Blending computer vision into discrete choice models

Sander van Cranenburgh
Transport and Logistics Group, Delft University of Technology

1. Introduction

Since its inception in the 1970s, Discrete Choice Models (DCMs) have become a key methodology in the transportation (as well as in various other fields, such as environmental economics, marketing and health economics) (Hess and Daly 2014). The vast majority of DCMs is grounded in utility theory, which postulates that decision-makers make choices by evaluating the utility of each available alternative and choose the maximum utility alternative (McFadden 1974). This behavioural foundation gives behavioural meaning to DCM's modelling outputs, and enables it to deduce economic evaluations, such as Willingness-to-Pay estimates, from observed choice behaviour. Given the paramount role of DCMs in underpinning macro-level transport policies, researchers are continuously striving to improve the modelling paradigm's behavioural realism and reach.

Notwithstanding these continuing efforts, current DCMs quite literally suffer from a blind spot: they cannot handle visual information. As a result, DCMs are of limited value when used to explain choice behaviour that involves visual stimuli, such as when decision-makers book a tourism destination, or a hotel room online. This blind spot in current DCMs hampers a deeper understanding of human choice behaviour in the presence of visual stimuli.

In the last decade major developments have taken place in Computer Vision (CV). State-of-the-art CV models are able to accurately detect scenes and objects in images (Gu et al. 2018). Nowadays, CV is used in a vast range of applications from detection faults in production processes to detecting tumours in MRI scans. Moreover, many pre-trained CV models are currently available. Such models have been trained on large amounts of data and can be further trained, using limited amounts of data, to conduct new related tasks. Thereby, the computational burden and the need for large amounts of data is substantially reduced .

This research aims to bring visual information to the realm of choice modelling, by blending computer vision into DCMs. In this study we compare and empirically test the performance of a variety of ways to do this. In these models visual information is fed to a Convolution Neural Network (CNN) of a CV model (LeCun et al. 1989), from which the feature map (representing the images) is extracted. This feature map, in turn, enters the utility function of a DCM as explanatory variables. In the absence of a suitable existing data set, i.e. a data set consisting of choices made in the context of visual stimuli, we create one ourselves by emulating choices. In this emulated data set, decision-makers are faced with a binary choice situation, in which each alternative consists of an image and price tag. The decision-maker needs to trade-off the image aesthetic value and the price of 'purchasing' it. The images and their associated aesthetic ratings are taken from the AVA data set (Murray et al. 2012). Images in this data set are rated on their aesthetic value by several hundreds of people. In consonance with Random Utility Theory, decision-makers are assumed to experience positive utility from the aesthetic value of the image, $\beta_{rating} \cdot \text{rating}$, and negative utility from the price, $\beta_{price} \cdot \text{price}$. When training the proposed models, they are fed with the images and the price, but not the ratings. Hence, the learning task essentially is to learn the aesthetic value of images, and the way in which it is traded-off against price.

2. Methodology: blending computer vision into discrete choice models

2.1. Conceptual framework

Most CV models conceptually consists of two parts: a Convolution Neural Network (CNN), which extracts features from the image, and a (object) classifier (see Figure 1). The feature extractor typically comprises of a series of convolution layers which ultimately map the image onto a lower dimensional space, called the feature map. The object classifier, typically an Artificial Neural Network, in turn classifies the image based on the extracted features.

