Predictions with Big Data

By Devavrat Shah

The primary vision of big data is the ability to make better decisions using the wealth of information available in data. So far, we have failed in realizing it.

Even though big data infrastructure is a solved problem, we cannot efficiently predict unknowns using data. This is the main impediment when making meaningful decisions using data. To address this challenge, we need an ultimate prediction engine that can consume large amounts of unstructured data and provide accurate predictions of the unknowns.

Background

Big data infrastructure is a solved problem. We know how to collect massive amounts of data (e.g., web scraping, social media, mobile phones), how to store it efficiently to enable queries at scale (e.g., Hadoop File System, Cassandra) and how to perform computation at scale with it (e.g., Hadoop, MapReduce). And yes, we can visualize it, too (e.g., *New York Times* visualizations). However, we are unable to make meaningful decisions using the data. Actually, we are terrible at it. The primary reason for this is our inability to grapple with the uncertainty — we cannot predict the unknowns well.

The next important innovation has to overcome this fundamental limitation. We need an ultimate prediction engine, an oracle, a prophet, the clairvoyant, the astrologer, and the fortune teller of big data. We can throw all the data we have at it to be processed, resulting in answers to questions about the unknowns. Of course, it cannot always predict the unknowns with certainty, but it will provide us the confidence associated with predictions based on the data made available to it. If we can achieve this, the ultimate dream and vision of big data will be realized.

Thinking About Data

It is important to understand how to think about the available data in order to build an ultimate (or for that matter, any) prediction engine. To help us model the data in our minds, consider the following example of data being stored in a potentially giant Excel file with multiple sheets. The data is organized in each sheet separately depending upon the type. The columns in each sheet correspond to different fields, while rows correspond to different records and each cell contains the specific information. Unlike a standard Excel file, in this model, each cell can have all types of information such as text, an image, an audio clip, or even a video in addition to the standard Excel data formats.

Through the lens of this mental model, let's consider an example scenario for the Intelligence Community. There is a wealth of data available about various forms of resource deployment including human and physical; communication between individuals and groups is available across various media; and financial transactions along with historical information about events of interest are available.

For all of this information, we may have data stored in different Excel sheets: one describing resource deployment, another describing communication between individuals and groups for each type of media, one containing financial transactions, and one containing historical information about events of interest. The information in cells of Excel sheets may be complex, such as audio, video, or text. Such a collection of information could be useful for predicting an impending event before it happens.

As another example, consider a retail organization where any operational decision involves knowing the unknowns about customers. Concretely, a retailer like Amazon would like to know what a customer may be interested in purchasing based on her or his recent and past history. The corresponding data involves the customer's history of purchase and transaction data, browse and search logs, reviews provided, and complaints filed. In addition, the retailer may have access to customer demographic information potentially obtained using third-party databases. And of course, product catalog information such as brand, textual description, price, and image are known.

Similar to the earlier example, all of this data can be viewed as a giant Excel sheet. Purchases on each row correspond to different transactions with each column containing a different attribute of the transaction, such as the time it was executed, the customer, product, price, discount, etc. Similarly, separate sheets for browse and search logs, reviews, and complaints filed may exist. Furthermore, another sheet can describe customer demographics with rows corresponding to customers and columns corresponding to different properties like age, sex, address, zip code, or ethnicity. Finally, there is a sheet storing product catalog information with each row corresponding to a product, and columns corresponding to price, brand, image, and description. While this data may actually be stored in Cassandra or Postgres, it perfectly fits our mental model of a giant Excel file.

An even simpler example is that of the highly popularized Netflix Prize challenge. Here, information on a large collection of movies and their star ratings given by a number of users are known. The goal is to predict a rating that a user might give to a movie for which his or her rating is unknown. Because the challenge is based only on such ratings, the corresponding data can be represented in an Excel file with a single sheet where each row corresponds to ratings of a user and each column corresponds to a movie. The value in a cell corresponds to the rating of a given user for a given movie.

Predicting the Unknown

In a world where everything is known, all the cells in all the rows and columns of all the Excel sheets are filled. In reality, many of these cells have missing information. The goal of the prediction engine is to fill the missing information for these cells based on all other available information.

In the Netflix example, in an ideal world, we would know the ratings of all users for all movies. In reality (and as in the challenge), only a few ratings for each user are known. The goal is to predict the unknown movie ratings for users — i.e., fill the empty cells in the only Excel sheet of the file. This is known as the recommendation problem, for which collaborative filtering is a popular solution. Indeed, viewed in this special case, the single sheet Excel file with cell information being ratings is the well-examined problem of recommendations or personalization.

Suppose we have an additional sheet in the same Excel file with information about various users' opinions expressed on the Internet Movie Database (IMDb). Then the information about user preferences expressed through IMDb can be used to further enhance predictions of users' ratings on the Netflix sheet. Indeed, this insight was used to show that the release of the ostensibly anonymous Netflix data set during the challenge was not really anonymous, as information about an anonymous Netflix user could be used to identify users from their public profiles on IMDb.[1]

Generally speaking, it makes sense that combining information across the sheets of an Excel file, when they are available, can greatly boost prediction of the missing information.

In the Intelligence Community example discussed earlier, using communication patterns, financial transactions, and mobilization of resources, an impending event, potentially rare, can still be predicted if we have enough collective data across all sheets of the Excel file.

