Train, Score, Repeat, Watch Out! Zillow's Andrew Martin on modeling pitfalls in a dynamic world.

08.30.2017

Andrew Martin|

The $1 Million Zillow Prize is a Kaggle competition challenging data scientists to push the accuracy of Zestimates (automated home value estimates). As the competition heats up, we've invited Andrew Martin, Sr. Data Science Manager at Zillow, to write about how his team handles the challenges of delivering new predictions on a daily basis and how the mechanics of the Zillow Prize competition have been structured to account for these challenges. Here's Andrew.

In 2014 when I joined Zillow, I was a year out of my graduate program in Statistics and was super stoked to get to work on one of the most interesting and well known predictive modeling products in the world, the Zestimate. One thing that became clear quite quickly was that the one-off modeling projects I had done in school or for academic researchers whom I consulted for had not prepared me for some of the modeling concerns that were commonplace when you're producing predictions on a daily basis. This was quite surprising given that home valuation prediction is a classic regression problem. Yet, I found that there was much to learn when you take this classic problem and move it to a setting where you are frequently retraining and rescoring your models. I think that sharing some of these learnings could benefit other data scientists and help those who are participating in the Zillow Prize challenge understand some of our competition design choices better.

Challenges In Updating Predictions Daily

Machine learning competitions are a powerful way for Data Scientists to learn a lot about the practice of model building and evaluation and for sponsors to collect new modeling ideas and improve their products. But, in many cases, these problems are simplifications of the predictions problems faced by practicing Data Scientists -- simply by the nature of the contest setup. Performance in a Kaggle competition is based on predictions made against a problem that is fixed in time. That's a major simplification. Here at Zillow, Zestimate home valuations are re-generated for all 100M+ homes in the United States on a daily basis. Updating estimates frequently is common practice in industry: from predicting what is the best time to buy an airline ticket or who will win an election. But, with this need for updating comes a host of additional considerations for Data Scientists that cannot be ignored if you want your predictions to seen credible.

Updating your predictions about a specific target frequently adds an extra dimension to all the stages of the work of a Data Scientist from feature engineering through deploying your new and improved estimates. While some of these concerns are more well-known I think others get less attention so I want to highlight them. They include the need to expand the metrics you measure your predictions with beyond simple accuracy, taking care when evaluating new models that you replicate the state of production systems in the past, and recognizing the human dimension and how past predictions are interpreted when deploying your improved estimates.

Feature Engineering - The danger of feedback loops

Feature engineering on a static problem is rather straightforward but once you introduce time into the problem, things get much more tricky. One of the most insidious problems that can come up is that of feedback loops. At a high level, what happens is that you have a false correlation between the target you're trying to model through a dependent feature that is influenced by the predictions your models made in the past. I'll give an example for the Zestimate, if we fit a model that uses yesterday's Zestimate value as a predictor of sale price as part of calculating today's Zestimate (i.e. we use a lagged Zestimate as a dependent variable), the resulting

model will lean heavily on the prior day's value. This occurs because Zestimates are often close to sale prices and so when training the model they appear to be a feature with lots of signal. Even under standard cross-validation strategies, models built with this feature will appear quite good. In reality the models you build will actually lead to trouble. A model with a lagged version of the prediction is an autoregressive model, even if not in the traditional linear time series sense. Like all such models, the problem of explosive behavior becomes a real concern. To see this, imagine that future predictions rely on yesterday's prediction plus some trend, but yesterday's predictions was built off of the prediction from two days ago plus some trend, and so on. The trouble starts when the dependence upon the prior value is so heavy that it's like having a linear model with a beta greater than one, in this case, your predictions will amplify themselves until they blow up towards infinity. Everyone has experienced this style of explosive blow-up when someone brings a microphone too close to the speakers of an amplified sound-system and suddenly a quiet tap is recycled into an ear-piercing screech over a few seconds by the resulting feedback loop.