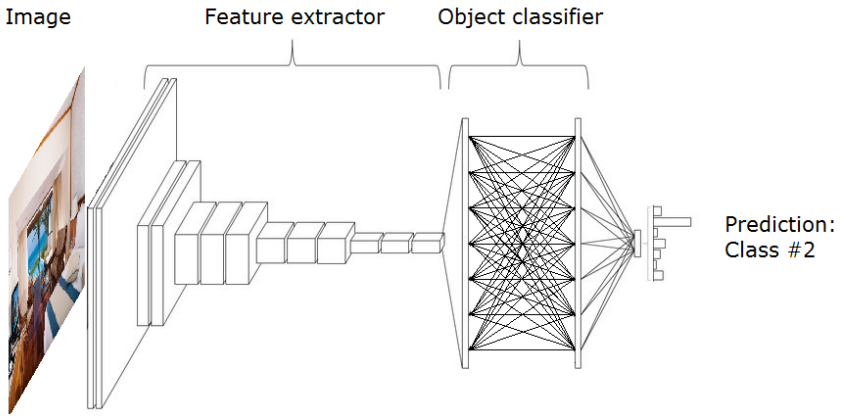


Figure 1: Typical CV model

In our modelling framework we treat the feature map extracted from the CNN as explanatory variables that enter the utility function of our DCMs, just like e.g. travel cost would. The observed part of utility V_i is assumed to take a linear-additive form, both for the utility associated with numeric attributes m as well as for the utility associated with the image's features k, see Equation 1, where x_{ikn} denotes the k^{th} feature for the image of alternative i, and w_k denotes the weight associated with feature k. Another possibility is to encode the feature space extracted from the CNN into another (lower dimensional space) feature space, using an ANN. The rationale to do this is that the aesthetic value of an image might not only be a linear-additive function of the features, but could also be caused by particular interactions between features.

Figure 2 provides visualisations of both approaches. The left-had side plot depicts the situation in which the feature space directly enters the utility function; the right-hand side plot depicts the situation in which the feature space is first fed to an ANN, before it enters the utility function. Note that the utilities of the left and right alternatives are depicted in Figure 2 by the upper and lower rectangular blocks on the right-hand sides, respectively. Importantly, the upper and lower parts of the network are fully identical: both in architecture as well as in the weights. In some ways, the proposed model architecture is very similar to Siamese architectures (Bromley et al. 1994), which are typically used for tasks like determination of whether, or not, two images belong to the same class (e.g. depict the Eiffel tower). In these networks typically a distance measure is computed between the features spaces of the images, where a low distance implies a large probability that both images belong to the same class. In contrast, in our models we do not compute the distance between features, rather we compute the total utilities.

$$U_{in} = \sum_m \beta_m x_{imn} + \sum_m w_k x_{ikn} + \varepsilon_{in}$$	*Equation 1*

$$U_{in} = \sum_m \beta_m x_{imn} + \sum_m w_k \tilde{x}_{ikn} + \varepsilon_{in}$$ $$where\ \tilde{x}_{ikn} = f(x_{ikn})$$	*Equation 2*

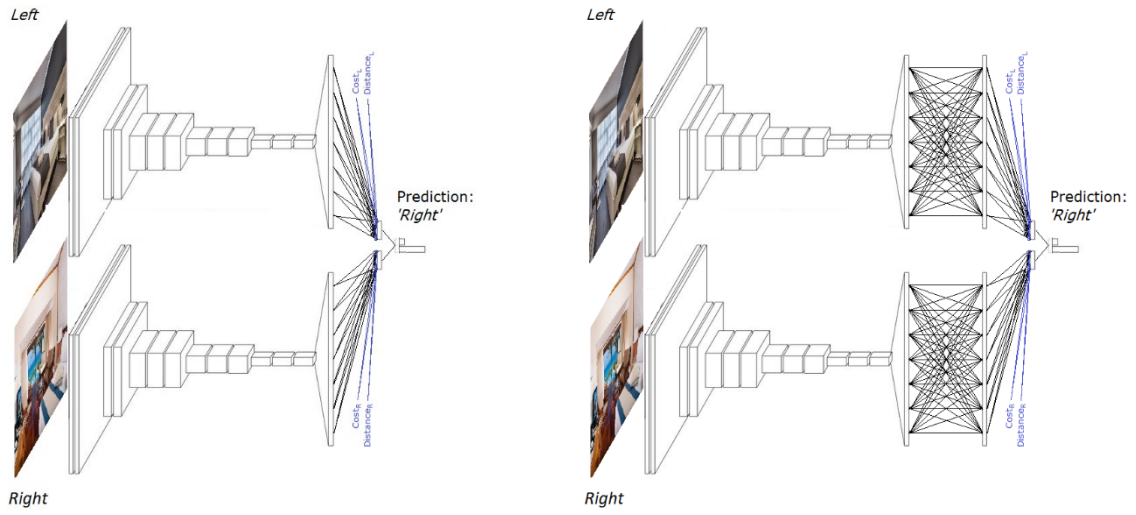


Figure 2: Visualisation of modelling framework

Upon recovering the model parameters (β,w) the weights w are not as interpretable as the classic utilities parameters, β. Although the weights can then still be conceived as marginal utilities –after all they reflect the marginal effect on utility– their interpretation is hampered because the features x_{ikn} themselves do not carry a meaningful behavioural interpretation.

