Machine Learning for Predicting Flight Ticket Prices

Supervisor: Boi Faltings, Igor Kulev

Student: Jun Lu

fppt.com

Overall work flow

unknown target function
$f: x \rightarrow y$

(ideal ticket price formula)

test samples
for ***specific routes***
$\mathcal{D}$:($\mathbf{x}_1$), … , ($\mathbf{x}_M$)

traning samples
for ***specific routes***
$\mathcal{D}$:($\mathbf{x}_1$,y_1), … , ($\mathbf{x}_N$,y_N)

learning
algorithm
$\mathcal{A}$

final hypothesis
$g \approx f$

$y_i = g(\mathbf{x}_i)$
predict to buy or
wait for ***specific routes***

(**historical records** in an airplane company)

('learned' formula to be used)

extract features for
specific routes

Identify and define inputs
and outputs

('learned' formula to be used)

extract features for
generalized routes

test samples
for ***generalized routes***
$\mathcal{D}$:($\mathbf{x}_1$), … , ($\mathbf{x}_K$)

$y_i = g(\mathbf{x}_i)$
predict to buy or wait for
generalized routes

Samples which have same departure dates

ippt.com

Data Collection

- From a major airplane search web site between Nov. 9, 2015 and Feb. 20, 2016 (103 observation days).
- We restricted the collecting data on non-stop, single-trip flights for 8 routes.
- Overall, we collected 36,575 observations for these 8 different airlines.
- We did not find any tickets that were sold out in the specific queried days.
- And also collected 12 new routes for generalized problem.

fppt.com

Data Description and Interpretation

- We split the datas between Nov. 9, 2015 and Jan. 15, 2016 as the training dataset; and the datas betweeen Jan. 16, 2016 and Feb. 20, 2016 as the testing dataset.
- Generalized problem dataset has the same period as the testing dataset of specific problem.
- The training dataset consists of N_tr=16,208 data samples
- The testing dataset consists of N_te=20,367
- The testing dataset for generalized problem: N_ge=14,160

fppt.com

Feature Extraction

- the *flight number*(encoded by dummy variables);
- the *minimum price so far;*
- the *maximum price so far;*
- the *query-to-departure* (number of days between the first query date(09.11.2015 in our case) and departure date)
- the *days-to-departure* (number of days between the query date and departure date)
- the *current price*
- *Many other features can be extracted, (e.g. indicator whether it is holiday or not)*

Specific problem - Regression

- Methodology
- Least Squares
- Neural Networks
- Decision Tree
- Adaboost-Decision Tree
- Random Forest
- K Nearest Neighbors
- Uniform Blending
- Regression Result

Regression - Methodology

- For the output, I set it to be the minimum price for each departure date and each flight.
- Our regression method is to predict the expected minimum price for a given departure date and a given flight with the input features.
- As a result, if the current price is less than the expected minimum price, we predict to buy; otherwise, we predict to wait.

Regression Result

Performance(%)	Model								
Routes \ Method	Optimal	Random Purch.	Linear Regression	NN	Decison Tree	KNN	AdaBoost	Random Forest	Uniform Blending
BCN→BUD	100.0	0.00	-42.17	67.03	42.31	38.19	48.49	54.67	44.37
BUD→BCN	100.0	0.00	1.91	57.12	72.66	91.96	71.59	96.78	80.17
CRL→OTP	100.0	0.00	10.34	18.60	30.40	37.48	41.01	56.35	25.68
MLH→SKP	100.0	0.00	-21.02	-10.74	22.37	36.07	31.50	77.17	29.22
MMX→SKP	100.0	0.00	-118.53	53.25	16.46	67.91	53.80	-0.69	65.70
OTP→CRL	100.0	0.00	-19.47	57.30	63.85	69.09	64.11	67.51	63.85
SKP→MLH	100.0	0.00	47.72	35.80	50.11	52.10	51.30	45.34	45.34
SKP→MMX	100.0	0.00	-96.14	64.63	43.33	30.07	43.33	46.34	49.76
Mean Performance	100.0	0.00	-29.67	42.87	42.68	52.86	50.64	**55.43**	50.51
Variance	0.00	0.00	2654.78	636.21	**37.04**	409.11	**143.49**	707.99	302.90

TABLE II

Regression Normalized Performance Comparison of 8 routes.

