amadeus

# Deep Choice Model Using Pointer Networks for Airline Itinerary Prediction

KDD2017

August 15th , 2017
Halifax, Nova Scotia, Canada

Alejandro MOTTINI
Rodrigo ACUNA-AGOST

Innovation & Research
Amadeus IT Group
France

# Outline

- Introduction
- Discrete Choice Model
- DCM Using Pointer Networks
- Results
- Conclusions

# Outline

Introduction

Discrete Choice Model

DCM Using Pointer Networks

Results

Conclusions

# Introduction

## Problem formulation

I want to travel to KDD 2017

I look for flights from Nice/France to Halifax/Canada, $13^{th}$ to 17th August 2017

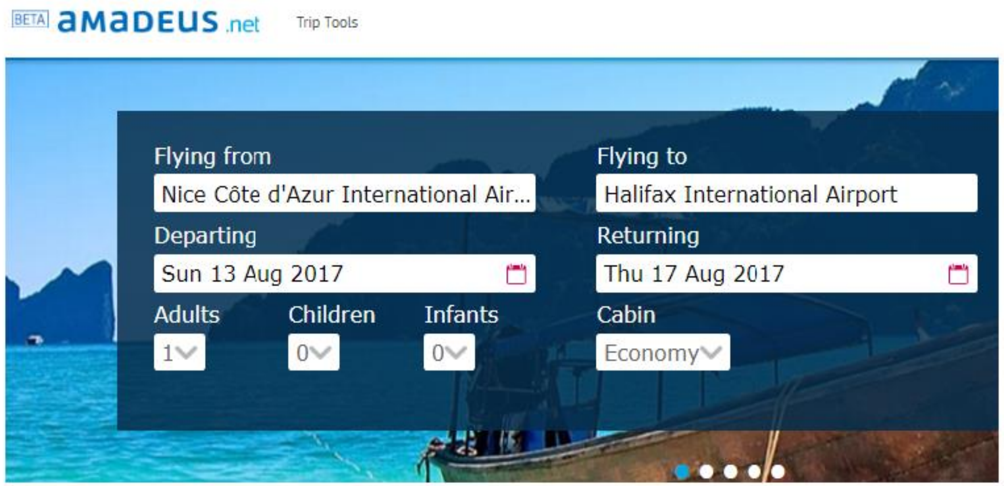


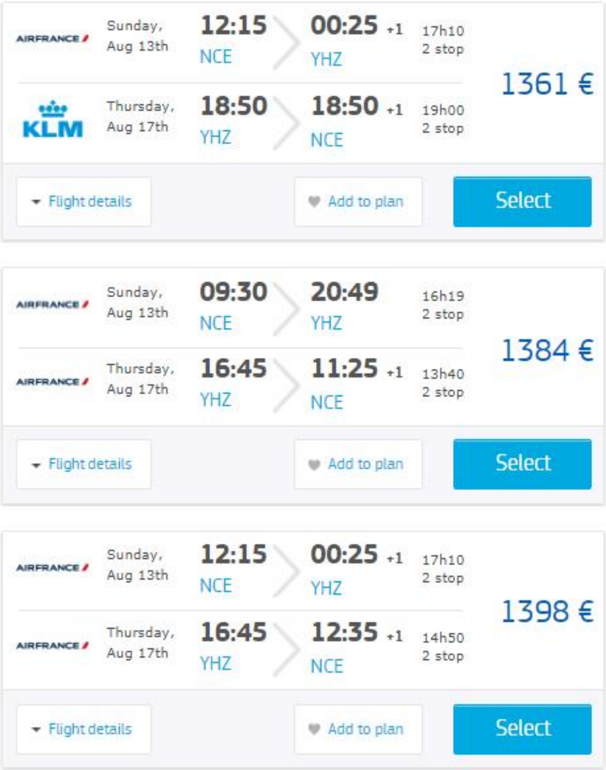
amadeus

# Introduction

## Problem formulation

Given a flight search request, which alternative is most likely going to be selected?




Cheapest?

Least connection time?

Fastest?