2.2. **Models**

Table 1 provides an overview of the models we estimate and train. In models 1 to 6 the CNN is not trained. That means that the feature space is constructed prior to estimation / training by feeding the images to a pre-trained CNN and store the features as explanatory variables. In contrast, in model 7 and 8 the CNN is trained together with the DCM. This has the advantage that the weights in the CNN can be optimised for the task at hand. However, a drawback is that it also makes the training more complex conduct and computationally considerably heavier.

Models 1 to 4 are conventional DCMs. Model 1 uses the numeric attributes, but ignores the feature map. This model serves as a benchmark to assess the increase in explanatory power when accounting for the visual information. Model 2 uses the feature map, but ignores the numeric attributes. This model reveals the explanatory power of the feature maps of themselves (when they are linearly mapped to the utility function). In model 3 both the numeric and visual information are used. We expect this model to outperform models 1 and 2. Model 4 is similar as model 1, but trained using a SGD algorithm, instead of a quasi-Newton optimisation algorithm, as is common for DCMs.

Unlike models 2 to 3, models 5 and 6 encode the feature space extracted from the CNN first onto another (lower dimensional) feature space using an ANN, before it enters the utility function. Similar as in models 2 to 3, model 5 only uses the feature map; and model 6 uses both the numeric attributes and the feature map. In the case that nonlinearities and complex interactions between the features result in higher (or lower) aesthetic values, then we will see that models 5 and 6 outperform models 2 and 3. Conversely, when the aesthetic value is predominantly a linear-additive function of the feature map with no particularly strong interaction effects, then we will see that models 5 and 6 will perform on par with models 2 and 3. Finally, in models 7 and 8 the CNN and the DCM are jointly optimised. Comparing the performance between these models and the models in which only the DCMs are trained (i.e. models 1 to 6) will reveal whether, or not, the extra efforts and computational complexity of joint training is worth it.

Table 1: Model overview

		Model variables		Feature encoding	Model components trained	Training algorithm
		Numeric	Feature map			
Model	**1**	X		N/A	DCM	Quasi-Newton
	2		X	No	DCM	Quasi-Newton
	3	X	X	No	DCM	Quasi-Newton
	4	X		N/A	DCM	SGD
	5		X	ANN	ANN & DCM	SGD
	6	X	X	ANN	ANN & DCM	SGD
	7	X	X	No	CNN & DCM	Adam
	8	X	X	ANN	CNN & ANN & DCM	Adam

2.3. Feature extraction and transfer Learning

For models 1 to 6 we extract the feature map from the CNN of GoogleNet (Szegedy et al. 2015). GoogleNet is chosen for this study because of its relatively modest size. The network is 22 layers deep and consumes about 6.8m weights. This network is designed to classify images into 1,000 concepts based on the ILSVRC2014 classification challenge. About 1.2m annotated images are used for its training. To obtain the feature map we have removed the final 3 layers of the network (these layers are called 'loss3-classifier', 'prob' and 'output'). The extracted feature map consists of $K = 1{,}024$ features.

To train the models 7 and 8 we use a transfer learning approach. The idea of transfer learning is to use a pre-trained network as the starting point for developing another network for a closely related task. Thus, rather than training the whole network from scratch (typically consisting of 1 to 100 million weights) we start at a fairly good starting point. Thereby, transfer learning lowers the computational burden and the need for large amounts of data.

3. Data

In the absence of a suitable existing data set, i.e. a data set consisting of choices made in the context of visual choice tasks, we create one ourselves by emulating choices. To do so, we use the AVA data set (Murray et al. 2012). This data set consists of approximately 172k images. Each image is rated by on average 222 people on its aesthetic value. To simplify the learning task, we took the top 10% highest rated images and the top 10% lowest rated images. This resulted in a data store consisting of 33k images. 70% of the images were used for the training data set, and the remaining 30% of the images were for the test data.

To construct choice tasks we randomly sampled two images from our data store, and randomly assigned a price tag to each alternative between 1 and 10 euros. A created choice tasks was considered admissible if the two alternatives were sufficiently dissimilar, in the sense that they did not have the same price, and not have the same aesthetic rating. In this way we created a total of 34k choice tasks for our training data set, and another 25k choice tasks for our test data set.

