Data-driven discrete choice modeling for human trajectory forecasting in crowds

Pierre Schutz
Visual Intelligence for Transportation Laboratory (VITA)
EPFL, Switzerland
pierre.schutz@epfl.ch

Abstract

In the last decade, data-driven methods using machine learning have achieved state-of-the-art performances in various applications, leading to a revolution in industries such as healthcare, finance, and transportation. While these new methods outperform the traditional knowledge-driven techniques predictions, their results are often difficult to interpret, and the models can lack robustness. Explainable AI research tries to alleviate to fix the issue, especially for safety-critical problems. In this work, we use a hybrid approach to tackle the problem of human trajectory forecasting in crowds. We demonstrate how we can compromise between interpretability and accuracy by using expert knowledge and constraints on a neural network input and output. The proposed model enhances the performances of a discrete choice model while preserving independent concepts that we can visualize to understand the model's prediction. We also show how data-driven techniques can help discrete choice model parameters' estimation when dealing with a large dataset for imbalanced classification.

1. Introduction

Humans navigate easily in crowds and follow complex social rules to avoid a collision, give the right of way, or follow someone else. Predicting human movement is challenging and has applications for building robots navigating in crowds, virtual reality environments, or simulating flows to assess buildings' safety. The ability to predict human motion in a social context with accuracy, robustness, and in an intelligible manner is crucial to building safe technology. This problem has been tackled in research using two paradigms. Early works focused on building hand-crafted functions using expert knowledge to model the decision-making process to infer a choice. Social Forces [7], ORCA [15], or discrete choice modeling [3] were able to model social interactions such as collision avoidance. More recently, with the breakthrough of deep neural networks, data-driven methods became the state-of-the-art solution for this problem. Using large datasets, they empirically learn the walking behavior and interactions without needing hand-designed functions. The SocialLSTM [1] or SocialGAN [5] achieved state-of-the-art performances by applying recent machine learning concepts to the problem of trajectory forecasting. These two paradigms differ in their approaches to tackling the problem and have different advantages and drawbacks. Knowledge-driven methods are robust and provide interpretable predictions. They are also lightweight: they do not require long training and high computational power. They are, by design, limited to the domain knowledge and miss all information not described by their hand-design functions. Also, these functions depend on assumptions that can be unadapted to a different context (a different country or crowd density). Data-driven methods' advantage is to learn everything from the data, making them adaptable to a new dataset from a different environment. They can model complex interactions and learn patterns unknown from the domain knowledge. Using recurrent connections, they can use the history of positions and interactions instead to make predictions, leading to higher accuracy. Nevertheless, the deep architecture using state-of-the-art models requires large datasets and computing power to be training. Their predictions are hard to explain, and they can perform poorly on unseen data or adversarial examples.

In this work, we take a hybrid approach to combine the strength of both, with a focus on discrete choice modeling (DCM). The goal is to demonstrate how we can use the DCM framework to provide interpretable predictions thanks to knowledge-based concepts while using a neural network architecture to replace hand-crafted functions to improve prediction accuracy. We also demonstrate how we can leverage data-driven techniques to handle dataset imbalance by adapting the DCM choice set, generating new examples, and estimating parameters of discrete choice models. While other hybrid approaches [6, 13, 14] use a neural network to express functions of the utility input missed by the expert modeler, express complex non-linear combinations, or increase the precision of a discrete choice using

a residual computed using a neural network [11]. We propose to use neural networks to replace any complex function used to design a concept in the DCM utility (transforming the constrained input into a score for each possible choice). We build a neural network for each concept described in the utility and constrain the input and output space to limit its prediction and maintain interpretability. We experiment with our method on the Trajnet++ framework using ORCA-generated synthetic datasets. We compare our hybrid model to a classic DCM [2, 12], to an input-constrained non-interpretable model, and to an unconstrained neural network to illustrate the benefits and costs of utility concepts constraints and interpretability.

2. Related work

We focus here on the human-to-human interactions in the context of human trajectory forecasting. Our problem does not include a computer vision component, nor tackle human-to-environment interactions. We group the research on social interactions into the two aforementioned paradigms and a combination of them. Knowledge-driven methods using expert design, and data-driven methods using deep neural networks.

2.1. Knowledge driven

The Social Forces [7] model forecast the next speed using attractive forces (towards a destination or a group of people) and repulsive forces (stay away from obstacles). The Reciprocal Velocity Obstacle (ORCA) [15] model uses collision-avoidance reasoning with the assumption that different agents all behave similarly. Discrete choice modeling [3] uses a grid to represent the space and select the subsequent action for each individual. Using a hand-designed score for each grid cell, this model focuses on specific interactions and provides high interpretability. These methods based on expert-designed functions provide interpretable results and are robust. Though, they don't capture all the complexity of interactions, are limited to the existing domain knowledge, and make predictions in the one-step horizon. They have low prediction accuracy compared to the data-driven methods discussed below, especially on multiple steps predictions.

2.2. Data driven

The research in human trajectory forecasting using neural networks (NNs) has developed fast in the last few years and has outperformed knowledge-driven methods. Using a Long short-term memory (LSTM) [8] network with feedback connections to capture time-series input, the SocialLSTM [1] model introduced a social pooling component that captures interactions with nearby pedestrians over time. More recent NN architectures also capture social interactions in time [5] using a Generative Adversarial Network (GAN) [5] or [4, 9, 17] using an attention mechanism [16]. These methods provide state-of-the-art results for multi-modal trajectory forecasting. While their predictions are highly accurate, they are hard to interpret.