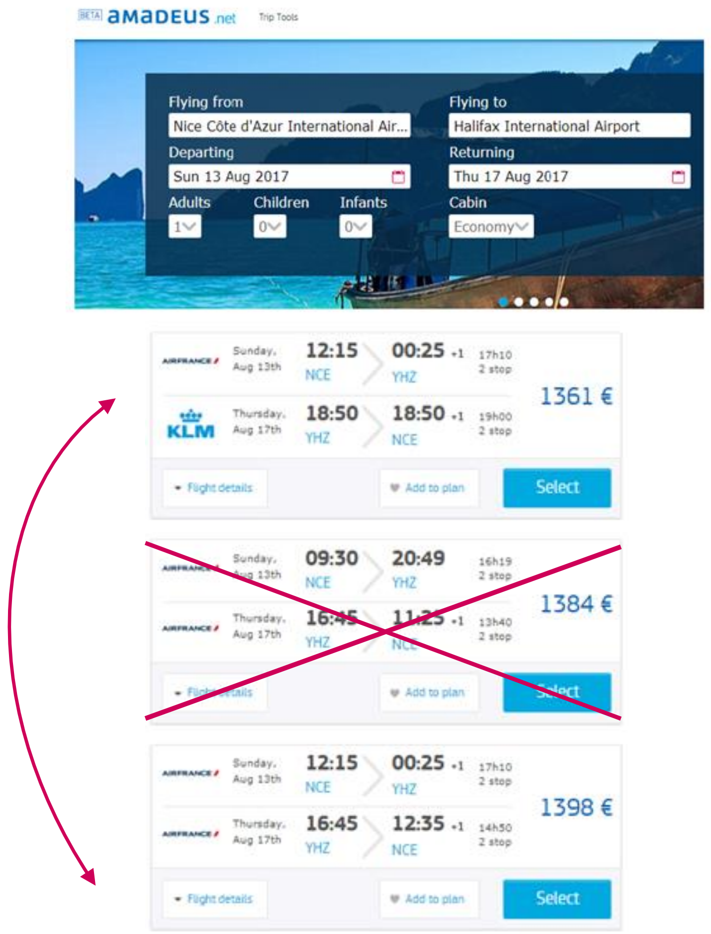
# Introduction

## Problem formulation

_ Predicting the user's choice has many direct applications:
- Filtering/sorting alternatives on website
- Revenue Management
- Price optimization
- …

_ Beneficial for all involved parties:
- Travel providers can increase their revenue and conversion rates
- Passengers can find the most relevant flights for their needs

_ Problem historically tackled with Discrete Choice Modeling (DCM)

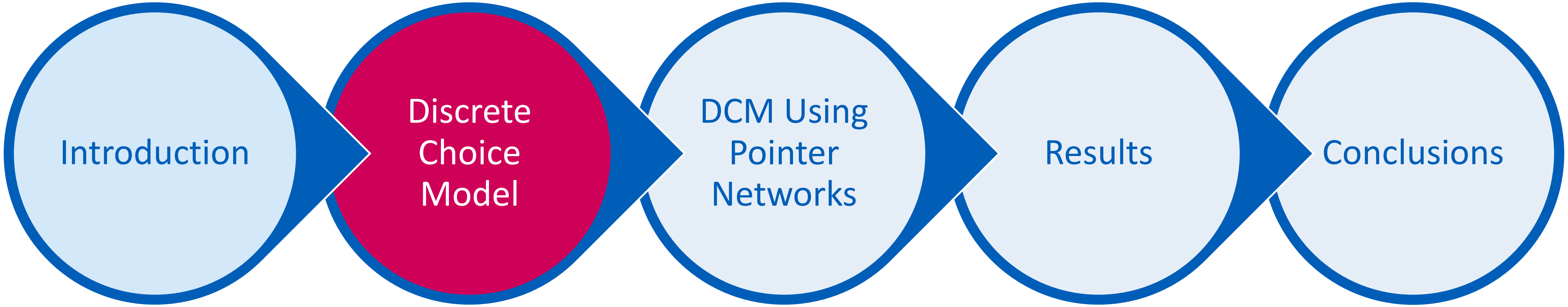






amadeus

# Outline


Introduction
Discrete Choice Model
DCM Using Pointer Networks
Results
Conclusions

# Discrete Choice Model

## Introduction

- Three basic components:
  - Decision maker
  - A choice set (alternatives)
  - The choice

- Faced with finite alternatives $J$, decision maker $i \in I$ choses one based on attributes (of alternatives and himself), and obtains an utility (benefit) $U_{i,j}$ from choosing each one

- Utility is unknown and unobservable, but approximated with a model:

$$U_{i,j} = \underbrace{V_{i,j}}_{\text{Representative Utility}} + \epsilon_{i,j} = \overbrace{\sum_k \beta_k x_{i,j,k}}^{\text{Deterministic component}} + \underbrace{\epsilon_{i,j}}_{\text{Random error}}$$

Ex: $V_{i,j} = a * price_{i,j} + b * tripDuration_{i,j} + \epsilon_{i,j}$

# Discrete Choice Model

## MNL

– Multinomial Logit Model (MNL):

