Predicting Airfare Prices

Manolis Papadakis

Introduction

Airlines implement dynamic pricing for their tickets, and base their pricing decisions on demand estimation models. The reason for such a complicated system is that each flight only has a set number of seats to sell, so airlines have to regulate demand. In the case where demand is expected to exceed capacity, the airline may increase prices, to decrease the rate at which seats fill. On the other hand, a seat that goes unsold represents a loss of revenue, and selling that seat for any price above the service cost for a single passenger would have been a more preferable scenario.

The purpose of this project was to study how airline ticket prices change over time, extract the factors that influence these fluctuations, and describe how they're correlated (essentially guess the models that air carriers use to price their tickets). Then, using that information, build a system that can help consumers make purchasing decisions by predicting how air ticket prices will evolve in the future. We focused our efforts on coach-class fares.

Related Work

There has been some previous work on building prediction models for airfare prices using Machine Learning techniques [1] [2] [3]. The various research groups have focused on mostly different sets of features and trained their models on different kinds of flights. A major distinction among these projects is the specific trend they are trying to predict. Specifically, we can categorize projects into 2 approaches: studying the factors that influence the average price of a flight [2], or those that influence the price of a specific flight in the days leading up to departure [1] [3]. We will use this distinction in the definition of our model.

There also exist commercial services, like Bing Travel (which actually evolved from the work in [1]), that perform this kind of prediction, but their models are not made public.

Data collection

There are various sources of airfare data on the Web, which we could use to train our system. A multitude of consumer travel sites supply fare information for multiple routes, times, and airlines. Most airlines also expose a booking system on their websites. These sites could be scraped to extract data for the subset of flights we are interested in. There are also applications, like KVSTool, that offer more direct access to the information published on the reservation systems, in a format that is more convenient to parse. Some research groups have made their datasets available to other researchers, although the set of features they have recorded may be different from the ones we are interested in.

One feature that is especially important for our purposes is historical ticket prices, both for the specific flight we are reasoning about, and as an average for similar flights in the past. The former is sometimes available from consumer travel websites and air travel analytics companies, but can also be collected manually over a period of time. The latter can be extracted from statistics released by certain air travel governing bodies (in addition to other aggregate statistics that may be useful).

There are certain features that would be very useful in training a prediction algorithm, but which the airlines, due to their competitive environment, do not release to the public. This includes the actual number of available seats on a flight, the distribution of ticket purchases over the lifetime of a fare and fine-grained sales figures. Also, the publicly available fare data does not include certain kinds of tickets, like consolidator and corporate tickets, which are negotiated in private with the airline. Finally,

many low-cost airlines do not publish their rates on the reservation systems, and are thus not indexed by most online systems.

We decided to use data collected from Bing Travel, which conveniently included historical information on lowest daily fares for the past 42 days. This freed us from having to collect such data manually, but did somewhat limit our design choices, if we wanted to remain compatible with the historical information available from the site. For example, we had to restrict ourselves to round-trip flights between US airports, and couldn't pick specific itineraries (e.g. only non-stop flights) or specific airlines. Also, the only departure date for which we could get a full fare history (i.e., up to the last day before departure; we assume flyers would buy their tickets at least one day in advance) was the day after our collection day, since online sites generally do not keep data on past flights.

Under these constraints, we picked 20 flights to consider: all round-trip flights between 5 major US hub airports (Atlanta, Chicago, Los Angeles, Dallas and Denver) leaving on 12/14/2012 and returning one week later. Due to our choice of features, we could only use information about those dates for which we had at least 7 days of prior fare history, i.e. 36 out of the 43 days (42 previous days plus the present day) that we had data for. This adds up to 720 data points.

The data collection phase was slightly involved, because Bing Travel does not present the historical information in text, but only in the form of a fare chart. To extract fare data about a flight we had to crop a screenshot of the site (the fare chart was not directly accessible as an image) to only the interior of the graph, and feed that to a graph digitizer (Engauge), which we had previously configured appropriately so that it could automatically detect the chart line and the axes (which we had to draw manually on the cropped image). With the added information about the minimum and maximum values on the axes, Engauge would export a CSV file with the day-to-price values represented by the chart. We then had to filter those values to only include points corresponding to integer-valued dates, and subsequently process those to extract the desired features.

Model

We can view the price of a flight as a random variable C. Based on our assumption that prices of different flights exhibit similar patterns of fluctuation over time, we can model C as the sum of two random variables, B and F. B models the “base” price of the ticket (which is constant for the duration of the ticket sale), and F its fluctuation. The parameters that determine the value of C affect B and F in different ways. Some only affect one of the two (e.g., the date of booking only affects F, by definition). Most of them affect both, but to different degrees (e.g., we expect that the date of travel and the historical average will mostly affect the base price).

Based on the above formulation, we can more formally reason about our objective. Our aim is to predict the fluctuation of ticket prices, so we are mostly interested in F, and, by extension, in those features that affect F the most (which we will have to discover).

Although we only care about the distribution of F, if we want to combine data for more than one flights, we will need to compensate for their difference in the value of B. We will, therefore, need to further simplify our model. One possibility, which aims to normalize fluctuation across different flights, is to assume that fluctuation is always relative to the base price, i.e. to make fluctuation into a multiplier ($C = BF$). Going a step further, we may assume that the features which affect B are completely disjoint from those that affect F, thus separating the problem of predicting F from that of predicting B.