2.3. Hybrid

A few works have recently tried to combine the strength of both paradigms, using discrete choice modeling coupled with neural networks to produce high accuracy explainable predictions. Three methods [6, 13, 14] help to discover new functions of the utility's parameters missed by the hand-designed specification using a deep neural network, improving the overall performance while maintaining interpretability under certain conditions. The SocialAnchors [11] propose to combine the output discrete interpretable output of a DCM model with a social interactions module using LSTM that refines the prediction using scene-specific residual.

3. Method

In this section, we define the trajectory forecasting problem and its adaptation to the DCM framework. We describe our proposed hybrid architecture and the three baselines we use to evaluate our approach. We also describe the constraints on NN input and output to preserve interpretability.

3.1. Problem statement

We tackle the problem of human trajectory forecasting in crowds, with a focus on social interactions. We define a scene as a 2d plane (top-view of a flat surface) with T time steps (also called frames). For each frame, we know the position coordinates of all pedestrians in the scene. For n pedestrians, a scene is defined as a list of trajectories $X = [X_1; X_2; :::; X_n]$. For a person i, a trajectory is defined as $X_i = (x_i^t; y_i^t)$ for time $t = 1; 2; :::; T$ a list of position coordinates.
Using the above definition, the problem is the following. Given all pedestrians' trajectories $X_i = (x_i^t; y_i^t)$ for $t = 1; 2; :::; t_{obs}$ (the observed frames) and the future ground truth $Y_i = (x_i^t; y_i^t)$ for $t = t_{obs} + 1; :::; T$, predict the future trajectories of all pedestrians $\hat{Y} = \hat{Y}_1; \hat{Y}_2; :::; \hat{Y}_n$ where $\hat{Y}_i$ is the predictor trajectory of pedestrian i.

3.2. Discrete choice models

Discrete choice models try to forecast the behavior of an individual in a choice situation. Using the random utility maximization, we assume that each alternative can be associated with a score called utility. For each choice situation, we select the alternative with the highest utility. The general formulation of the utility is, for an alternative j and a decision maker n:

$$U_{in} = V_{in} + \quad _{in} \tag{1}$$

where the deterministic part of the utility V_{jn} is a function of its attributes, and ε_{in} is a random term representing the uncertainty deriving from the presence of unobserved attributes.

3.2.1 Choice set

In the context of human trajectory forecasting, [3] defines a dynamic and individual-based choice set (see Fig. 1) and discretizes the space in front of a pedestrian based on her current speed direction and norm. Three speed regimes are assumed from the current velocity: accelerated, constant, and decelerated corresponding to fractions of the current speed. The direction angle is chosen in a set of 11 radial directions described in Fig. 1. The choice set is a combination of speed regime and radial direction making a total of 33 choices for a given pedestrian and time. We refer to the selected choice as the anchor (speed vector corresponding to the choice).

Figure 1. Angular and speed regime choice based on current speed direction and norm. (Source: [12])

3.2.2 Utility specification

The deterministic part of the utility is defined (using specifications from [2, 12]) as follows:

$$
\begin{aligned}
U_{jn} = & \underbrace{\beta_{dir}\, dir_j}_{\text{keep direction}} + \underbrace{\beta_{ddist}\, ddist_j + \beta_{ddir}\, ddir_j}_{\text{towards destination}} \\
& + \underbrace{\beta_{acc}\, I_{j;acc}\, ff_{n;acc} + \beta_{dec}\, I_{j;dec}\, ff_{n;dec}}_{\text{free flow}} \\
& + \underbrace{\beta_{acc}\, lead_{j;acc} + \beta_{dec}\, lead_{j;dec}}_{\text{leader-follower}} \\
& + \underbrace{\beta_{col}\, col_j}_{\text{avoid collision}} + \underbrace{\beta_{odist}\, odist_j + \beta_{odir}\, odir_j}_{\text{avoid occupancy}}
\end{aligned}
\tag{2}
$$

where U_{jn} is the deterministic part of the utility for pedestrian n and choice j. The β parameters have to be estimated, and dir_j, $ddist_j$, $ddir_j$, $ff_{n;acc}$, $ff_{n;dec}$, $lead_{j;acc}$, $lead_{j;dec}$, and col_j corresponds to utility concepts described in [12]. The occupancy parameters $odist_j$ and $odir_j$ corresponds respectively to occupancy and angle concepts from [2].

The underlying concepts of this specification are the following:

- keep direction: Pedestrians tend to maintain their direction
- towards destination: A pedestrian wants to minimize the distance to the destination and the angle between the current direction and the destination's direction
- free flow: In free flow conditions (no social interaction), the desired speed drives the speed regime choice. The maximum speed v_{max} is used as the reference for the pedestrian's desired speed
- leader-follower: Pedestrians tend to follow the tracks of people heading in the same direction. The relative speed of a leader with respect to a pedestrian impacts her choice to accelerate or slow down
- avoid collision: When a neighboring pedestrian is heading towards a choice, it becomes less desirable to avoid a collision
- avoid occupancy: A radial choice containing a neighbor is less desirable, especially if the neighbor is close to the pedestrian

3.3. Problem adaptation

The DCM principles limit the information used for the trajectory prediction task to a frame's state. The utility specification described above adds constraints to the input of our problem, and the discretization of the space in a choice set also constrains the output. Our initial problem Sec. 3.1 is limited to a one-step prediction task (we don't know the history of positions and speed, and we predict only the next speed). We reformulate it within this DCM framework as:

We define a frame $F_t = [P_1^t; P_2^t; :::; P_n^t]$ as a list of pedestrians in state P_i at time t. For a pedestrian i, a state is defined as $P_i^t = [(x_i^t; y_i^t); (vx_i^t; vy_i^t); (gx_i^t; gy_i^t)]$ where x_i and y_i are the position coordinates, vx_i and vy_i are the speed coordinates, and gx_i, gy_i the destination (position for $t = T$) coordinates.