If $\epsilon_{i,j}$ i.i.d and Gumbel distributed and alternatives independent

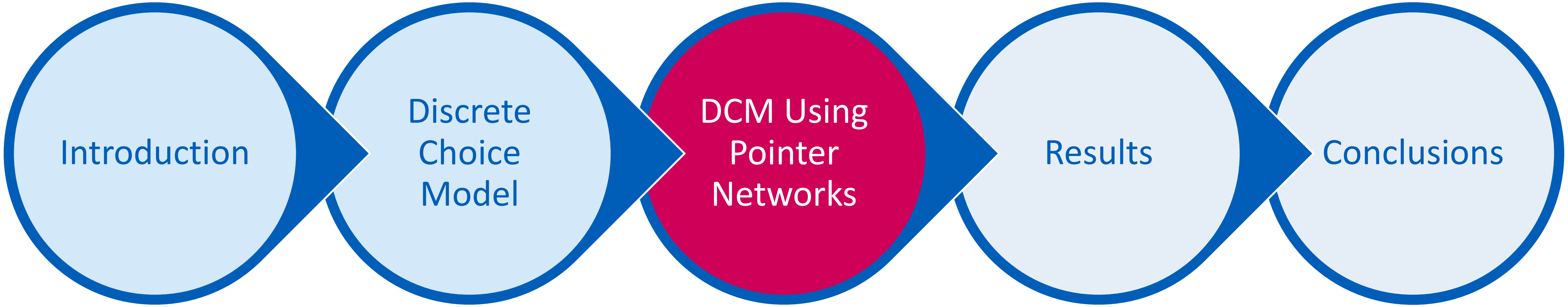
$$P_{i,j} = \frac{\exp(V_{i,j})}{\sum_{k=1}^{J} \exp(V_{i,j})}$$

- Most widely used model in industrial applications
- Fast
- Good performance
- Simple to interpret

- Only linear on inputs
- Assumes independence of alternatives and individuals
- Requires different models for distinct markets
- Requires feature engineering

D. McFadden. 1973. *Conditional logit analysis of qualitative choice behavior*. Frontiers in Econometrics (1973), 105-142

# Outline


Introduction
Discrete Choice Model
DCM Using Pointer Networks
Results
Conclusions

# DCM Using Pointer Networks

## Seq2Seq and Attention Mechanism

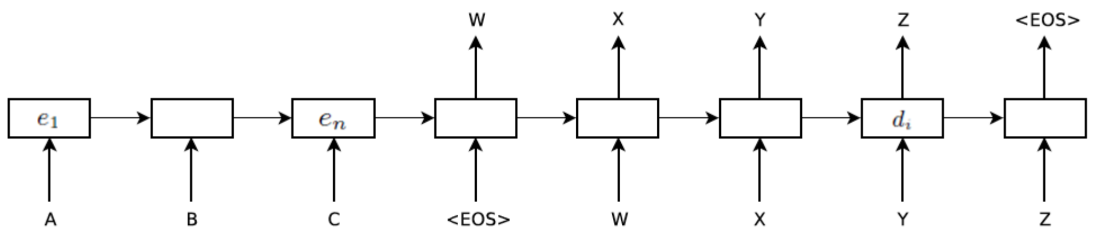
_ Pointer Networks (Ptr-Net)*:

- Combine Seq2seq framework and a modified Attention Mechanism (AttMec)
- Target problems where the outputs are discrete and correspond to positions in the input

_ Seq2seq*:

- Encoder/Decoder (Enc./Dec.) architecture
- Enc. encodes input sequence into fixed-length vector
- Dec. outputs target variable length output sequence
- Enc./Dec. usually  implemented with RNN

$$p(y_i|y_1, \ldots, y_{i-1}, X) = g(y_{i-1}, d_i, c)$$
$$c = q(e_1, \ldots, e_n)$$
$$e_j = f(x_j, e_{j-1})$$




Encoder/Decoder Architecture. <EOS> special end of sequence token.   Taken from Sutskever et al. 2014

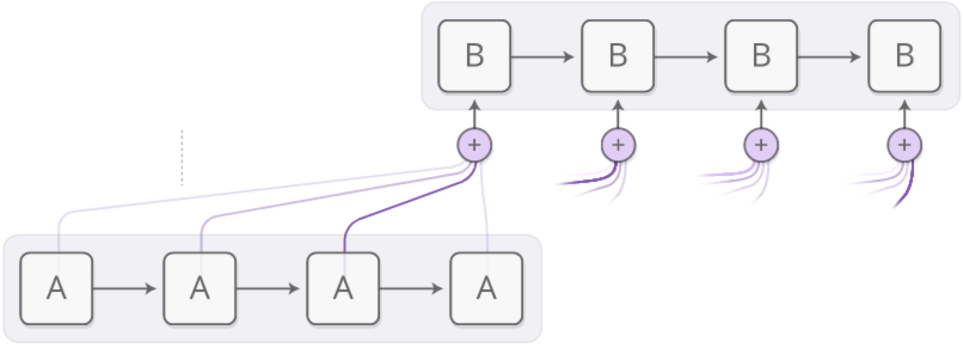
O. Vinyals, M. Fortunato, and N. Jaitly. 2015. *Pointer networks*. In Proc. NIPS
Sutskever, Vinyals, Le. 2014. *Sequence to Sequence Learning with Neural Networks*. In Proc. NIPS

