Active learning optimization as a function of label cost and mistake cost

Roii Spoliansky | Lead Data Scientist

26/10/2017

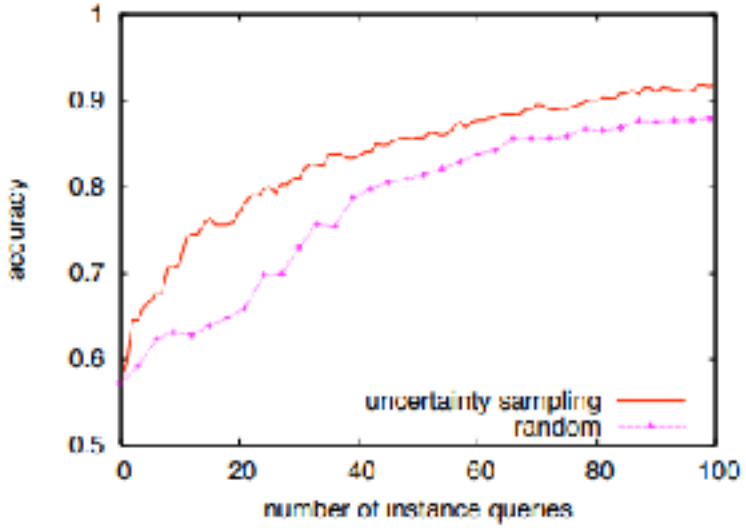
Agenda

- Active learning 101
- My use case for active learning
- Labeling costs
- Integrating the label cost & calculating the mistake cost
- Expected improvement
- Optimizing the number of labels

Active learning 101

Active learning goal

- To deal with unlabeled data in an environment where it is possible to label data, but with a cost



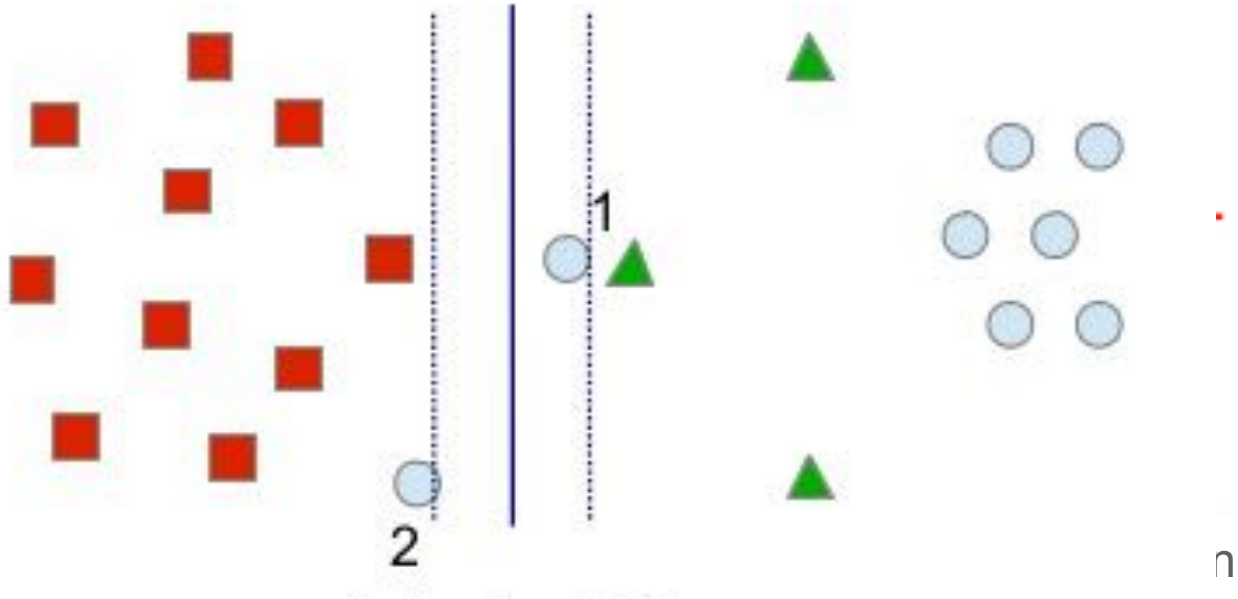
Active learning 101

Active learning goal

- To deal with unlabeled data in an environment where it is possible to label data, but with a cost

Approaches to active learning

- Uncertainty sampling:
  - least confident
  - margin
  - entropy
- Committee of models
- Expected model change
- Version space reduction



It is possible to use with all model ning

My use case for active learning

PayPal has 17 million merchant accounts

- In order to better understand them I wanted to detect their business model
- The smaller sellers will not always have a business model because not all of them are businesses
- I wanted to detect which seller is a business using all the data I have

Problem - I don't have any labeled data

Labeling costs

Limited resources

- Active learning will take into consideration the data and the model
- But what about my labeling capabilities?? – not all samples will take the same time to label

Creating a cost function

- Understanding the business problem (wage model or time per label) or using the initial labeling to learn the cost function for my data

Let’s simplify the problem

- We are trying to classify sentences (1- 10 words) for sentiment analysis
- We have outside workers labeling with a wage model:
  - Per minute
  - Per sentence
  - Per word