The task is the following: Given all pedestrian states for a given frame F_t, and the ground truth next speed of the primary pedestrian $(vx_1^{t+1}; vy_1^{t+1})$, predict the anchor (choice) a_1^t corresponding to the closest approximation (example: Fig. 3) of this true next speed.

3.4. Neural network constraints

When replacing a hand-designed function with a neural network, we need to make sure that the model learns the concept for which it is supposed to replace the function. We also limit the model prediction to the choice set to t within the DCM framework.

3.4.1 Input constraints

Our models use the data from the six concepts defined in Sec. 3.2.2 to explain the pedestrian trajectory. We suppose that limiting the input of each utility concept' NN to only what the hand-designed function needs will ensure that the NN learns the concept only. If one concept's NN learns multiple other concepts, the overall model would not be interpretable as each neural network score would not correspond to a given concept.

We describe here what processing and selection process each concept applies to the original input, to limit the information to what is necessary for the hand-crafted functions of [2, 12].

- keep direction: The angle between an anchor and the current speed direction
- towards destination: Distance between an anchor position and the destination, and angle between the anchor direction and the destination direction
- free flow: The ratio between the current speed and the maximum speed
- leader-follower: The distance between the pedestrian position and each neighbor's position. The angle between an anchor direction and neighbors' direction. The difference between the anchor speed and neighbors' speed norm. A potential leader (Fig. 2) indicator function (for each anchor: neighbors that are too far, not going in a similar direction, or not positioned in the anchor's direction cone are not potential leaders)
- collision: The distance between an anchor position and each neighbor's position. The angle between an anchor direction and each neighbor's direction. The sum of the anchor speed and neighbor speed norms. A potential collider indicator function (for each anchor: neighbors that are too far, not going in the opposite direction, or not positioned in the anchor's direction cone as not potential colliders)
- occupancy: The distance between an anchor position and each neighbor's position. The angle between an anchor direction and each neighbor's direction. A cone indicator function (for each anchor: neighbors that are not in the direction cone are ignored)

3.4.2 Output constraints

The DCM framework requires us to predict an anchor from the choice set, transforming the problem of next speed prediction into a classification problem. The model chooses between the possible speed regimes and radial angles by predicting a score for each of the 33 available choices. We select the anchor with the maximum score as the prediction. The prediction is accurate if the chosen anchor corresponds to the closest choice to the pedestrian's true next speed (Fig. 3).

Figure 2. Leader selection. Anchor direction cone between d_l and d_r. (Source: [12])

Figure 3. Ground truth anchor is the closest choice to the pedestrian's next speed

3.5. Models

We present in this section the models implemented to evaluate our proposed approach (ConceptNnDCM) and understand the benefits and costs of using the DCM input preprocessing and having interpretable results. We implemented four models with different properties that all follow the DCM framework (they are limited to a one-step input and predict the anchor corresponding to the primary pedestrian's next speed). The LearnDCM (original DCM),

UtilityNnDCM (non-interpretable), and NnDCM (no utility input processing) models are baselines to demonstrate the impact of using neural networks, having interpretable concepts, and using utility input processing.

3.5.1 LearnDCM

The LearnDCM (Fig. 4) corresponds to an extended version of the original DCM implementation [2, 12]. It is a classic discrete choice model using hand-designed functions. The and exponential parameters are estimated using gradient descent with a cross-entropy loss. The exponential parameters are initialized using the SocialAnchors DCM [11] and can be frozen (original DCM speci cation) or estimated during training. The model returns anchor scores for each utility concept that we can interpret.

Figure 4. LearnDCM model architecture

3.5.2 ConceptNnDCM

The ConceptNnDCM (Fig. 5) is our proposed approach. It replaces the hand-designed functions of the classic DCM with a neural network for each concept. This model uses the same input as the LearnDCM. Instead of the original functions, each concept uses a small fully-connected neural network fed with the concept's processed input. Similarly to the LearnDCM, each NN return anchors scores for the corresponding concept and can be visualized and interpreted.

Figure 5. ConceptNnDCM model architecture

3.5.3 UtilityNnDCM

The UtilityNnDCM (Fig. 10a) uses the same input as the ConceptNnDCM. Nevertheless, it combines all inputs before feeding them to a single neural network. The output is an overall anchors score. The model output is not interpretable and aims at evaluating the impact of interpretability.

3.5.4 NnDCM

The NnDCM (Fig. 10b) is a single neural network model that does not use the utility concepts of input processing. It uses the initial frame data (positions, current speed, and primary pedestrian destination) and returns the overall anchors score. It aims to demonstrate the impact of utility processing.

4. Experiments

We describe here the experiments done to show how data-driven techniques help us design the choice set and handle dataset imbalance when estimating models' parameters. We observe the importance of the different concepts between the original (LearnDCM) and our (ConceptNnDCM) approach. Finally, we evaluate how using a NN in our model impacts interpretability and performance.

4.1. Dataset

We use a synthetic dataset within the interaction-centric Trajnet++ framework [10], generated using ORCA [15]. The dataset corresponds to about 5 million frames extracted from 50 thousand scenes with 4, 5, or 6 pedestrians starting around a circle and going in opposite directions. We do not evaluate the 'leader-follower' concept due to the dataset limitations (ORCA does not model leader-follower interactions). No collisions occur in both the training and testing set.