# DCM Using Pointer Networks

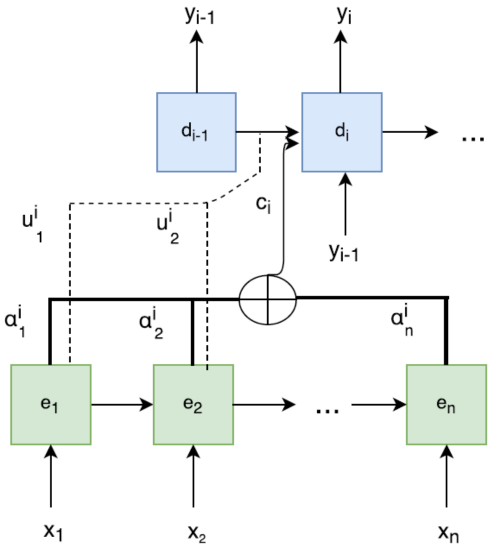
## Seq2Seq and Attention Mechanism

### Attention Mechanism*:

- Connects Enc. And Dec.
- Dec. can access entire sequence of Enc. states (instead of just last one)
- Dec. uses these vectors adaptively while decoding (weighted sum)
- Weights computed with compatibility function between Dec. state and all Enc. states




Dec. B focuses on different information from Enc. A at every step of the decoding process. Taken from [Olah & Carter, 2016]




$$p(y_i|y_1, ..., y_{i-1}, X) = g(y_{i-1}, d_i, c_i)$$
$$d_i = h(d_{i-1}, y_{i-1}, c_i)$$
$$c_i = \sum_{j=1}^{n} \alpha^i_j e_j$$
$$\alpha^i_j = \frac{\exp(u^i_j)}{\sum_{k=1}^{n} \exp(u^i_k)}$$
$$u^i_j = a(d_{i-1}, e_j)$$

D. Bahdanau, K. Cho, and Y. Bengio. 2015. *Neural machine translation by jointly learning to align and translate*. In Proc. ICLR.

# DCM Using Pointer Networks

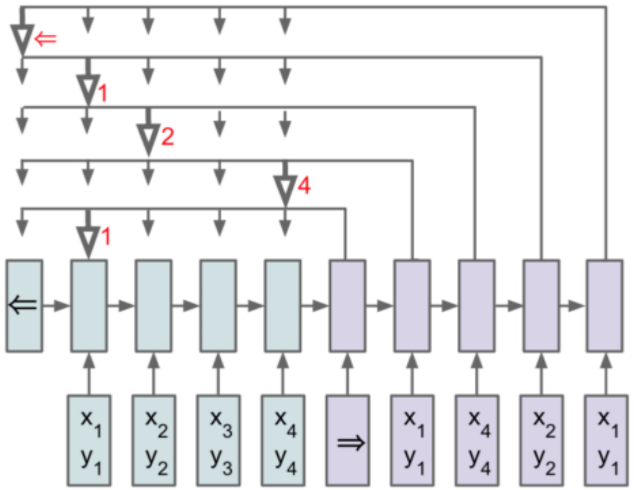
## Pointer Networks

- In Seq2Seq with AttMec, the size of the output dictionary is fixed *a priori*
- Not directly applicable to problems where output dictionary size depends on length of input sequence.
- Ptr-Net adapt attention mechanism to create pointers to elements in the input sequence

- Ptr-Net:
  - Enc. states do not propagate extra information to Dec.
  - Uses $u_j^i$ as pointers to the input sequence elements

$$u_j^i = v^T \tanh(W_1 e_j + W_2 d_i)$$

$$p(y_i | y_1, \ldots, y_{i-1}, X) = \frac{\exp(u_j^i)}{\sum_{k=1}^{n} \exp(u_k^i)}$$




Ptr-Net: Output of the att. mec. is a softmax distribution with dictionary size equal to the length of the input. Taken from Vinyals et al. 2015

