

Satisficing and Optimizing metric

(SPEECH)

It's

(DESCRIPTION)

Text, Setting up your goal. Satisficing and optimizing metrics.

(SPEECH)

not always easy to combine all the things you care about into a single row number evaluation metric.

In those cases I've found it sometimes useful to set up satisficing as well as optimizing matrix.

Let me show you what I

(DESCRIPTION)

New slide, Another cat classification example. A table is presented. The three columns contain labels of, Classifier, Accuracy, and Running time. The three rows contain the letters A, B, and C. The table is filled as, A and Accuracy, 90 percent. B and Accuracy, 92 percent. C and Accuracy, 95 percent. A and Running time, 80 milliseconds. B and Running time, 95 milliseconds. C and Running time, 1,500 milliseconds.

(SPEECH)

mean.

Let's say that you've decided you care about the classification accuracy of your cat's classifier, this could have been F1 score or some other measure of accuracy, but let's say that in addition to accuracy you also care about the running time.

So how long it takes to classify an image and classifier A takes 80 milliseconds, B takes 95 milliseconds, and C takes 1,500 milliseconds, that's 1.5 seconds to classify an image.

So one thing you could do is combine accuracy and running time into an overall evaluation metric.

And so the costs such as maybe the overall cost is accuracy minus 0.5 times running time.

But maybe it seems a bit artificial to combine accuracy and running time using a formula like this, like a linear weighted sum of these two things.

So here's something else you could do instead which is that you might want to choose a classifier that maximizes accuracy but subject to that the running time, that is the time it takes to classify an image, that that has to be less than or equal to 100 milliseconds.

So in this case we would say that accuracy is an optimizing metric because you want to maximize accuracy.

You want to do as well as possible on accuracy but that running time is what we call a satisficing metric.

(DESCRIPTION)

The accuracy column is circled, and hand-written optimizing nearby. The Running time column is circled, and hand-written satisfying nearby.

(SPEECH)

Meaning that it just has to be good enough, it just needs to be less than 100 milliseconds and beyond that you don't really care, or at least you don't care that much.

So this will be a pretty reasonable way to trade off or to put together accuracy as well as running time.

And it may be the case that so long as the running time is less than 100 milliseconds, your users won't care that much whether it's 100 milliseconds or 50 milliseconds or even faster.

And by defining optimizing as well as satisficing matrix, this gives you a clear way to pick the, quote, best classifier, which in this case would be classifier B because of all the ones with a running time better than 100 milliseconds it has the best accuracy.

So more generally, if you have N matrix that you care about it's sometimes reasonable to pick one of them to be optimizing.

So you want to do as well as is possible on that one.

And then N minus 1 to be satisficing, meaning that so long as they reach some threshold such as running times faster than 100 milliseconds, but so long as they reach some threshold, you don't care how much better it is in that threshold, but they have to reach that threshold.

Here's another example.

Let's say you're building a system to detect wake words, also called trigger words.

(DESCRIPTION)

A line is drawn on the page, creating a new column separate from the table. Hand-written label, wake words and trigger words.

(SPEECH)

So this refers to the voice control devices like the Amazon Echo where you wake up by saying Alexa or some Google devices which you wake up by saying okay Google or some Apple devices which you wake up by saying Hey Siri or some Baidu devices we should wake up by saying you ni hao Baidu.

Oh I guess, you want to read the Chinese, that's ni hao Baidu.

Right, so these are the wake words you use to tell one of these voice control devices to wake up and listen to something you want to say.

And for these other Chinese characters for ni hao Baidu.

So you might care about the accuracy of your trigger word detection system.

So when someone says one of these trigger words, how likely are you to actually wake up your device, and you might also care about the number of false positives.

So when no one actually said this trigger word, how often does it randomly wake up?

So in this case maybe one reasonable way of combining these two evaluation matrix might be to maximize accuracy, so when someone says one of the trigger words, maximize the chance that your device wakes up.

And subject to that, you have at most one false positive every 24 hours of operation, right?

So that your device randomly wakes up only once per day on average when no one is actually talking to it.

So in this case accuracy is the optimizing metric and a number of false positives every 24 hours is the satisficing metric where you'd be satisfied so long as there is at most one false positive every 24 hours.

To summarize, if there are multiple things you care about by say there's one as the optimizing metric that you want to do as well as possible on and one or more as satisficing metrics were you'll be satisfice.

Almost it does better than some threshold you can now have an almost automatic way of quickly looking at multiple core size and picking the, quote, best one.

Now these evaluation matrix must be evaluated or calculated on a training set or a development set or maybe on the test set.

So one of the things you also need to do is set up training, dev or development, as well as test sets.

In the next video, I want to share with you some guidelines for how to set up training, dev, and test sets.

So let's go on to the next.