- Random Forest Regression gets the best performance;
- However, it's variance is not small enough;
- In this case, although for some routes, it gets good performance, for other routes, it gets bad performance. From the aspect of the clients, it is not fair for the some clients to buy tickets for which the system may predict badly.
- The preferred method in regression is AdaBoost-Decision Tree Regression method, which has smallest variance and a relative high performance.

Specific problem - Classification

- Methodology
- Solving Imbalanced Data Set
- Identification of Outliers
- Logistic Regression
- Neural Networks
- Decision Tree
- Adaboost-Decision Tree
- Random Forest
- K Nearest Neighbors
- Uniform Blending
- Classification Result

Classification - Methodology

- For the output, I set the data of which the price is the the minimum price from the query date to the departure date to be 1(namely Class 1 - to buy), otherwise, I set it to be 0(namely Class 2 - to wait).
- The classification method is to predict to buy or to wait with the input features.
- As a result, if the prediction is to buy, we buy the ticket, and we only buy the earliest one.

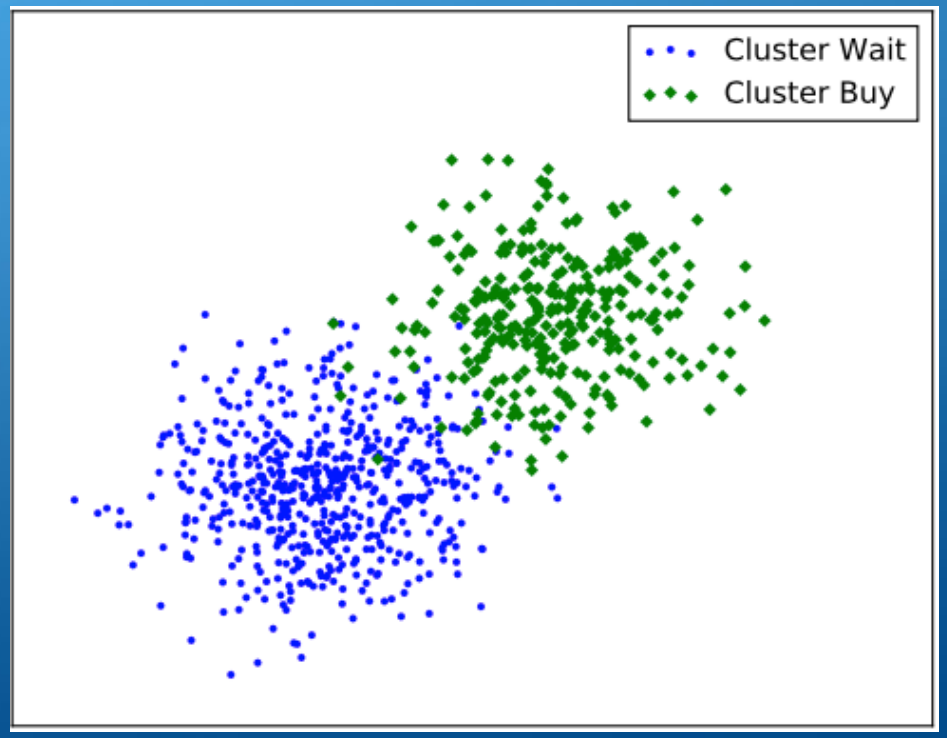
Solving Imbalanced Data Set

- Random Under Sampling
- Random Over Sampling

- Firstly, the *Random Under Sampling* would not be useful in our problem, because in our problem, the data set is very unbalanced, i.e. the buy entries is very sparse. If we use *Random Under Sampling*, we will lose many information.
- As a result, we preferred the second method, which is *Random Over Sampling*

Identification of Outliers

- K-Means
- Mixture of Gaussians



Classification Result

Performance(%)	Model								
Routes \ Method	Optimal	Random Purch.	Logistic Reg	NN	Decison Tree	KNN	AdaBoost	Random Forest	Uniform Blending
BCN→BUD	100.0	0.00	36.13	36.13	54.67	17.58	75.27	29.95	48.49
BUD→BCN	100.0	0.00	39.97	39.98	11.02	64.62	60.34	86.60	30.32
CRL→OTP	100.0	0.00	18.60	18.60	70.51	83.48	71.69	52.81	45.73
MLH→SKP	100.0	0.00	-22.16	-22.15	57.76	38.35	32.64	0.67	46.34
MMX→SKP	100.0	0.00	53.25	53.26	-26.42	34.44	50.76	-34.16	65.15
OTP→CRL	100.0	0.00	57.30	57.30	80.35	87.69	85.33	19.83	71.71
SKP→MLH	100.0	0.00	31.03	31.03	-28.2	55.28	65.02	33.61	49.12
SKP→MMX	100.0	0.00	64.63	64.63	23.63	14.99	49.76	9.57	57.8
Mean Performance	100.0	0.00	34.84	34.84	30.42	49.55	**61.35**	24.86	51.83
Variance	100.0	0.00	660.74	660.74	1592.45	679.48	151.25	1148.16	**144.56**