Our synthetic decision-makers are assumed experience a positive utility from the aesthetic value of an image, and a negative utility from its price. Equation 3 shows the utility function:

$$U_{in} = \beta_{rating} \cdot R_{in} + \beta_{price} \cdot C_{in} \quad where \quad \beta_{rating} = 0.6,\ \beta_{price} = -0.3 \qquad \textit{Equation 3}$$

where R_{in} denotes the rating of the image of alternative i and C_{in} denotes its price. Hence, the image rating is taken as its aesthetic value. In line with our conceptual model, decision-makers are assumed to maximise utility (Equation 4).

$$y_{in} = \begin{cases} 1 & if \quad U_{in} > U_{jn}\ \forall\, i \neq j \\ 0 & otherwise \end{cases} \qquad \textit{Equation 4}$$

Note that the choices are deterministically determined, as the utility function (Equation 3) does not involve the usual error term ϵ (Equation 3). It was deemed that the data generating process was sufficiently stochastic by itself due to the fact that the rating R is a stochastic variable.

Figure 3 shows a randomly selected choice observation from the test data set. The aesthetic ratings of the left and right images are respectively, 7.36 and 3.99. Applying Equation 3 yields the highest utility for alternative 1. Hence, alternative 1 is the chosen alternative in this choice observation.

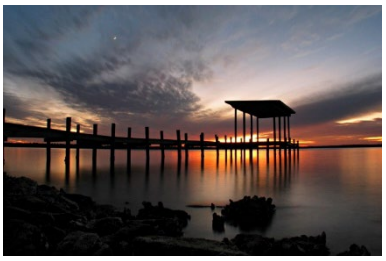
	Alternative 1	*Alternative 2*
		
Price	€8.5,-	€3
Choice	X	

Figure 3: Randomly selected observation from the test data set (images are shown without rotation)

4. Empirical application

4.1. Model estimation and training

Models 1 to 4 are conventional DCMs. Model 1 to 3 estimated using a standard quasi-Newton optimisation algorithm (full information maximum likelihood). Models 4 to 6 are trained using a Stochastic Gradient Descent (SGD) algorithm. For models 5 and 6 an ANN architecture with 2 hidden layers, with two nodes at the first hidden layer and one node at the second hidden layer (which represents

the utility), was found to overall give the 'best' results. Each model is trained 100 times, to minimise the risk of presenting results caused by getting stuck in a local solution.

Finally, models 7 and 8 are trained using the widely used Adam algorithm (Kingma and Ba 2014). A minibatch size of 70 choice tasks is used. All images are downscaled to 224 x 224 pixels, with three RGB colour channels. The ANN of model 8 consists of two hidden layers. The first hidden layer consist of 50 nodes, and the second one of one node (which represents the utility). Furthermore, to avoid the network to learn the latent ratings of the individual images by hard (as opposed to the underlying characteristics of the images that explain their ratings), images are randomly augmented, by randomly shifting the images a number of pixels to the left or right, and by randomly rotating the images. At the start of the training, the weights of the ANN are randomly assignment. The β associated with the price of purchasing the image is given a negative starting value of -0.3. The model training is terminated after 1,000 iterations. Figure 4 shows the loss during training, with on the x-axis the iteration number and on the y-axis the cross-entropy. Every 25th iteration the model performance is evaluated based on 10 minibatches from the test data set (depicted in purple). During the training the networks with the (up until then) best performances are stored (depicted by the red circles) for later evaluation on the full test data set.

