AI systems.

Important directions for future work include integrating novel graph learning techniques to our networks and extending our work to other formalisms including, e.g., PCP-nets [Cornelio *et al.*, 2013] and LP-trees [Li and Kazimipour, 2018]. PCP-nets are a particularly interesting direction as they have been proposed as an efficient way to model uncertainty over the preferences of a single or multiple agents [Cornelio *et al.*, 2015]. Another important extension involves setting contexts for different preference and ethical priority encodings. CP-nets and many other preference formalisms model a particular domain but do not give us any insight into when we may need to pass between one context and another.

## References


[Allen *et al.*, 2015] T. E. Allen, M. Chen, J. Goldsmith, N. Mattei, A. Popova, M. Regenwetter, F. Rossi, and C. Zwilling. Beyond theory and data in preference modeling: Bringing humans into the loop. In *Proc. 4th ADT*, 2015.

[Allen *et al.*, 2017] T. E. Allen, J. Goldsmith, H. E. Justice, N. Mattei, and K. Raines. Uniform random generation and dominance testing for cp-nets. *JAIR*, 59:771–813, 2017.

[Amor *et al.*, 2016] N. B. Amor, D. Dubois, H. Gouider, and H. Prade. Graphical models for preference representation: An overview. In *Proceedings of the 10th International Scalable Uncertainty Management (SUM 2016)*, pages 96–111, 2016.

[Bache and Lichman, 2013] K. Bache and M. Lichman. UCI Machine Learning Repository, 2013. University of California, Irvine, School of Information and Computer Sciences.

[Bellet *et al.*, 2015] Aurélien Bellet, Amaury Habrard, and Marc Sebban. *Metric Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2015.

[Boutilier *et al.*, 2004] C. Boutilier, R. Brafman, C. Domshlak, H.H. Hoos, and D. Poole. CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *Journal of Artificial Intelligence Research*, 21:135–191, 2004.

[Brandt *et al.*, 2016] F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, editors. *Handbook of Computational Social Choice*. Cambridge University Press, 2016.

[Cornelio *et al.*, 2013] C. Cornelio, J. Goldsmith, N. Mattei, F. Rossi, and K.B. Venable. Updates and uncertainty in CP-nets. In *Proc. 26th AUSAI*, 2013.

[Cornelio *et al.*, 2015] C. Cornelio, U. Grandi, J. Goldsmith, N. Mattei, F. Rossi, and K.B. Venable. Reasoning with PCP-nets in a multi-agent context. In *Proc. 14th AAMAS*, 2015.

[Domshlak *et al.*, 2011] C. Domshlak, E. Hüllermeier, S. Kaci, and H. Prade. Preferences in AI: An overview. *AI*, 175(7):1037–1052, 2011.

[Fagin *et al.*, 2006] Ronald Fagin, Ravi Kumar, Mohammad Mahdian, D. Sivakumar, and Erik Vee. Comparing partial rankings. *SIAM J. Discret. Math.*, 20(3):628–648, March 2006.

[Fattah *et al.*, 2018] Sheik Mohammad Mostakim Fattah, Athman Bouguettaya, and Sajib Mistry. A CP-Net based qualitative composition approach for an IaaS provider. In *International Conference on Web Information Systems Engineering*, pages 151–166. Springer, 2018.

[Fürnkranz and Hüllermeier, 2010] J. Fürnkranz and E. Hüllermeier. *Preference Learning*. Springer, 2010.

[Goldsmith and Junker, 2009] J. Goldsmith and U. Junker. Preference handling for artificial intelligence. *AI Magazine*, 29(4), 2009.

[Goldsmith *et al.*, 2008] J. Goldsmith, J. Lang, M. Truszczyński, and N. Wilson. The computational complexity of dominance and consistency in CP-nets. *Journal of Artificial Intelligence Research*, 33(1):403–432, 2008.

[Goyal and Ferrara, 2017] Palash Goyal and Emilio Ferrara. Graph embedding techniques, applications, and performance: A survey. *CoRR*, abs/1705.02801, 2017.

[Kahneman, 2011] Daniel Kahneman. *Thinking, fast and slow*. Farrar, Straus and Giroux, New York, 2011.