TABLE V

Classification Normalized Performance Comparison of 8 routes.

- As we see, AdaBoost-DecisionTree, KNN, and Uniform Blending get positive performance for all the 8 routes and have smaller variance over these routes compared to other classification algorithms.
- The AdaBoost-DecisionTree method gets the best performance and a relative low variance over 8 routes.
- And as expected, the uniform blending method has the lowest variance just like the theory of uniform blending describes.

Specific problem - Q Learning

- Methodology
- Q-Learning Result

Q Learning - Methodology

- Standard Q-learning formula is:

  $Q(a', s') = R(s', a') + \gamma \max_{a'}(Q(a', s'))$

- In our case:

  $Q(b, s) = -\text{price}(s)$ [no future award]

  $Q(w, s) = \max(Q(b, s'), Q(w, s'))$ [no instant awd]

- Averaging step: Our equivalence class is the set of states with the same flight number and the same days before takeoff:

  $Q(a', s') = \text{Avg}_{s*\sim s}(R(s*, a') + \gamma \max_{a'}(Q(a', s')))$

Q Learning - Result

Performance(%)	Model		
Routes \ Method	Optimal	Random Purch.	Q Learning
BCN→BUD	100.0	0.00	68.76
BUD→BCN	100.0	0.00	61.81
CRL→OTP	100.0	0.00	51.13
MLH→SKP	100.0	0.00	54.01
MMX→SKP	100.0	0.00	61.27
OTP→CRL	100.0	0.00	72.08
SKP→MLH	100.0	0.00	65.29
SKP→MMX	100.0	0.00	6.50
Mean Performance	100.0	0.00	55.11
Variance	0.00	0.00	380.06

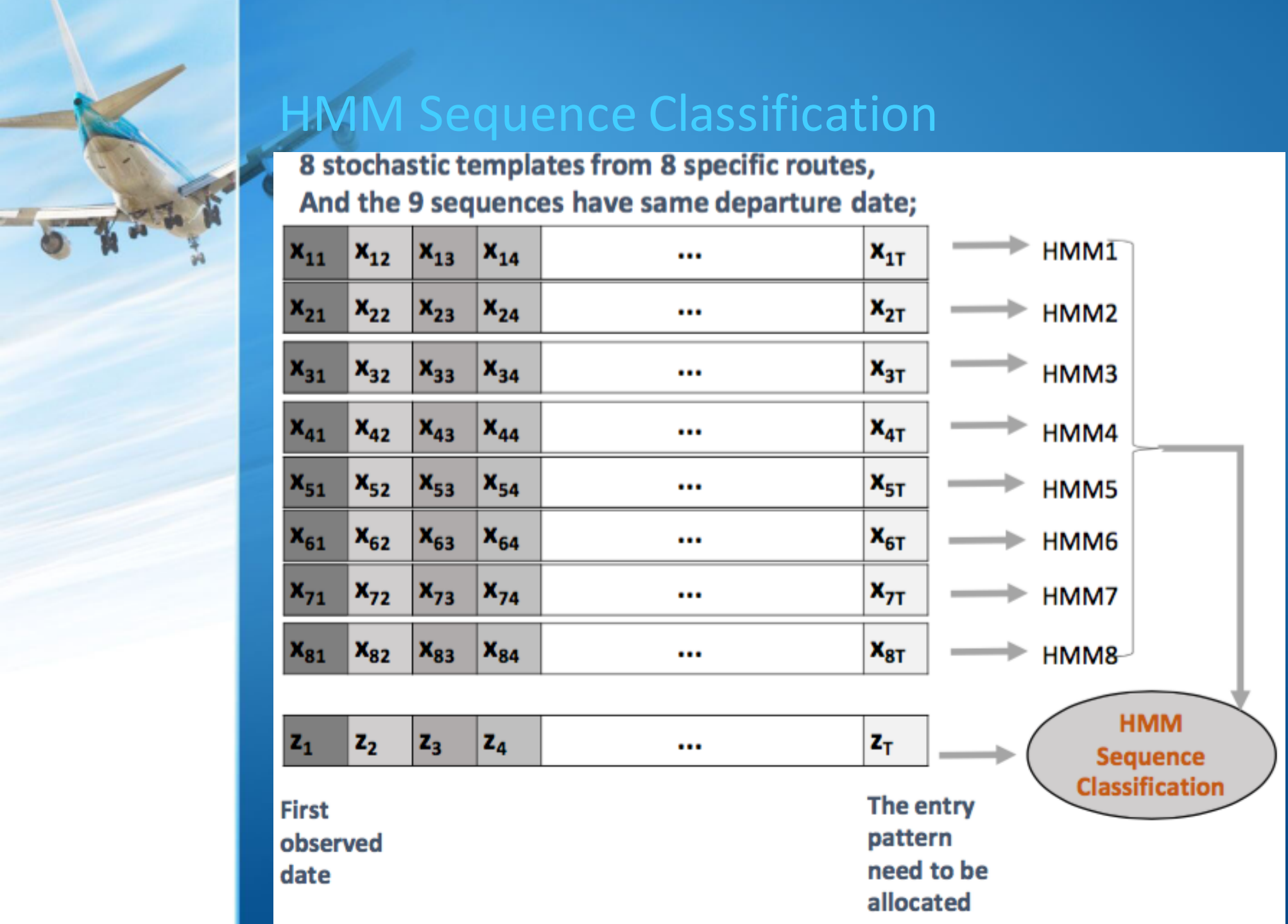
TABLE VI
Q Learning Performance.