# DCM Using Pointer Networks

## Proposed Model

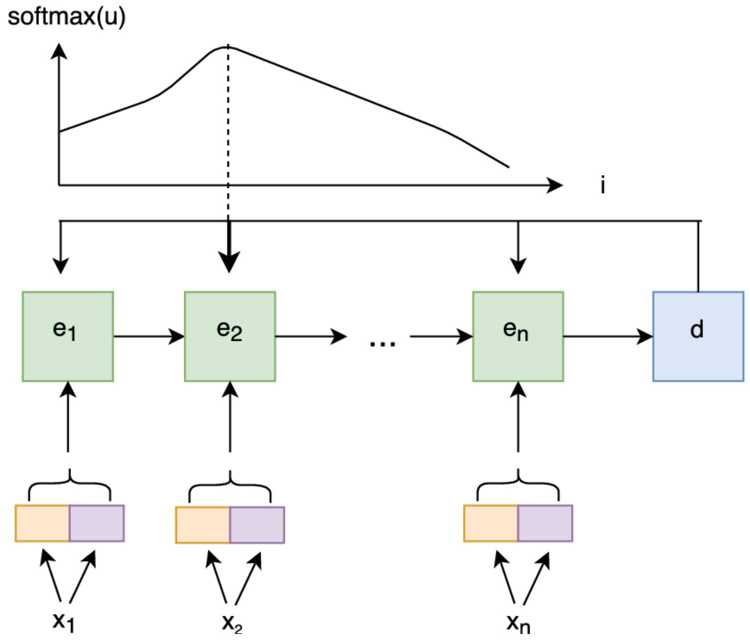
### _ Modified Ptr-Net framework:

- Each user only chooses 1 alternative -> no need to produce output sequence (sort using u vector directly)
- Replace RNN Dec. with FC. Dec.
- Simplify eq. (simpler computation of $d$ / $e_j$ matching)
- Add embeddings and normalization (numerical and categorical features)

$$d = \tanh(W_2 e_n + b)$$

$$u_j = d^T W_1 e_j$$

$$p(y_j|X) = \frac{\exp(u_j)}{\sum_{k=1}^{n} \exp(u_k)}$$






amadeus

# Outline

Introduction → Discrete Choice Model → DCM Using Pointer Networks → Results → Conclusions

# Results

## Validation

_ Validated on real dataset combining airline bookings and search logs