Since we don't want to reason about B, our initial approach was to substitute its (unknown) value for each flight with an approximation. Possible candidates included the price of the same flight a year ago, the historical average for that route, or the average price over the whole period from the moment

ticketing opens up to 2 months before the date of departure (prices exhibit little fluctuation with demand over that time, which must mean that the offered fare is a reasonable one for the flight). Of these values, only the first two were readily available, and the aggregate data we could find was often unusable, because it averaged over all fare classes, while we only cared about coach-class tickets. Moreover, from comparing the past averages with the data collected from Bing Travel, we found that they had little correlation with current prices. As a result, we opted not to follow this approach, and instead chose to record previous days' prices in relation to the price at each specific day, to compensate for differences in the base price across different flights.

Feature selection

There are multiple features that can be used to train a predictor, and for some it makes sense to collect them over a period of time.

We expect certain features to be mostly relevant for the base price of a flight, e.g.:

- characteristics of the carrier (size, business model – i.e. low-cost or traditional)
- date/time of flight, and of return leg, if applicable (date – especially if it's close to a major holiday, time of day, day of week – e.g. the majority of business flight itineraries is Monday to Friday)
- duration of stay (for round-trip flights)
- size/model of aircraft (a proxy for flying efficiency)
- whether the destination is international or domestic
- flight distance (proxy for operating cost, and potentially for buyers' types)
- competition on the same airport (market share of airline in terms of passengers – especially whether it is the dominant carrier on the airport, competition index – e.g. HHI, whether there are any low-cost competitors)
- competition on the same route (ratio of offered seats over all similar flights)
- itinerary characteristics (number of stops, layover duration)
- size of departure/arrival airport (total daily passengers, total number of connecting airports)
- market size of departure/arrival city (city size, population, income per capita)

As explained above, we will ignore such features, and instead focus on features that mostly affect price fluctuation:

- number of unsold seats (and recent fluctuations on that number), as a measure of demand
- state of competing options on the same route (number of available seats on similar flights, current price for those flights, recent price fluctuations)
- date/time of booking, especially days left until departure
- recent price for the same ticket, and recent fluctuations of the price

We chose to consider only features in the last two categories, specifically the number of days left until departure and the fare price on the previous 7 days (relative to the current day's price). The historical information that Bing Travel recorded was the lowest fare across all airlines, so the second parameter (competition) was irrelevant in our case. As for the number of unsold seats, besides the fact that there is no easily accessible historical data on it, it would be of limited use for training our system, because the number that is released to ticket reservation systems is only a lower bound on the actual value (i.e., “4 available seats” actually means “at least 4 seats left”), and the same seat may be sold as part of multiple fares. We had hoped to use it in conjunction with the number of available seats as shown on airplane seat maps, but this information is also inaccurate: tickets may be sold with no assigned seats, and a

portion of the seats may be held for assignment at the gate, so the seats shown as “available” on a seat map may actually be over- or under-estimating the actual number of available seats.

Hypothesis

Our overarching goal is to help ticket buyers make optimal purchasing decisions. In pursuit of this goal, there are various levels of granularity we could choose for our predictions:

- predict the price of the ticket on an arbitrary future date
- predict the minimum value that the fare is going to reach from today up to the day of the flight (or up to a number of days in the future)
- predict if the price of the ticket will drop in the future

We chose to use the third option, essentially viewing our application as a supervised classification problem.

Training & Evaluation

We used the Weka Machine Learning Suite to experiment with different algorithms, and tested their effectiveness using 70% Hold-out Cross Validation. Among the top performing algorithms were the following (the distribution of positive and negative instances in our dataset was 57% – 43%):

Algorithm	**Accuracy**	**Negative examples**		**Positive examples**	
		Precision	**Recall**	**Precision**	**Recall**
Ripple Down Rule Learner	74.5%	0.726	0.542	0.753	0.872
Logistic Regression	69.9%	0.613	0.59	0.75	0.77
Linear SVM	69.4%	0.6	0.627	0.76	0.737

Also of note is the behavior of the simple Decision Table algorithm, which achieved 68% accuracy using only 2 features: the number of days until departure and the price on the previous day. The resulting rules table was heavily weighted towards “True”, and as a result was excellent at correctly classifying the cases where prices would indeed drop (96.2% recall for positive instances), but very inaccurate on negative instances (22.9% recall for those).

It is interesting to note that getting good performance on a metric like the above might not actually translate to good performance on the application we care about, i.e. advising consumers on their buying strategy. A more realistic evaluation strategy would be to use our predictor to give recommendations to hypothetical buyers under various situations, then check whether following our strategy would actually save them money, and/or how close our advising scheme is to the optimal (i.e. an oracle on future prices).

Such a system would work as follows: Given a flight that the user is interested in booking, it runs the predictor on said flight, and, based on the result, advises the user to either buy right away (“buy”), or wait for a predicted future drop in price (“wait”), perhaps with an accompanying measure of confidence. The default advice, in the absence of sufficient evidence suggesting the opposite, would be “buy”. The maximum expected gain would be compared to the potential loss (perhaps relative to the current price), to quantify whether the projected gain justifies the risk.

Future Work

The greatest shortcoming of this work is the shortage of data. Anyone wishing to expand upon it should

seek alternative sources of historical data, or be more methodical in collecting data manually over a period of time. Additionally, a more varied set of flights should be explored, since it is entirely plausible that airlines vary their pricing strategy according to the characteristics of the flight (for example, fares for regional flights out of small airports may behave differently than the major, well-flown routes we considered here). Finally, it would be interesting to compare our system's accuracy against that of the commercial systems available today (preferably over a period of time).

References

[1] To Buy or Not to Buy: Mining Airfare Data to Minimize Ticket Purchase Price, Etzioni et. al, SIGKDD 2003

[2] Modeling of United States Airline Fares – Using the Official Airline Guide (OAG) and Airline Origin and Destination Survey (DB1B), Krishna Rama-Murthy, 2006

[3] A Regression Model For Predicting Optimal Purchase Timing For Airline Tickets, Groves and Gini, 2011