4.1.1 DCM choice set de nition

The original choice set speci cations [2, 12] are based on a dataset different from ours. We thus need to adapt our choice set to our dataset. We want to have suf cient precision between the angle and speed choices to predict the decision-making process accurately. We keep the 33 choices (11 radial choices, and 3-speed choices) from the original implementation [3].

Speed choices Speed choices try to model the pedestrian's ability to increase/reduce speed in free ow or due to social interactions. The constant speed regime is the most frequent, usually when no interaction occurs. The original speed regime (0.5, 1, 1.5) corresponds to a high change in

speed cardinality that rarely occurs in our dataset, leading to each choice corresponding to respectively (0.7%, 98.1%, 1.2%) of the data. Indeed, a change of 50% in speed in a frame (given 2.5 frames per second) is excessively high. Using the distribution of speed norm (Fig. 6), we update the speed regimes to (0.9, 1, 1.1) regimes splitting the dataset into (10.1%, 80.3%, 9.6%) groups. We base our selection on a trade-off between a large enough change of speed and not too small accelerated and decelerated groups, given that the most observed values are near constant speed.

Figure 6. Distribution of speed norm

Angle choices Angle choices re ect the change of direction to reach the goal or avoid another pedestrian. In most cases, the pedestrian keeps her trajectory straight (center choice) as she tries to reach her destination using the shortest path. The original angle choices centers are (-72.5°, -50°, -32.5°, -20°, -10°, 0°, 10°, 20°, 32.5°, 50°, 72.5°) with a span of (25°, 20°, 15°, 10°, 10°, 10°, 10°, 10°, 15°, 20°, 25°). It splits the dataset into the following groups (0.01%, 0.04%, 0.26%, 1.02%, 4.76%, 87.8%, 4.78%, 1.02%, 0.26%, 0.04%, 0.01%). The group centered at 0° contains 88% of the dataset values. This is an issue, we observe many scenes where collision avoidance interactions occur, but the change in angular direction is smaller than 5°. Because of the precision of anchors, all changes smaller than 5° correspond to the central anchor. We, therefore, update the angle centers and span to increase the precision of the anchor towards the center. Based on the next speed absolute angle distribution (Fig. 7), we select the centers (-65°, -30°, -15°, -7°, -2.5°, 0°, 2.5°, 7°, 15°, 30°, 65°) and span of (45°, 20°, 10°, 6°, 3°, 2°, 3°, 6°, 10°, 20°, 45°), splitting the dataset into (0.05%, 0.59%, 2.10%, 4.71%, 21.83%, 41.47%, 21.83%, 4.72%, 2.10%, 0.59%, 0.05%). This new angular choice set preserves the same number of choices and the overall eld of view (170°).

Figure 7. Distribution of absolute angle (in degrees) between current and next speed

4.1.2 Data imbalance

Once our choice set is de ned, we can compute the dataset labels. We sample the labels for each frame using the primary pedestrian's next speed. The target anchor is the anchor with the smallest euclidean distance to the speed. Even after building an adapted choice set for our dataset, the classes (anchors indices) or our problem are still heavily imbalanced (see Tab. 1).

Anchor index	Angle	Speed	Dataset %
16	0°	1	47.5
17	2.5°	1	14.7
15	-2.5°	1	14.6
22	-65°	1.5	0.0074
11	-65°	1	0.0021
21	65°	1	0.0018

Table 1. Train data top and bottom 3 classes

An imbalanced dataset is an issue for training our neural networks (and DCM parameters). During the gradient descent, the model will see many more examples of some classes, and learn them better (at the cost of rare classes). We rst use a weighted cross-entropy loss function. The weights are inverse of the class counts, giving a higher penalty on errors made for a less frequent class. Using them, the model will learn each class equally regardless of its cardinality. Though, this method does not help classes that do not have enough examples to be properly learned. To deal with this limitation, we implemented a second technique taking advantage of the synthetic nature of the data. We generated many new scenes and kept only the examples from rare classes. We build a new train dataset with arti - cially more examples for rare classes. This method helped improve the overall learning (especially the accuracy of rare classes) with a small increase in the train dataset size.

4.2. Concepts importance

The concepts de ned by the DCM utility try to describe the multiple and independent elements that intervene in human walking decision-making. These concepts help us decompose what is driving a choice. Nevertheless, for the interpretation of these concepts to be meaningful, they need to have a suf cient impact on the overall anchors' score and affect the nal choice. In this section, we evaluate the importance of each concept for the LearnDCM and ConceptNnDCM models.

For a single example, we evaluate a concept's importance using the following metrics:

- Score delta between rst and last anchors (FL)
- Score delta between the rst and second anchor (FS)
- Variance between anchor scores (V)

We compute for each metric the average value per concept over all frames. This helps us understand the overall impact of a given component. We also compute the percentage of the test dataset for which a given concept is the most important (highest metric value). These percentages help us understand how often a concept is dominant compared to others.

We present a summary of our results for the LearnDCM model in Tab. 3. We observe that the destination concept is, by far, the most important and solely determines the vast majority of predictions (Highest variance for 100% of frames). The average variance shows us that the destination concept has an order of magnitude greater than all other concepts, highlighting its importance. The collision and occupancy concepts have the highest rst-second anchor score different in about 1% of cases, mostly when the subject is far from the destination and has neighbors nearby.