Feedback loops can also occur via variables that are not direct output of an earlier model. Imagine a model that is built to decide which homes to recommend to some users visiting Zillow which uses clicks or other interaction counts with the content it is recommending as dependent variables. Here, once you turn on the model, you create a feedback loop between the model and user behavior by not showing some homes to users based upon what's being clicked on but since click counts are used to predict what to show, the model will quickly decide to only show homes that users have already clicked on. Which homes gets shown isn't likely to be the best recommendation but simply the ones that was randomly shown to the first few users and so had their count totals updated. Detecting these sorts of loops is difficult, sometimes it's not obvious which dependent variables might be causally connected to the model's predictions. An additional safeguard to use when deploying models like these is an A/B test of the model vs. the previous solution that monitors the key performance metrics the model is supposedly optimizing. These

sorts of problems just don't crop up in standard textbook problems or ML competitions where you only have to make one prediction at a fixed time point.

Data Isn't Static - Time delays and data value updates complicate performance evaluation

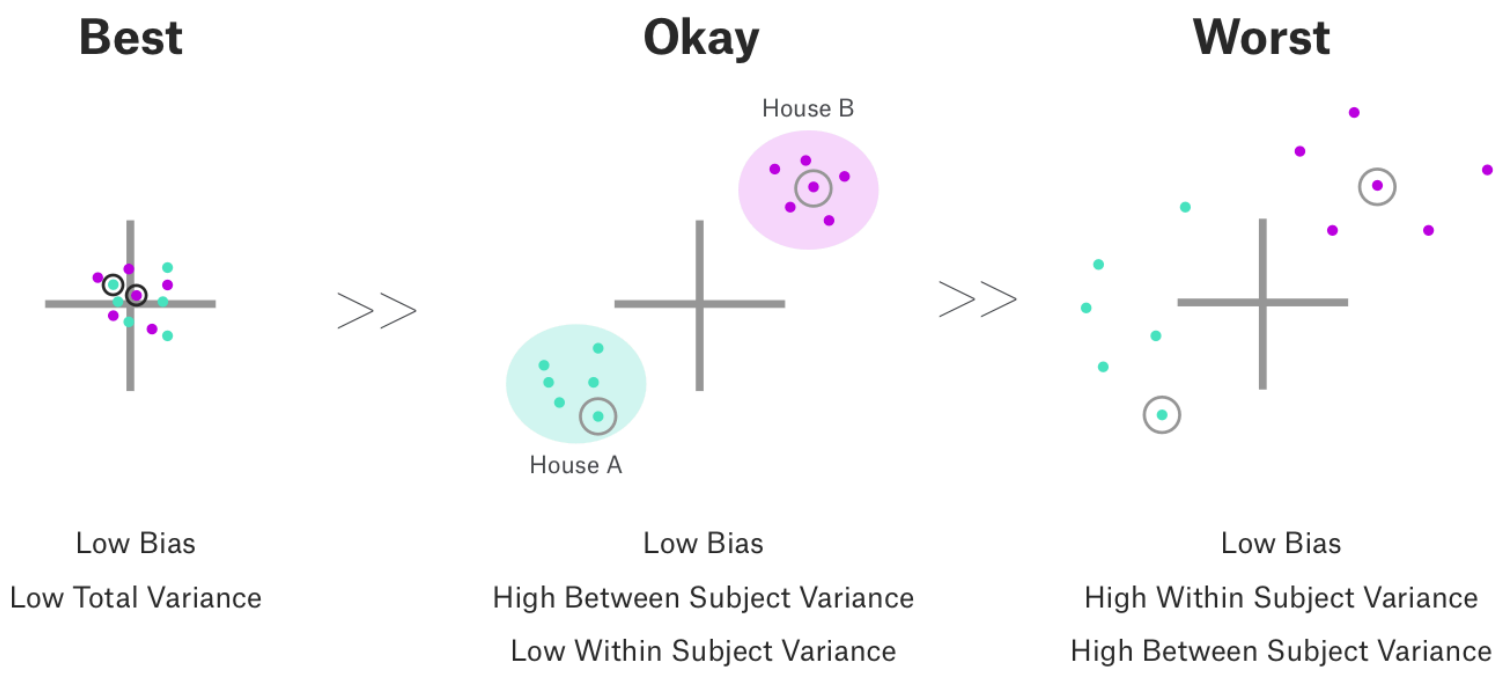
Chances are that when delivering a predictive modeling product that the data you base your predictions on is changing. Practice problems and competitions are often constructed with data sets collected long after the prediction event in question occurred. But making predictions in real-time, i.e. nowcasting, is plagued by lagged delivery of information. For home value prediction this is the natural state of the world for the ground truth labels - i.e. sales. Those familiar with residential real-estate in the United States know that it often takes more than a month to close a transaction after a buyer and a seller agree to the sale price, then the sale must be recorded by an MLS or county assessor and transmitted to Zillow. This means that we effectively face a forecasting problem when trying to predict the value that a home would sell for today since all the ground truth labels are likely at least a month out of date. In order to even test models that are effective given these data lags we ended up having to build a very sophisticated system for re-creating the data state that accurately reflects the changing nature of our data.