There lies true value in big data, and its extraction relies on an effective prediction engine.

In summary, an ultimate prediction engine should solve the problem of missing information. And it's much more than the classical problem of recommendation or any other known prediction problem, including regression.

The Celect Engine

Celect has made significant progress towards realizing this dream of building an ultimate prediction engine. Celect's prediction technology, powered by the Celect Engine, can accept data essentially in the form of a giant Excel file with multiple sheets. Celect asks the end user to identify the type (or in Celect's language, *action*) for each data unit. In the mental model discussed earlier, types correspond to different sheets. Each data unit then corresponds to a row of one Excel sheet. The columns have natural association to what are called actors, businesses, or features. The value of each data unit can be effectively anything, including numbers, text, images, audio, or video.

The end user, after throwing all the data at Celect Engine, can query the Engine to predict the unknown value in a given cell. And Celect Engine responds with the prediction and a confidence score based on all the available information.

Although this may appear a simple task, in reality there is a problem of sparsity: information is incomplete and often rare. This makes it really hard to predict well. For example, in the case of predicting rare events, we are faced with exactly such a difficulty: the auxiliary information across domains may seem innocuous individually. Only by cleverly stitching together all the data, as performed by Celect Engine, an accurate prediction may surface from the collective data.

Using the retail example we discussed earlier: the interest of a retailer is primarily in the action of customer purchase. However, the data associated with it is actually very sparse — an individual customer buys very few products in any given retailer's catalog over the duration of a year. On the other hand, the data associated with browse and search logs is quite a bit richer. Therefore, by using all such information to predict the relatively rare event of a purchase, one can achieve significantly better accuracy. Indeed, for Celect's retail customers, more accurate predictions lead to better online personalization (20 percent increase in revenue) as well as in-store assortment optimization (7 percent increase in revenue). A reader may be left wondering how well an approach like collaborative filtering (CF) performs. The performance gains obtained by Celect in online personalization are primarily with respect to CF-based solutions. This is principally because CF-like approaches do not stitch together data across Excel sheets and they do not naturally handle complex data forms like text, images, audio, and video.

Similar insights have been remarkably effective across different domains, including financial markets and social media. For example, when utilized for predicting the price of Bitcoin, a simple trading strategy using the resulting predictions led to doubling of investment over a period of 50 days without incurring a giant volatility penalty (concretely, Sharpe ratio is 4.1).[2] In the context of Twitter, it led to accurate prediction of future trends.[3] Specifically, it predicts trends with a true positive rate of 95 percent and a false positive rate of 4 percent. And it does so by delivering predictions on average 1 hour and 45 minutes in advance. A priori, we did not expect it to perform so well given that all prior

attempts in the literature, some with detailed context-specific model, failed at getting close to such a remarkable performance.

Prediction Provenance

Any prediction system will have errors or mispredictions. Therefore, it is important to understand how to handle such scenarios. In the context of Netflix, presenting a wrong movie to customers only so often is inconsequential. However, in scenarios where humans are involved in making decisions based on predictions, if the decisions have critical consequences such as mobilizing expensive resources for the Intelligence Community, mispredictions are expensive. In such scenarios, one way to guard against misprediction is to explain to the end user why the system has made a given prediction and provide the provenance of the prediction. Then the end user can judge whether the prediction is meaningful or not. Providing such evidence for a prediction can also help decision makers interpret the prediction and justify consequential decisions to the rest of the organization, if needed. Therefore, it is important for a prediction system to not only provide accurate predictions and confidence, but also a proof, certificate, or provenance of predictions. The Celect Engine naturally provides a narrative proof — for every prediction, it produces existing data points that are effective witnesses for the prediction, and by expressing the data in Celect's language, it leads to semantic understanding of the proof.

Conclusion

The ultimate vision of big data is to aid decision making using a wealth of information from the data. The key impediment in realizing this vision is the inability to make accurate predictions. We need an ultimate prediction engine. The classical recommendation systems, in a sense, took the first steps towards designing such an engine. The Celect Engine has made significantly more progress towards achieving this ultimate goal. There lies true value in big data, and its extraction relies on an effective prediction engine. Q

Devavrat Shah *is an Associate Professor with the department of Electrical Engineering and Computer Science at MIT. He is a co-founder and Chief Scientist of Celect, which helps retailers decide what to put where by accurately predicting customer choice using omni-channel data. His primary research interest is in developing large-scale machine learning algorithms for massive unstructured data. Shah has made contributions to the development of "gossip" protocols and "message-passing" algorithms, which have been pillars of modern distributed data processing systems. He received the 2010 Erlang Prize from INFORMS. He is a distinguished alumni of IIT Bombay, from where he graduated in 1999 with the President of India Gold Medal.*

ACKNOWLEDGEMENTS

This article is based on numerous discussions and collaborations author has had with colleagues at MIT and Celect. In particular, author would like to acknowledge George Chen, Vivek Farias, Ying-zong Huang, and Vighnesh Sachidananda.

REFERENCES

[1] *How to break anonymity of the Netflix Prize Dataset,* A. Narayanan and V. Shmatikov, IEEE symposium on security and privacy, 2008.
[2] *Bayesian regression and Bitcoin,* D. Shah and K. Zhang, Allerton 2014.
[3] *A latent source model for non-parametric time-series classification,* G. Chen, S. Nikolov and D. Shah, NIPS 2013.