

Machine learning made easy

- [RSS](#)

<input type="text" value="Search"/>
<input type="text" value="Navigate..."/> ▼

- [Home](#)
- [Contents](#)
- [Ideas](#)
- [Popular](#)
- [Links](#)
- [About](#)
- [Backgrounds](#)

What you wanted to know about Mean Average Precision

2012-08-09

Let's say that there are some users and some items, like movies, songs or jobs. Each user might be interested in some items. The client asks us to recommend a few items (the number is x) for each user. They will evaluate the results using mean average precision, or MAP, metric. Specifically $\text{MAP}@x$ - this means they ask us to recommend x items for each user. So what is this MAP?

First, we will get M out of the way. MAP is just an average of APs, or average precision, for all users. In other words, we take the mean for Average Precision, hence Mean Average Precision. If we have 1000 users, we sum APs for each user and divide the sum by 1000. This is MAP.

So now, what is AP, or average precision? It may be that we don't really need to know. But we probably need to know this:

- we can recommend at most x items for each user
- it pays to submit all x recommendations, because we are not penalized for bad guesses
- order matters, so it's better to submit more certain recommendations first, followed by recommendations we are less sure about

So basically we select x best candidates (in order) and that's it.

Here's another way to understand average precision. Wikipedia says AP is used to score document retrieval. You can think of it this way: you type something in Google and it shows you 10 results. It's probably best if all of them were relevant. If only some are relevant, say five of them, then it's much better if the relevant ones are shown first. It would be bad if first five were irrelevant and good ones only started from sixth, wouldn't it? AP score reflects this.

The name was misleading for us; we'd prefer "order-matters recall" or something like that.

The formula is: $\sum_{i=1:x} (\text{precision at } i * \text{change in recall at } i)$

Precision at i is a percentage of correct items among first i recommendations.

Change in recall is $1/x$ if item at i is correct (for every correct item), otherwise zero. Let's assume that the number of relevant items is bigger or equal to x : $r \geq x$. If not, change in recall is $1/r$ for each correct i instead of $1/x$.

For the interested, Ben Hamner implemented a bunch of [metrics](#) for Kaggle, among them MAP and AP in various languages. You might want to look at some [test cases \(in Matlab\)](#) to check your understanding of how to calculate AP.

For example: `test_case(1:5, [6 4 7 1 2], 2, 0.25);`

This means that actual items were 1:5, that is [1 2 3 4 5]. We recommend [6 4 7 1 2], so we get 4, 1 and 2 right, but with some incorrect guesses in between.

It's AP@2, So in fact only two first predictions matter: 6 and 4. First is wrong, so precision@1 is 0. Second is right, so precision@2 is 0.5. Change in recall is 0 and 0.5 (that's $1/x$) respectively, so `AP@2 = 0 * 0 + 0.5 * 0.5 = 0.25`

Posted by Zygmunt Z. 2012-08-09 [Kaggle](#), [basics](#)

Tweet

+1

0

[Kaggle job recommendation challenge »](#)

Comments

Recent Posts

- [What you wanted to know about AI, part II](#)
- [What you wanted to know about AI](#)
- [Juergen Schmidhuber's answers from the Reddit AMA](#)
- [Torch vs Theano](#)
- [Loading data in Torch \(is a mess\)](#)
- [Interactive in-browser 3D visualization of datasets](#)
- [How to run external programs from Python and capture their output](#)

Twitter

Follow [@fastml](#) for notifications about new posts.

- Status updating...

Follow @fastml

Also check out [@fastml_extra](#) for things related to machine learning and data science in general.

GitHub

Most articles come with some [code](#). We push it to Github.

<https://github.com/zygmuntz>

Copyright © 2015 - Zygmunt Z. - Powered by [Octopress](#)