[Kambhampati, 2019] S. Kambhampati. Synthesizing explainable behavior for human-ai collaboration. In *Proc. 18th AAMAS*, 2019.

[Kendall, 1938] M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.

[Kingma and Ba, 2014] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv/1412.6980*, 2014.

[Krizhevsky *et al.*, 2012] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. Advances in Neural and Information Processing Systems (NeurIPS)*, pages 1097–1105, 2012.

[Li and Kazimipour, 2018] Minyi Li and Borhan Kazimipour. An efficient algorithm to compute distance between lexicographic preference trees. In *Proc. 27th IJCAI*, pages 1898–1904, 2018.

[Loreggia *et al.*, 2018a] A. Loreggia, N. Mattei, F. Rossi, and K. B. Venable. On the distance between CP-nets. In *Proc. 17th AAMAS*, 2018.

[Loreggia *et al.*, 2018b] A. Loreggia, N. Mattei, F. Rossi, and K. B. Venable. Preferences and ethical principles in decision making. In *Proceedings of the 1st AAAI/ACM Conference on AI, Ethics, and Society (AIES)*, 2018.

[Loreggia *et al.*, 2018c] A. Loreggia, N. Mattei, F. Rossi, and K. B. Venable. Value alignment via tractable preference distance. In R. V. Yampolskiy, editor, *Artificial Intelligence Safety and Security*, chapter 18. CRC Press, 2018.

[Loreggia *et al.*, 2019] A. Loreggia, N. Mattei, F. Rossi, and K. B. Venable. CPMETRIC: Deep siamese networks for learning distances between structured preferences. *arXiv preprint arXiv:1809.08350*, 2019.

[Mattei and Walsh, 2013] N. Mattei and T. Walsh. PREFLIB: A library for preferences, HTTP://WWW.PREFLIB.ORG. In *Proc. 3rd ADT*, 2013.

[Mattei and Walsh, 2017] N. Mattei and T. Walsh. A PREFLIB.ORG Retrospective: Lessons Learned and New Directions. In U. Endriss, editor, *Trends in Computational Social Choice*, chapter 15, pages 289–309. AI Access Foundation, 2017.

[Popova *et al.*, 2013] A. Popova, M. Regenwetter, and N. Mattei. A behavioral perspective on social choice. *AMAI*, 68(1–3):135–160, 2013.

[Pu *et al.*, 2011] P. Pu, B. Faltings, L. Chen, J. Zhang, and P. Viappiani. Usability guidelines for product recommenders based on example critiquing research. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*, pages 511–545. Springer, 2011.

[Rossi and Loreggia, 2019] F. Rossi and A. Loreggia. Preferences and ethical priorities: Thinking fast and slow in AI. In *Proc. 18th AAMAS*, pages 3–4, 2019.

[Rossi and Mattei, 2019] Francesca Rossi and Nicholas Mattei. Building ethically bounded AI. In *Proc. 33rd AAAI*, 2019.

[Rossi *et al.*, 2011] F. Rossi, K.B. Venable, and T. Walsh. *A Short Introduction to Preferences: Between Artificial Intelligence and Social Choice*. Morgan and Claypool, 2011.

[Russell *et al.*, 2015] Stuart Russell, Daniel Dewey, and Max Tegmark. Research priorities for robust and beneficial artificial intelligence. *AI Magazine*, 36(4):105–114, 2015.

[Sen, 1974] A. Sen. *Choice, Ordering, and Morality*. Blackwell, 1974.

[Sohn, 2016] K. Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems (NeruIPS)*, pages 1857–1865, 2016.

[Wang *et al.*, 2009] Hongbing Wang, Shizhi Shao, Xuan Zhou, Cheng Wan, and Athman Bouguettaya. Web service selection with incomplete or inconsistent user preferences. In *Proc. 7th International Conference on Service-Oriented Computing*, pages 83–98. Springer, 2009.

[Wong *et al.*, 2016] S. C. Wong, A. Gatt, V. Stamatescu, and M. D McDonnell. Understanding data augmentation for classification: When to warp? In *Proc. of the 2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–6, 2016.

[Xing *et al.*, 2002] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. J. Russell. Distance metric learning with application to clustering with side-information. In *Proc. 15th NeurIPS*, pages 505–512, 2002.