The results for the ConceptNnDCM model are summarized in Tab. 4, we observe from average and percentage metrics that the destination concept is still the most important. Nevertheless, the difference with other concepts is much smaller (compared to LearnDCM). Free ow is the second most impactful concept with the highest variance in 41% of frames. The collision and occupancy are also signi cant in about 10% of frames each. We also note that the direction concept produces only zero scores. This is due to the arti cially balanced dataset in training, which leads to no difference in frequency between classes. We discuss this issue further in the next section.

Overall, the ConceptNnDCM seems to make better use of the different concepts than the LearnDCM. Indeed, they play a greater role in the nal prediction and are all (expect direction) dominant for some frames. We believe that it is due to the ConceptNnDCM model using information from the input that hand-designed functions do not describe. This difference can also be due to the higher representation power of the NN, leading to a more complex combination of each concept's input.

4.3. Interpretable predictions

When replacing the hand-written functions with NNs, we increased the representation power of our model. Nevertheless, as we discussed in the Sec. 3.4, this can be at the cost of prediction interpretability if the problem is not constrained and the neural networks learn overlapping concepts. Our initial hypothesis is that we can preserve the interpretability by limiting the input of each concept's NN to the data used by the hand-written functions.

To evaluate interpretability, we visualize the anchors' scores overall dataset and on selected examples. We present here the results from LearnDCM and ConceptNnDCM on an example frame (Fig. 8) with collision avoidance interactions. In this example, the true anchor index is 3: a speed decrease and a turn to the right. In this example, the LearnDCM and ConceptNnDCM respectively predict anchors 4 and 3, both corresponding to a speed decrease and a right turn.

Figure 8. Position and speed of pedestrians in a selected frame

The LearnDCM score is mostly driven by the destination concept. We observe meaningful activation of direction, free ow, and occupancy concepts. The collision concept surprisingly returns a positive value in the cones with potential colliders. We believe this is due to the dataset: the pedestrians start in a circle and all go towards the center. Their collision avoidance choice usually makes them go towards the neighbors' direction (or near it) as the neighbors move towards the center.

The ConceptNnDCM activation has three main concepts (collision, free ow, and destination), while the direction

and occupancy concepts have no activation. For the direction concept, this issue is present on the whole dataset and is due to the train dataset balancing. A possible fix would be to use the direction concept from LearnDCM or train independently on the original unbalanced data. Training this concept independently on unbalanced data can also help. Regarding occupancy, we observe that the model uses either only collision for some examples, or occupancy for others (in this example, collision is used). We believe this is due to the two concepts sharing many processed inputs, and having a similar role (handling repulsive interactions). An adapted fix would be to have a single concept for all collision avoidance behavior learned from the social environment. Otherwise, we would need to ensure that the collision and occupancy concepts do not overlap. For instance, we could use an indicator function based on distance such that the occupancy handle short distance cases and the collision one handles longer distances. Finally, we observe in Fig. 9 that the collision concept correctly penalizes directions in which some potential colliders are present.

(a) LearnDCM (b) ConceptNnDCM

Figure 9. Collision scores map on example frame

In summary, both models' concepts' activation being dependent on their limited input, they show some similarities in the activation. For instance, the free flow concept incentivizes in both models the choice of a reduced speed. Nevertheless, some concepts fail to be learned in the ConceptNnDCM model (direction, occupancy). In this work, we limited the constraints applied to NNs to the input. Nevertheless, we can likely enhance the interpretability using additional constraints. On the output, we could constrain the sign (occupancy or direction should be negative). Also, the output to the angle choices only for the direction concept, and the speed regimes only for the free flow will help the model learn better the concepts. Alternatively, a custom loss designed individually for each concept (for instance the angular error only for the direction concept) would narrow down the task each network is supposed to solve. Finally, training (or pretraining) each concept individually on an adapted subset of the dataset (ex: unbalanced frames without interaction for the direction concept) could also improve the interpretability. Numerous limitations of the two models are also due to the synthetic dataset and the limited set of interactions. Extensive testing on other synthetic and real-world datasets would help better understand what each NN is learning.

4.4. Predictive performances

The main goal of replacing hand-crafted functions with neural networks is to increase the predictive power. We evaluate here the performances of our proposed approach (ConceptNnDCM) against the three baseline models (LearnDCM, UtilityNnDCM, NnDCM) using the following metrics:

Metrics

- Accuracy (Macro / Weighted)
- F1 score (Macro / Weighted)
- Prediction distance (Distance between the true speed and the predicted anchor)

Note: Macro (M) corresponds to computing the metric value for each class and averaging the results over all classes regardless of the size of each class. Weighted (W) takes into account imbalance in the dataset using a weighted average between classes.

Model	Acc M/W	F1 M/W	Pred distance
LearnDCM	0.234/0.273	0.091/0.327	0.047
ConceptNnDCM	0.321/0.268	0.164/0.290	0.039
UtilityNnDCM	0.277/0.240	0.144/0.286	0.044
NnDCM	0.213/0.206	0.099/0.247	0.0514

Table 2. Models performances

Results We observe that all models with utility input processing outperform the NnDCM (Tab. 2) model with raw frame data. The features computed by the utility such as relative distances and angle to neighbors or the destination could be the explanation, as they might be hard to learn for a shallow model like the NnDCM.

The ConceptNnDCM and UtilityNnDCM outperform the classic LearnDCM model on Macro metrics and distance. The macro metrics highlight the impact of data balancing methods that improved prediction accuracy for rare classes. The confusion matrices (Fig. 15, Fig. 16) shows that all models learn all classes evenly (center diagonal). Though, the LearnDCM fails to predict the speed regime (almost always predicts the constant speed) while it succeeds for the angles (Fig. 15a).