### Airline bookings

- Personal Name Record (PNR) created at reservation time
- Contain the travel itinerary of the traveler (org/dest, date)
- Can include additional data elements (age, gender, etc)

### Search logs

- Contain travel itinerary request (org/dest, date)
- Complete information about the market context
- Which alternatives the customer saw when booking

_ Matchings produced for each booking/search request using information such as booking and search dates

_ Result contains a set of alternatives presented to each user and their corresponding choice

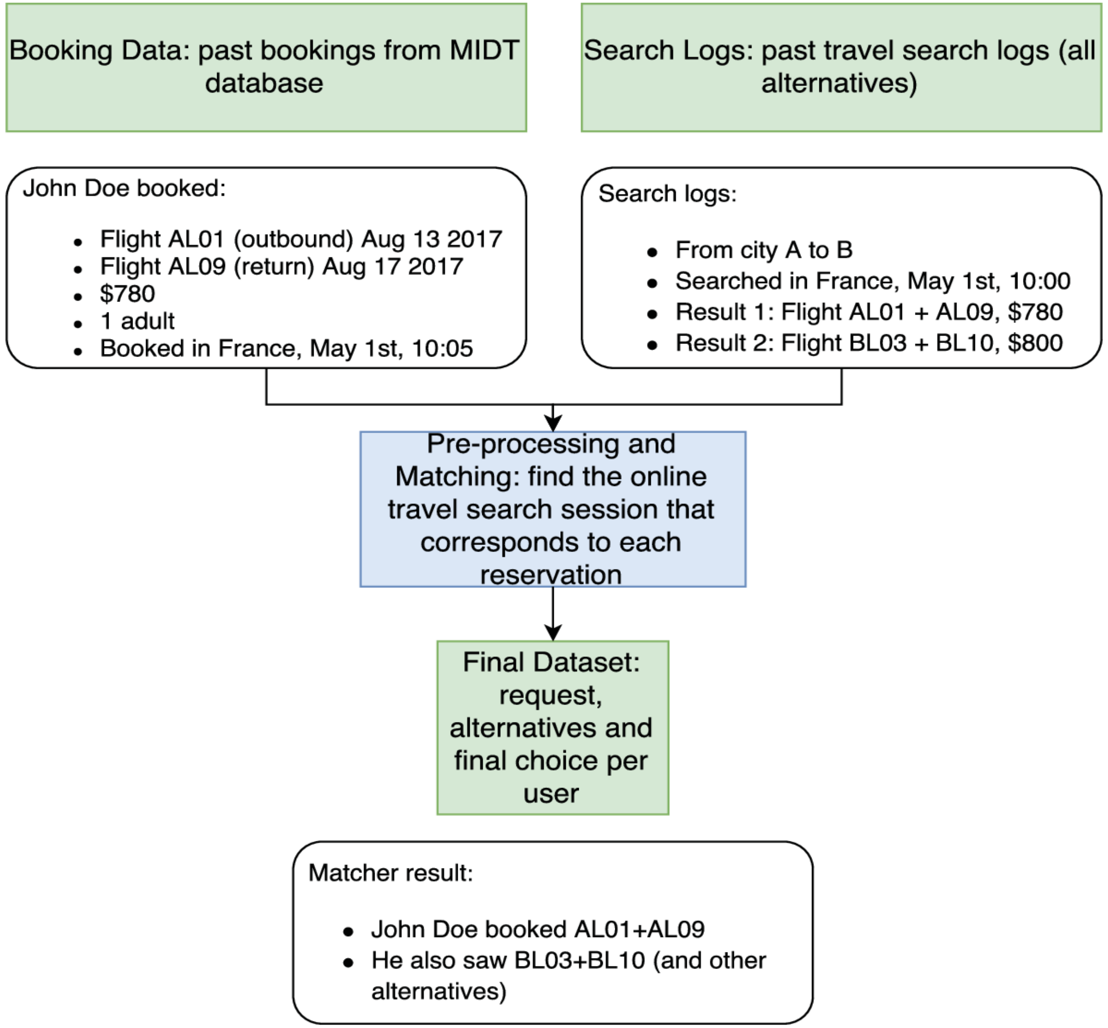
_ Matching is not perfect:
- No direct link between the two data sources
- Booking and search times differ