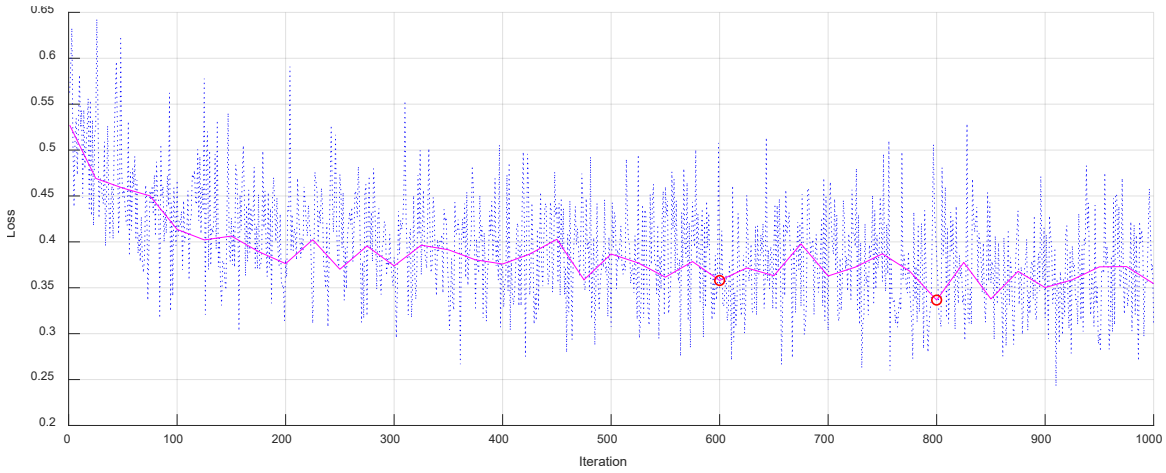


Figure 4: Training model 7

4.2. **Estimation results**

Table 2 reports the estimation and training results. The rho square and its machine learning equivalent, the cross entropy, are computed by evaluating the model performance on the full test data set (consisting of 25k choice tasks). Based on Table 2 a number of observations can be made. Firstly, the feature space holds explanatory power to explain choice behaviour. This can be inferred from the non-zero rho squares of model 2 and model 5. Interesting to mention in this regard is that in model 2 most parameters are found to be statistically significant. Secondly, model 3, 6, 7 and 8 are able to capture the trade-offs between the numeric attributes and the visual stimuli. These models considerably outperform the models that do not jointly consider the numeric attributes and visual stimuli, and thus their trade-offs (i.e. models 2, 3, 4 and 5). Thirdly, counter to our prior expectations, encoding the feature map using an ANN prior to letting the feature map enter the utility function does not seem to improve the performance. Fourthly, for optimisation problems with a large number of parameters SGD is orders of magnitudes faster than the commonly used quasi-Newton based optimisation algorithms for estimating DCMs. This supports the findings by Lederrey et al. (2018). However, we also note that the decrease in estimation time comes at a slight deterioration in the fit. Despite that model 2 is a special case of model 5 –and hence should perform on par or better than model 2– model 5 performs slightly worse

than model 2. Fifthly, joint optimisation of the CNN and the DCM considerably improves the performance as compared to only training the DCM. This shows that the additional efforts required to retrain the CNN pays-off in terms of increased model accuracy.

Table 2: Estimation / training results

MODEL	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7	Model 8
Explanatory variables used								
Cost	X		X	X		X	X	X
Latent features		X	X		X	X	X	X
Feature encoding	N/A	No	No	N/A	ANN	ANN	No	ANN
Model components trained	DCM	DCM	DCM	DCM	ANN & DCM	ANN & DCM	CNN & ANN & DCM	CNN & ANN & DCM
Training algorithm	quasi-newton	quasi-newton	quasi-newton	SGD	SGD	SGD	Adam	Adam
No. observations (training)	34k	34k	34k	34k	34k	34k	70k	70k
No. parameters trained	1	1024	1025	2	4103	4107	6.8m	6.9m
Cross-entropy (out-of-sample)	0.548	0.619	0.434	0.537	0.623	0.441	0.331	0.334
rho^2 (out-of-sample)	0.21	0.11	0.37	0.22	0.10	0.36	0.52	0.52
Estimation time	10 sec[I]	6.2 h[I]	6.2 h[I]	2 sec[II]	11 sec[II]	11 sec[II]	2.5 h[II]	2.5 h[II]

[I] Using 4 CPUs (Xeon @ 3.60 GHz)
[II] Using GPUs (GeForce RTX 2080Ti)

4.3. **Validation**

As we partly[1] controlled the data generating process, we can validate our model beyond looking at the empirical prediction performance. For word limitations, we limit our analyses in the remaining part of this paper to our best performing and most promising model: model 7. To validate this model we conduct two analyses.

First, Figure 5 shows the relation between the true utility difference $V_2 - V_1$ and the predicted choice probability for alternative 2. Each data point (blue) represents a choice task in the test data set. In line with expectations it shows that a positive (negative) utility difference is associated with choice probabilities larger (lower) than $p = 0.5$. To further visualise this relationship, we fitted a logit curve onto these data point (depicted in red). Noticeably, this curve nicely fits the data and crosses the origin (as it should).