The ConceptNnDCM outperform both UtilityNnDCM and NnDCM. It is surprising as we expected that using separated models for each concept or using utility processed input would negatively impact the performances. These results can be explained by the limited amount of rare examples, even with the additional data generation. Also, all NN models are shallow. Therefore, designing the best features has an important impact on how the NN model learns.

Overall, the performances of all models are poor: 32%/27% (Macro/Weighted) accuracy. We believe this is due to the following reasons. First, the limitations of the DCM framework: the choice set precision (the distance between the true anchor and true next speed is 0.0116, about a fourth of the models' distance error), and the limitation to current frame information (instead of all the observed passes frames). The limited number of neural network parameters and a small dataset with few diverse examples can lead to the failure to learn the complex patterns of direction and speed decision-making.

5. Conclusion

In this work, we built a classic DCM model and adapted it to the Trajnet++ framework to serve as a baseline for our work and other models. We show how using data-driven techniques can help us handle class imbalance during training by adapting anchor precision, generating more rare examples, and using weighted loss to improve the learning of our models. We evaluated how each DCM concept impacts the final prediction, highlighting the importance of the destination concept in the original DCM, and showing how the ConceptNnDCM concepts have a more balanced impact. Finally, we show how replacing the hand-design DCM functions with a neural network improves the model performances while preserving some of the interpretability. We finally discuss the limitations of this interpretability based on the constraints we apply to the input, output, and loss, and the performance limitations of all our models within the DCM framework.

6. Future work

We discuss here the potential improvements and further works to deal with limitations observed in this work, evaluate the models on a real-world dataset, and improve overall performances and interpretability.

Choice set Evaluate how increasing the number of choices or the precision of each choice affects performances and at what cost.

Dataset Train and test models on another synthetic dataset that is not limited to collision interactions. Also, evaluate models on real-world datasets from the Trajnet++ framework to observe the impact of the leader-follower concept.

Interpretability Implement a custom loss function for each DCM concept to train both LearnDCM and ConceptNnDCM to improve predictions (especially on the speed regimes). Add new constraints on outputs for the concepts limited to only angular or speed choices. Train the concept's NN independently on a selected subset of the data to improve interpretability and independence between concepts. Combine the concepts from LearnDCM and ConceptNnDCM, keeping the most basic concepts like direction and free flow as hand-designed functions, while having more complex concepts like collision and leader-follower use a neural network. Finally, we used in this work both collision and occupancy concepts. These two concepts share some of their input overlapping input. Implementing tighter constraints to ensure their independence, or combining them into a single concept would ensure they are not conflicting.

Concepts Using the NN-based models, we observed a correlation between the speed and angular choice in free flow conditions. Using a concept combining both information could better model walking behavior outside of social interactions. Similarly, we could combine the collision and occupancy concepts as they tackle a very similar task. NN models could also be used to process multi-step inputs (history of all pedestrians' positions and speed) instead of the current frame information only. Enriching the input using passed frames would likely lead to better predictions by giving the model a chance to detect movement patterns on multiple steps (inertia).

Comparison and combination with NN Using deeper NN architecture, trying to use attention mechanisms and feedback connections (while enriching input with historical information) would like to improve NN model performances. Replacing the L-MNL DCM block by ConceptNnDCM could also be interesting to have a more competitive model compared to data-driven approaches while keeping some interpretability. Finally, comparing this hybrid approach to state-of-the-art data-driven models on multi-step predictions.

References

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–971, 2016. 1, 2

[2] Gianluca Antonini, Michel Bierlaire, and Mats Weber. Discrete choice models of pedestrian walking behavior. *Transportation Research Part B: Methodological*, 40(8):667–687, 2006. 2, 3, 4, 5

[3] Gianluca Antonini, Santiago Venegas, Jean-Philippe Thiran, and Michel Bierlaire. A discrete choice pedestrian behavior model for pedestrian detection in visual tracking systems. IEEE, 2004. 1, 2, 3, 5

[4] Francesco Giuliari, Irtiza Hasan, Marco Cristani, and Fabio Galasso. Transformer networks for trajectory forecasting, 2020. 2

[5] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks, 2018. 1, 2

[6] Yafei Han, Francisco Camara Pereira, Moshe Ben-Akiva, and Christopher Zegras. A neural-embedded choice model: Tastenet-mnl modeling taste heterogeneity with flexibility and interpretability, 2020. 1, 2

[7] Dirk Helbing and Péter Molnár. Social force model for pedestrian dynamics. *Physical Review E*, 51(5):4282–4286, may 1995. 1, 2

[8] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 2

[9] Vineet Kosaraju, Amir Sadeghian, Roberto Martín-Martín, Ian Reid, S. Hamid Rezatofighi, and Silvio Savarese. Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks, 2019. 2

[10] Parth Kothari, Sven Kreiss, and Alexandre Alahi. Human trajectory forecasting in crowds: A deep learning perspective, 2020. 5

[11] Parth Kothari, Brian Sifringer, and Alexandre Alahi. Interpretable social anchors for human trajectory forecasting in crowds, 2021. 2, 5

[12] Th. Robin, G. Antonini, M. Bierlaire, and J. Cruz. Specification, estimation and validation of a pedestrian walking behavior model. *Transportation Research Part B: Methodological*, 43(1):36–56, 2009. 2, 3, 4, 5