Labeling costs

Sentence	1$ Per sentence	0.2$ Per Word	20$ Per Minute
‘Sure…’	1$	0.2$	2$ (6 sec)
‘I think this is good’	1$	1$	0.4$ (1.2 sec)
‘The book you gave me was exactly what I needed’	1$	2$	0.2$ (2 sec)

How to get the cost function →

- Use business understanding
- Use linear regression
- Extrapolate a linear function based on a small number of samples

Integrating the label cost & calculating the mistake cost

The results we have so far

- A vector showing the certainty we have in each sample
- A vector of costs for each sample

Multiplying/weighted multiplying them will give us the best sample to label next

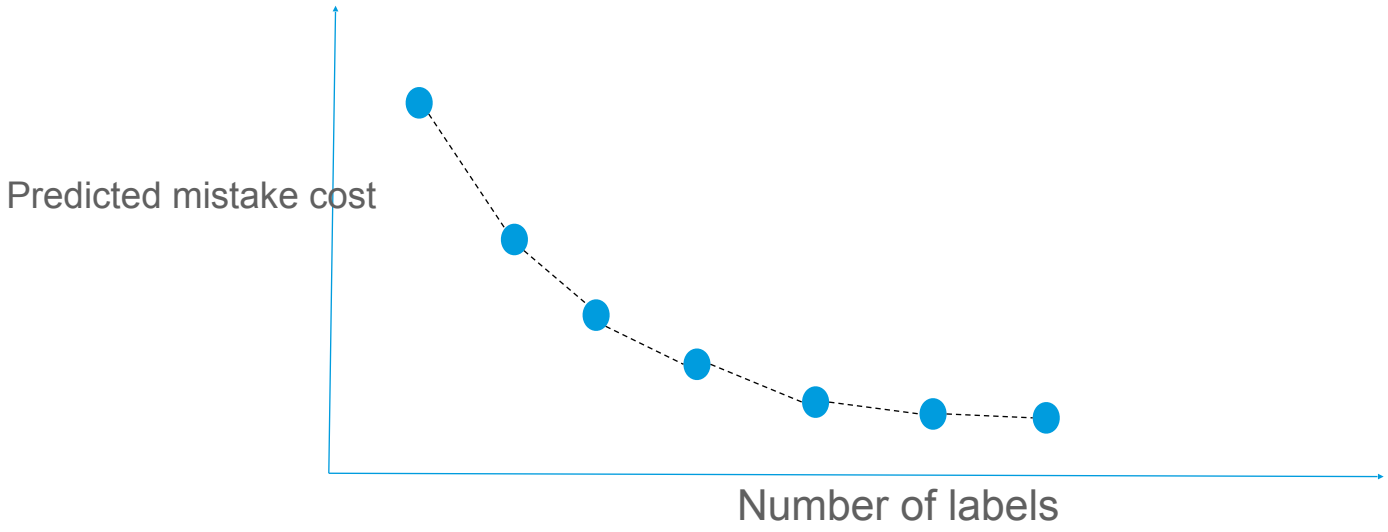
But how many more labels do I need? Is it even worthwhile to keep labeling?

- The certainty vector can be used a possibility of error vector
- Multiplying the FN and FP costs with an adjusted certainty vector will give us our future costs from using the current model

Expected improvement

Using the improvement between label iteration to detect the expected improvement

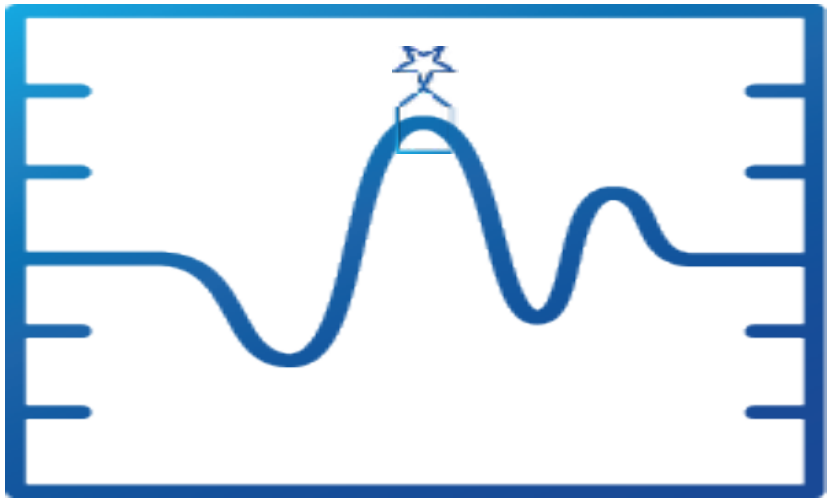
- Run a few iteration of labeling
- Understand the predicted mistake cost for each iteration
- Calculate the expected improvement for the next iteration
- We can also calculate an expected benefit for each unlabeled point (As a function of labeling order)



Optimizing the number of labels

Combine the label cost and mistake cost

- We now know the expected saving from each additional label and the cost of each label
- We can determine the optimal number of labels needed for our model
- If the model is going to work on future data it is possible to calculate the time until return of investment



Tips for a smother journey

- Understanding the business problem and constraints can save a lot of time (always true)
- Cluster the data before initial sampling
- It is better not to use batches of one (even due it is tempting)
- In the learning phase use the same model you are going to use for production

Q&A

And thanks for listening