# Results

## Validation

### Final dataset:

- Numerical and categorical features
- Some shared between alternatives in session (origin, destination, ...)
- Others depend on alternative (price, trip duration, ...)
- Alternatives sorted by ascending price in each user session
- 34K user sessions
- Max. 50 alternatives per session
- 35 medium-haul European O&D






amadeus

# Results

## Validation

_ Evaluation:

- Top-N accuracy (rank alternatives by decreasing probability, consider a prediction Top-N accurate if ranking of choice <= N )
- Airline market share (count number of real and predicted choices associated to each airline, normalize by number of sessions)
- % of sessions where real choice in top 15 alternatives, but predicted choice after top 15

_ Compare performance against MNL, Machine Learning (ML) based method and heuristics

_ ML:

- Train classifier (GBT) on each alternative independently (soft classification) (sessions are shuffled)
- Sessions are regrouped and the probability estimates normalized with softmax
- Classifier learns if an alternative was chosen or not by some user.

| | |
|---|---|
| Alternative 1 | 0.01 |
| Alternative 2 | 0.03 |
| | ... |
| | ... |
| | $\Sigma = 1$ |

_ Heuristics:

- Predicted choice = cheapest itinerary
- Predicted choice = shortest itinerary

# Results

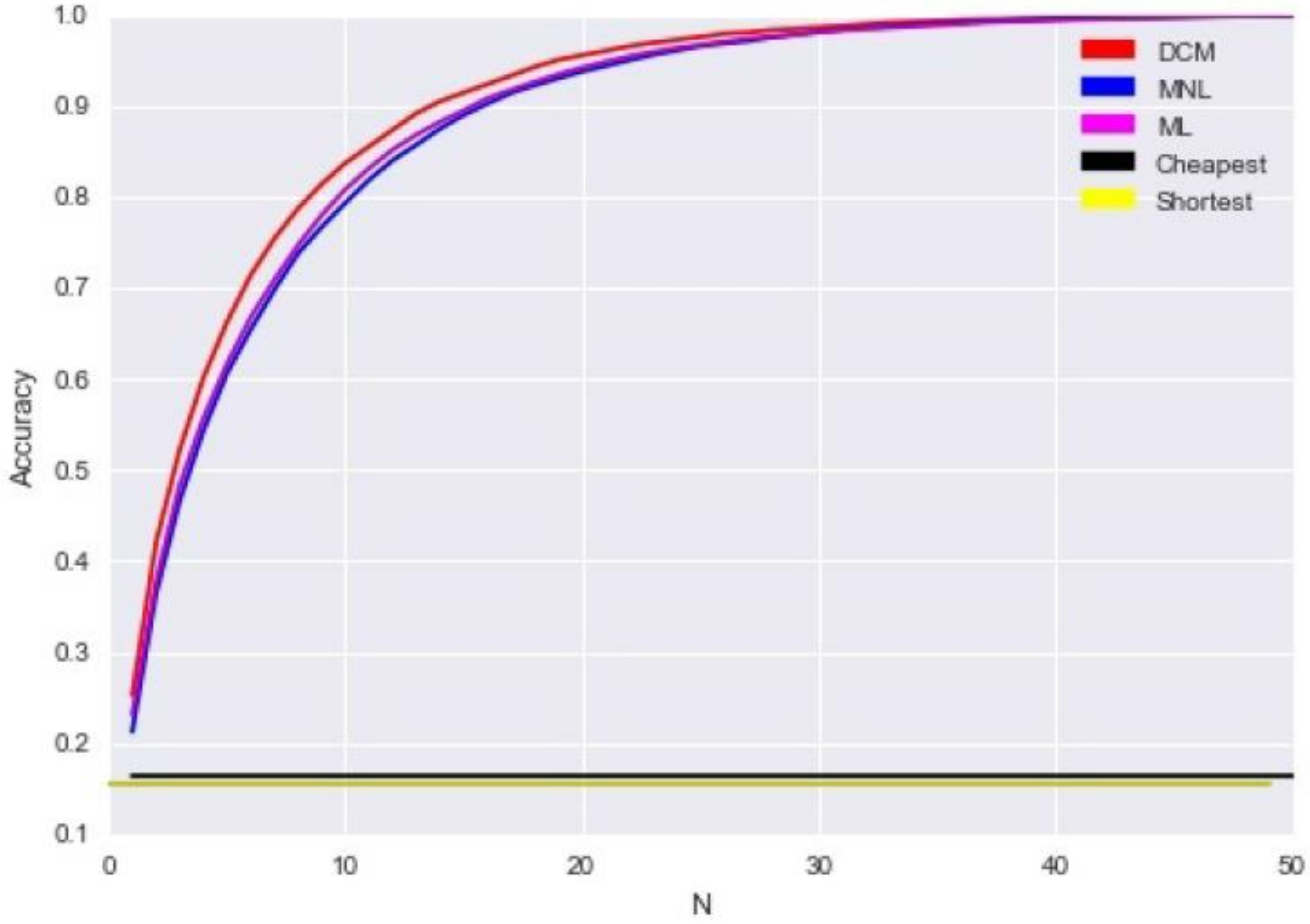



Fig: Top-N accuracy for the compared methods

# Results

| Method | Top-1 Acc. | Top-5 Acc. |
|---|---|---|
| DCM | 25.3 | 66.3 |
| ML | 23.1 | 61.7 |
| MNL | 21.2 | 60.6 |
| Cheapest | 16.4 | 16.4 |
| Shortest | 15.4 | 15.4 |

| Method | % |
|---|---|
| DCM | 6.9 |
| MNL | 7.1 |
| ML | 13.6 |

Percentage of sessions that have the real choice within the top 15 alternatives but predicted choice after the top 15

- Applications such as dynamic pricing are sensitive to small increase in Top-1/5 Acc. :
  - If an airline knows their itinerary is the most likely choice of a user, they can increase the price slightly
  - Even a 1% increase per user can lead to a significant overall increase in profit
- Top-15 Acc. important for ranking/sorting:
  - Most websites show approximately 15 results per page
  - Users usually only look at first page
- % of sessions with real choice in top 15 alternatives, but predicted choice after top 15:
  - DCM outperforms other methods
  - Business importance: not placing the optimal alternative in the first page of the search results could lead to a lower conversion rates