[13] Brian Sifringer, Virginie Lurkin, and Alexandre Alahi. Enhancing discrete choice models with neural networks. In *hEART 2018–7th Symposium of the European Association for Research in Transportation conference*, number CONF, 2018. 1, 2

[14] Brian Sifringer, Virginie Lurkin, and Alexandre Alahi. Enhancing discrete choice models with representation learning. *Transportation Research Part B: Methodological*, 140:236–261, 2020. 1, 2

[15] Jur van den Berg, Ming Lin, and Dinesh Manocha. Reciprocal velocity obstacles for real-time multi-agent navigation. In *2008 IEEE International Conference on Robotics and Automation*, pages 1928–1935, 2008. 1, 2, 5

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[17] Anirudh Vemula, Katharina Muelling, and Jean Oh. Social attention: Modeling attention in human crowds, 2017. 2

Appendix

Models architecture

(a) UtilityNnDCM

(b) NnDCM

Figure 10. Models architecture

Concepts importance

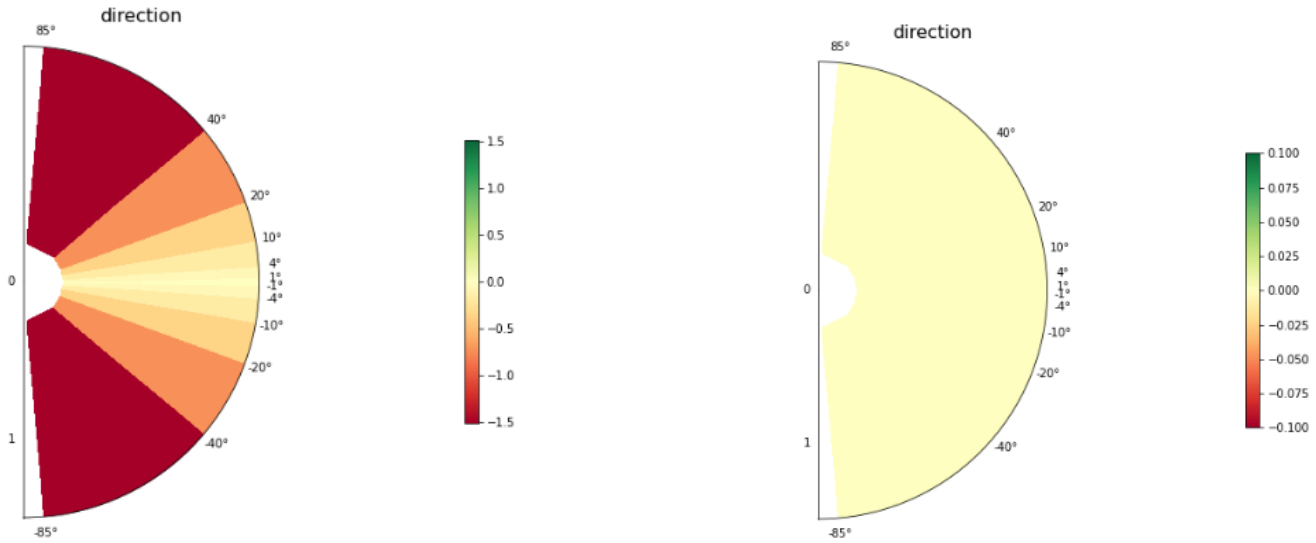
	Direction	Destination	Free flow	Collision	Occupancy
Average first-last span (FL)	1.51	**10.91**	1.01	0.20	0.30
Average first-second span (FS)	0	**0.24**	0.0	0.0046	0.00010
Average variance (V)	0.28	**11.37**	0.23	0.0084	0.018
Maximum FL %	0	**100**	0	0	0
Maximum FS %	0	**99.1**	0	0.89	0.021
Maximum V %	0	**100**	0	0	0

Table 3. Utility concepts importance metrics - LearnDCM

	Direction	Destination	Free flow	Collision	Occupancy
Average first-last span (FL)	0	**8.07**	6.58	2.75	1.58
Average first-second span (FS)	0	**0.66**	0.35	0.12	0.0039
Average variance (V)	0	**6.31**	3.04	1.13	1.12
Maximum FL %	0	**46.67**	36.14	7.97	9.22
Maximum FS %	0	**57.68**	35.53	6.35	0
Maximum V %	0	40.15	**40.91**	7.73	11.20