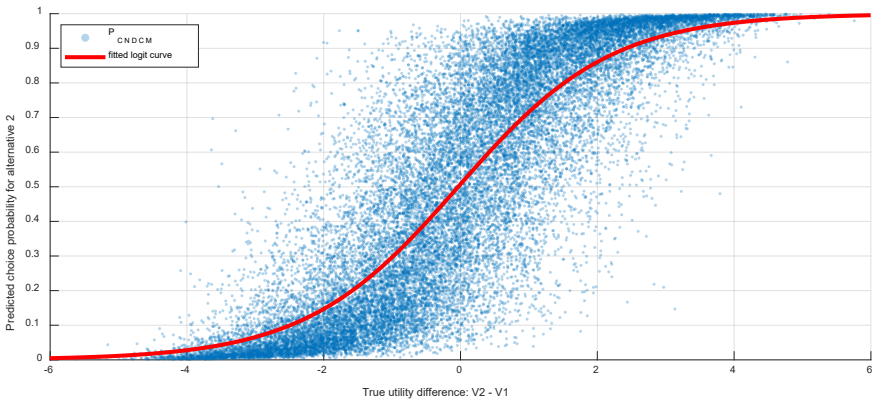


Figure 5: Relation between utility difference and predicted choice probability

Second, Figure 6 shows the predicted choice probability for the aesthetically more attractive and more expensive alternative as a function of the boundary rating of the choice task. As in the previous figure, each data point (blue) represents a choice task in the test data set, and the red line is a fitted logit curve. The taste parameters used to construct the choices: $\beta_{cost} = -0.3$ and $\beta_{rating} = 0.6$, imply a boundary rating

[1] We had no control over the ratings given to the images

of 0.6/0.3 = 2. That is, the higher rated and more expensive is chosen if the difference in rating exceeds two times the difference in costs.[2] Hence, in these data a boundary rating of 2 acts as a threshold value: in case the boundary rating of a choice task is larger than 2 the model should predict that the higher rated and more expensive alternative is chosen, while in case the boundary rating is smaller than 2 the model should predict that the lower rated and cheaper alternative is chosen. In case the boundary rating equals 2, the model should predict that a decision-maker is indifferent between the two alternatives, implying $p = 0.5$. Figure 6 supports these expectations. The fitted red Logit curve crosses the $p = 0.5$ just slightly above the boundary rating of 2. This indicates the model has been able to capture the preferences underlying the data generating process. Another observation is that the point cloud is fairly scattered, signalling a substantial amount of unexplained variance, which presumably is partly caused by noise in the image ratings.

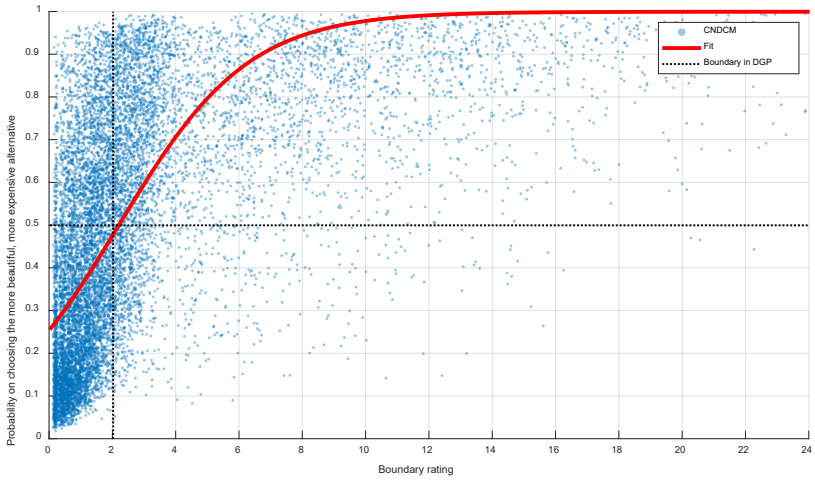


Figure 6: Predicted choice probability as a function of the boundary rating

4.4. **Proof of concept: Willingness-to-Accept inference for a landscape deterioration**

The promise of embedding CV models in DCMs is that we can derive behavioural insights and economic outputs, such as Willingness-to-Pay (WTP) and Willingness-to-Accept (WtA) estimates, regarding visual stimuli. However, unlike fully theory-driven models, inference of the WTP and WTA from the model cannot be done by looking at the estimated / trained model parameters. Rather, we have to use the notion of indifference to obtain WTP and WTA estimates, in a similar way as is proposed in Van Cranenburgh and Kouwenhoven (2019).