# Results

## Results

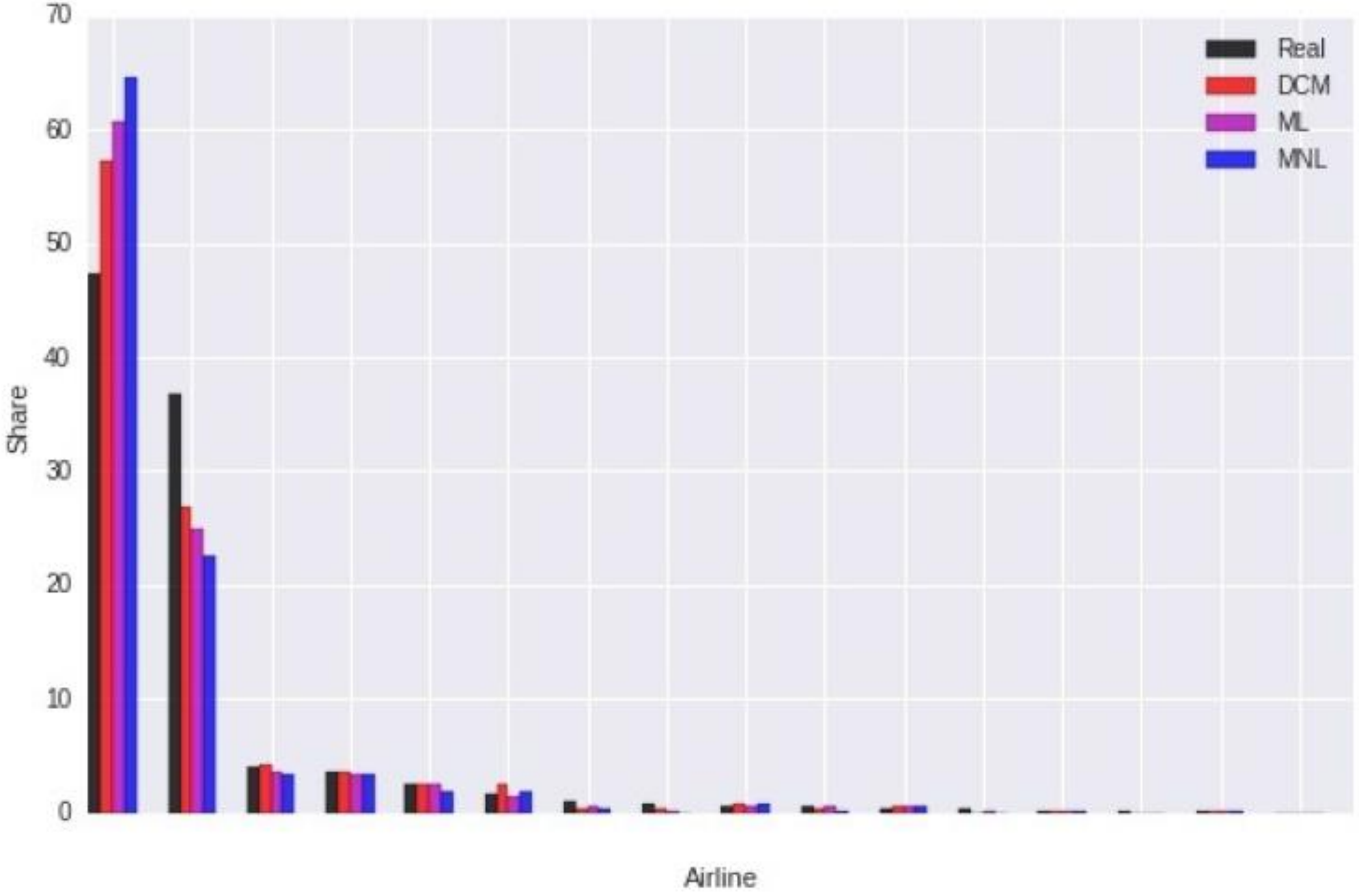
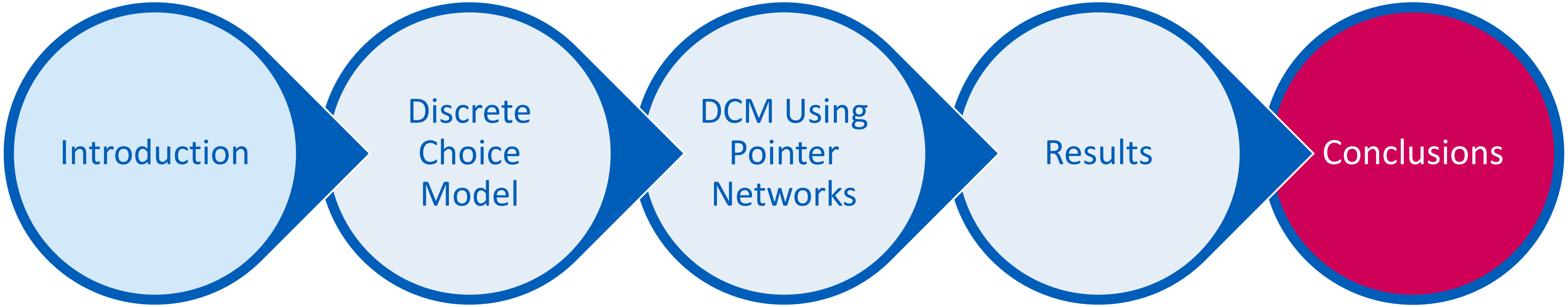



Fig: Real and predicted airline share for the compared methods. The airlines have been anonymized.

- Ours approximates real market share per airline better
- Good market share estimation important for diff. airline applications (schedule planning, prediction of potential impact of new flight/route)

# Outline


Introduction
Discrete Choice Model
DCM Using Pointer Networks
Results
Conclusions

# Conclusions

## Problem

- Travel providers are interested in understanding how passengers choose among alternative itineraries
- Used for different commercial applications
- Problem historically tackled with Discrete Choice Model (MNL)

## What we propose

- New choice model based on Pointer Networks
- Combines Seq2Seq with the Attention Mechanism
- Learns to point to the alternative most likely to be chosen
- Evaluated on real dataset combining on-line user search logs and airline bookings

## Results

- Compared against MNL and ML based method
- Outperform both models in terms of prediction accuracy and other business metrics.
- Non-linear with respect to the inputs, no statistical independence assumptions of the alternatives
- No previous data segmentation is required, no feature engineering



amadeus

Strictly CONFIDENTIAL

# Thank you!