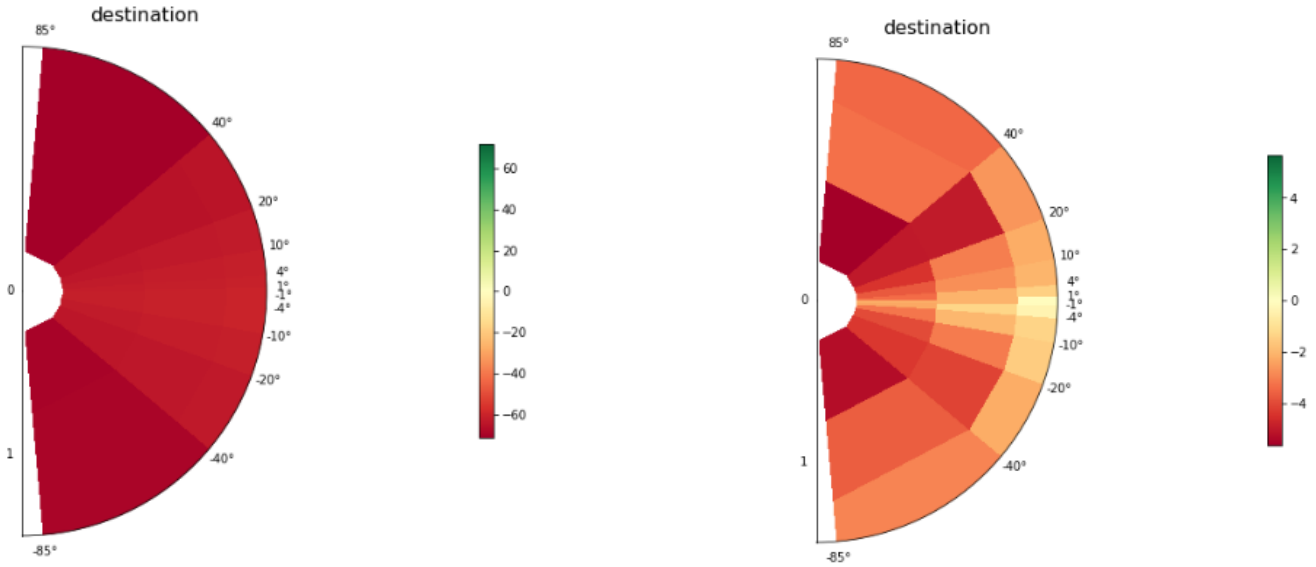
Table 4. Utility concepts importance metrics - ConceptNnDCM

Interpretability



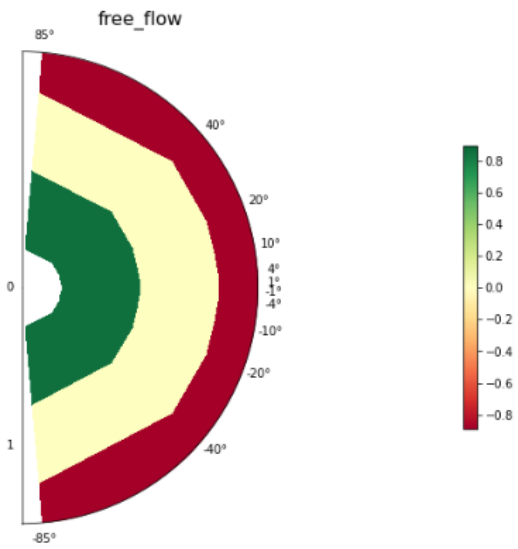
(a) LearnDCM (b) ConceptNnDCM

Figure 11. Direction scores map on example frame

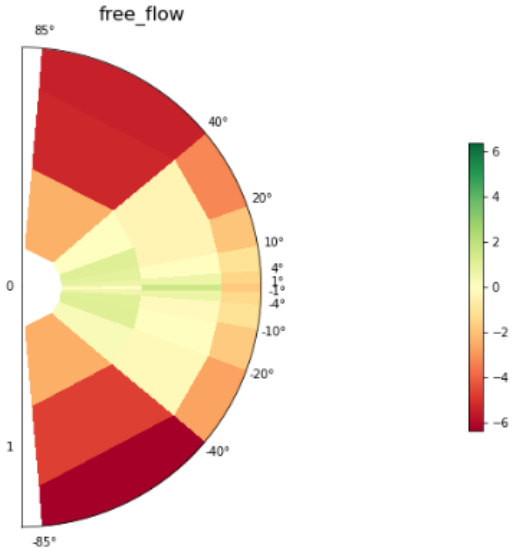


(a) LearnDCM (b) ConceptNnDCM

Figure 12. Destination scores map on example frame

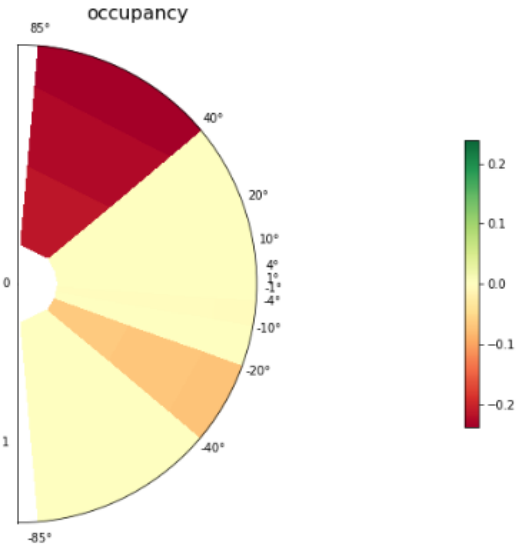


(a) LearnDCM

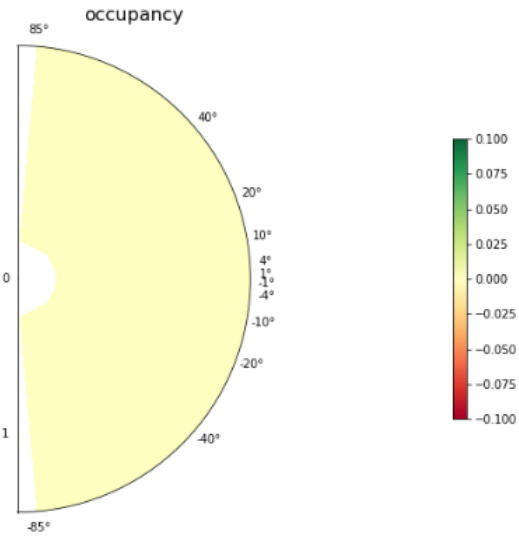


(b) ConceptNnDCM

Figure 13. Free flow scores map on example frame



(a) LearnDCM



(b) ConceptNnDCM

Figure 14. Occupancy scores map on example frame