- As we see, the Q-Learning method described in [Etzioni et al., 2003] has an acceptable performance and the variance is not large as well.
- The performance of it is a little worse than AdaBoost-DecisionTree Classification and Uniform blending Classification algorithms.

Generalized problem

- HMM Sequence Classification
- Uniform Blending
- Generalized Problem Result

HMM Sequence Classification



Uniform Blending

- The uniform blending algorithm is just to allocate the 8 models to the new route respectively
- Then average over the 8 models.

Generalized Problem Result

Performance(%)	Model			
Routes \ Method	Optimal	Random Purch.	Uniform	HMM
BGY→OTP	100.0	0.00	80.52	87.09
BUD→VKO	100.0	0.00	-75.1	-48.6
CRL→OTP	100.0	0.00	51.06	63.11
CRL→WAW	100.0	0.00	-16.57	17.32
LTN→OTP	100.0	0.00	9.47	53.22
LTN→PRG	100.0	0.00	21.14	45.83
OTP→BGY	100.0	0.00	24.95	62.33
OTP→CRL	100.0	0.00	-56.99	0.54
OTP→LTN	100.0	0.00	-105.07	-84.45
PRG→LTN	100.0	0.00	-0.18	38.16
VKO→BUD	100.0	0.00	-48.61	-15.97
WAW→CRL	100.0	0.00	-3.3	35.12
Mean Performance	100.0	0.00	-14.84	31.71
Variance	0.00	0.00	2637.84	2313.25

TABLE VIII
Generalized Model Performance.

- The uniform blending does not get any improvement.
- HMM Sequence Classification algorithm makes 9 routes get improvement, 3 routes have negative performance.
- Although the average performance is 31.71%, which is lower than that of the specific problem, it makes sense because we did not use any historical data of these routes to predict

Conclusion

- For the specific problem and from the aspect of performance, AdaBoost-Decision Tree Classification is suggested to be the best model, which has 61.35% of optimal performance and has relatively small performance variance over the 8 different routes.
- From the aspect of performance variance for different routes, Uniform Blending Classification is chosen as the best model with relatively high performance.
- On the other hand, the Q-Learning method got a relatively high performance as well.
- Compare the validation curve of classification methods to that of regression methods, we could find that the cross validation error or precision of all the 5 folds in classification has far smaller variance than that of regression (shown in the report).
- For the generalized problem(i.e. predict without the historical), we did not test many models. However, the HMM Sequence Classification based AdaBoost-Decision Tree Classification model got a good performance over 12 new routes, which has 31.71% of optimal performance.

Thank You !