To illustrate this point, consider the following situation. A local government considers building a small wind park on a hill top near a village, see Figure 7. The left-hand side plot shows the current situation; the right-hand side plot shows the proposed situation with the wind park.[3] In the current situation the villagers pay a yearly tax of €5 to maintain the current landscape. With the revenue generated by the wind park this yearly tax could go down. The question is how much should the yearly tax decrease in order to have a positive welfare effect? To answer this question, we use model 7 to simulate the effect of the tax level T on the choice probability for alternative 2 (the proposed policy), see Figure 8. Of course, this model is not trained on real data concerning landscape preferences. But, for the purpose of illustration we pretend it is.

[2] in case neither of the alternatives is a dominating alternative.
[3] Note for the sake of illustration here we deliberately choose an aesthetically attractive status quo image, and a less so attractive image for the wind turbine policy.

Alternative 1 (status quo)	Alternative 2 (proposed policy)
	
Yearly tax €5,-	Yearly tax €*T*

Figure 7: Proposed landscape policy

Figure 8 shows that a tax level of €2.7 makes the average villager indifferent. Hence, all else being equal, a tax level lower than €2.7 would result in a net welfare gain for the villagers, and the average WTA equals €5-€2.7 = €2.3 per year.

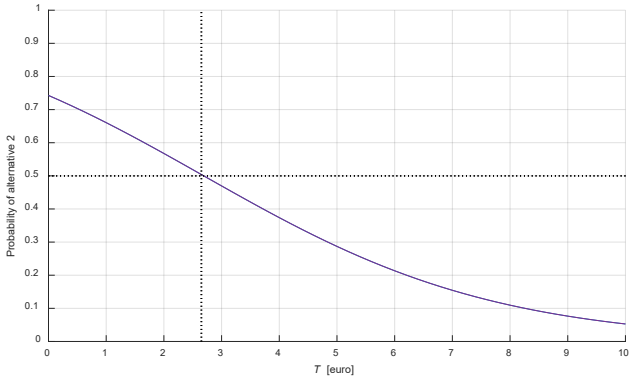


Figure 8: Predicted choice probability as a function of the tax level T

5. Conclusions and next steps

This research has proposed a method to bring visual information into the realm of discrete choice modelling. For this aim, we have blended computer vision models in theory-based discrete choice models. We show that behavioural insights and economic outputs regarding visual stimuli, such as Willingness-to-Accept estimates for landscape deteriorations, can be obtained.

There are plenty avenues for further research. The most obvious one is to collect data involving real trade-offs between numeric attributes and visual stimuli (images). This can be done in a variety of contexts, such as hotel bookings e.g. using data from websites like Booking.com, or landscape choices. In the latter case one could administer a Stated Choice (SC) experiments involving choices across different landscape types. Another important next step is to build further trust in the proposed method, e.g. by investigating the rationale for the model's predictions. This could be done by pioneering techniques such as Layer-wise Relevance Propagation (Bach et al. 2015) in this context.

References

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R. & Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, *10(7)*, e0130140.

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E. & Shah, R. (1994). Signature verification using a" siamese" time delay neural network. Advances in neural information processing systems.

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J. & Chen, T. (2018). Recent advances in convolutional neural networks. *Pattern Recognition*, *77*, 354-377.

Hess, S. & Daly, A. (2014). *Handbook of choice modelling*: Edward Elgar Publishing).

Kingma, D. P. & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, *1(4)*, 541-551.

Lederrey, G., Lurkin, V. & Bierlaire, M. (2018). SNM: Stochastic Newton Method for Optimization of Discrete Choice Models. The 21st IEEE International Conference on Intelligent Transportation Systems.

McFadden, D. L. (1974). Conditional logic analysis of qualitative choice behavior. In P. Zarembka (Eds.), *Frontiers in Econometrics* (pp. 105-142). New York: Academic Press.

Murray, N., Marchesotti, L. & Perronnin, F. (2012). AVA: A large-scale database for aesthetic visual analysis. 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. & Rabinovich, A. (2015). Going deeper with convolutions. Proceedings of the IEEE conference on computer vision and pattern recognition.

Van Cranenburgh, S. & Kouwenhoven, M. (2019). Using Artificial Neural Networks for Recovering the Value-of-Travel-Time Distribution. International Work-Conference on Artificial Neural Networks, Springer.