While the current state of the residential housing stock is what we need for producing today's Zestimate we have to be careful when putting together data to train and benchmark our models. A home that sold four years ago and was remodeled with an additional 500 square feet of living area, added as part of a master bedroom and bathroom addition creates problems when back-testing or training a model if you use the current home state. The past sale was of the smaller, un-remodeled home and that should be reflected in the sale price. If we instead try and train a model based upon the current state and that four year old sale price we're introducing erroneous signal into the model. Similarly, when we go to evaluate potential model improvements with historical data we need to be careful to

be faithful to the lag in sales data by not including sales we didn't have at the time we are training and predicting. Failure to carefully do this back-testing would result in biased model performance assessments and could lead to picking the wrong spot on the bias-variance tradeoff.

Evaluating Temporal Consistency - a new piece to the bias-variance trade-off

Most predictive modelers are very familiar with the bias-variance tradeoff. You can often get more accuracy from a model if you're willing to accept some bias (systematic shifts) in your predictions if in return the variance (spread) in your predictions is smaller. Many popular ML algorithms use some form of randomness (bootstrap sampling, random starting weights, drop-out, etc..) to improve accuracy but at the expense of potentially increasing variance. The usual ways of balancing these trade-offs via cross-validation optimize a cross-sectional bias-variance tradeoff. It's a cross-sectional method because only one of the many predictions made for each target is usually paired with a ground truth label. But, when you are making repeated predictions on the same targets you also need to be sensitive to the temporal (within group) variance or what we like to call temporal consistency. Optimizing accuracy metrics calculated via cross validation does not distinguish between solutions with high or low temporal variance since it finds the best trade-off between total bias and total variance.



When evaluating the predictions for two homes the usual method pairs only one of a series of predictions (blue circled points) to calculate error (distance from the crosshairs) and its component bias and variance. Minimizing error might not distinguish the situation in the center from the right despite the clearly better consistency of the center model.

The idea here is that we might be willing to accept additional bias in repeated individual predictions if from day to day the predicted value does not shift dramatically i.e. we have lower temporal (within subject) variance. We find that consumers of predictive models are often quite sensitive to consistency as they perceive (correctly) that poor consistency can be a sign of a bad model. This means that optimizing a single accuracy metric in isolation can lead to bad outcomes. Of course, optimizing on a single accuracy metric is the heart of most machine learning contests but when making predictive models for consumers it's better to look at a suite of metrics and optimize across a few key ones. At Zillow, potential model improvements measured by a half dozen metrics. Accuracy and overall bias and variance is included of course but, we also look at measures of temporal consistency.

Why Zillow Prize is Different

The Zillow Prize competition on Kaggle departs from the usual contest set-up in part to better reflect these challenges. Participants in both rounds of the Zillow Prize will be making multiple forecasts for the price a home might sell at and no one will know which homes sell until after final submissions are made. We also replicate a bit of the dynamic nature of the Zillow database by sharing home attributes for both the 2016 and 2017 assessor files from the three round one counties of Ventura, Orange, and Los Angeles in the first round. Participants who make it to the second round should pay particularly close attention to the issues I've raised above as we plan on sharing much more extensive historical data and so careful construction of offline model benchmarking systems will be critical to their continued success. If you have a question about the topics in this post or about Zillow and the Zestimate in general, good news! I will be

hosting an AMA with Kaggle on Sept 7th at 9:00 am PTD. Keep an eye on the Zillow Prize forums for a post with more details.

Note: For more from Andrew, check out Zillow's Data Science Blog.