

LAKNATH'S COLUMN


Ramblings of a wanderer

WHO AM I? HOME

MAR 12 2012
4 COMMENTS

MACHINE LEARNING, SAAS

MACHINE LEARNING IN SAAS PARADIGM



Our ultimate objective is to make programs that learn from their experience as effectively as humans do. We shall...say that a program has common sense if it automatically deduces for itself a sufficient wide class of immediate consequences of anything it is told and what it already knows.

John Mccarthy, “Programs with Common Sense”, 1958

For over a half a century, machine learning has been a strong research topic within the academic circles. But in the last decade or so it has made heavy in roads to the practical world of tech industry and today it's no secret that most of the large players are using numerous machine learning techniques to enhance various aspects of their workflows. In this post I'm

hoping to look at a few ways how a SaaS application (presumably run by a startup without an infinite amount of resources) can use machine learning to enhance its overall experience.

Personalise user experience

In your experience of using SaaS products, how many times you may have found that your favourite item is at the bottom of the list and you have to scroll half a mile or navigate through several layers of menus? Usability usually favours the majority and you may discover the bitterness of being stuck in the minority.

For instance, if you visit the same coffee shop every morning and buy the same drink, and if the shop owners are good at their business then they should know your preferences after a few days and you won't have to go through the ordering routine everyday. It may be just that the shop owner confirms "Same as usual?" and that's it! So if your application is a bit more intelligent (i.e., good at its business), it could take a leaf out of this scenario and save user's time, and in turn, improve customer satisfaction by learning more about their preferences. However to be on the safe side, just as in the case of coffee shop owner's confirmation, you may need to give an extra setting option to the user confirming whether it's preferable for the app to learn user behaviours and adapt.

Reduce your support requests

Say you have a great product on your hands and it's getting more and more traction. If you have experienced this situation, one thing that you won't miss is the number of support requests that would be sky-rocketing in parallel to the hotness of your product. Given that startups have limited man power, human work hours should be efficiently utilized in more rewarding tasks such as adding new features, targeting new sales channels and fixing bugs, instead of spending them answering repetitive support issues that can be easily avoided. One remedy would be to evaluate the usability of your application, which certainly is prudent, but it won't hurt to make your application a bit more intelligent to identify user troubles.

In your neighbourhood if you had noticed a stranger wondering around looking lost, wouldn't you offer your help? Similarly, your app can do the same and be kind enough to identify a new user wondering around and offer help. Not only will you be saving your team's man hours, but

you will be also saving user's precious time, and as an added bonus impress the user even more so in your product.

Make the search intelligent

Search is a window to your application data and improving the quality of search will directly influence the user experience. Rather than making the user guess under what keywords his target content is indexed under, what if your application is good at identifying user intent behind the search? That would certainly be the icing on top of your search functionality. To make it even better, mix some fuzzy-ness to auto correct a search term when there's an obvious error. Of course all this is easier said than done and every company is not a Google. But you can take an initiative by analysing search terms to identify weak spots, start addressing them first and moving forward as a minor experimental optimisation process.

A good application wide search will greatly help answer most questions your users may have. From my experience, most of the support questions are recurring in nature so if users can easily find answers from your community forum or support articles, it will help lighten your support load.

Finding more details about users (enrichment)

Well, this is more of a grey area. Gossiping on other's juicy dirty secrets is usually frowned upon, but a little awareness of what's going on around you could be useful and even healthy. Most of the large companies are already digging up your day-to-day buying patterns to better target you but the amount of how deep you dig into user information (or abstain from it) is certainly up to you. One way to look at this would be how you treat advertisements – as long as they are relevant and useful in achieving your goal you won't mind it. But the second it falls below your requirements and becomes nagging, it will be a nuisance and called spamming. Likewise if you can give users a coupon they can't ignore it's likely they won't mind and you can comfort your conscience by thinking you are doing a service rather than snooping around.

Gauging user reaction to new features

It's a well-accepted practise for applications to use a simple voting system to collect desired new features by their users. Usually what happens is that application admins put up a set of

features they feel important and users vote on them. Taking this approach one step further and factoring each new feature into some set of attributes, you can know what users really need, and at the same time know more about your users. Of course, collaborative filtering (CF) is not a new technology: all social networking services and e-commerce giants are using it to rate new items and know preferences of new users back and forth. But, even without being a massive social networking service, you can still use CF to get to know the attributes of your user base, such as technical savviness, seeking automation. etc.

Conclusion

This only sums up some of the more straight-forward scenarios where machine learning techniques can play a part in improving SaaS applications. It certainly is an exciting field in which I'm trying to get a grasp on as a passive interest and hoping to carry out experiments to learn the applicability of various ideas. It would be exciting to hear more ideas and how well they have worked out for you, so please feel free to share them here.

Some pointers to get started/keep an eye on:

- Good place to have some quick practical experience – [Stanford online ML course](#) by [Prof. Andre Ng](#)
- [CMU Machine Learning course](#) by [Prof. Tom Mitchell](#)
- [Probabilistic Graphical Models](#) by [Prof. Daphne Koller](#)
- Lots of theories by Prof. Andre Ng – [Stanford CS229](#)
- [MLSS](#) resources – [2008](#), [2009](#), [2010](#), [2011](#).
- MIT ML [lecture notes](#)
- Various research publications – [Google](#), [Microsoft](#), [Academia](#)
- Google [prediction API](#)
- What's happening in the world of big data – [NoSQL weekly](#).

4 thoughts on “Machine Learning in SaaS paradigm”

**Greg** says:[March 13, 2012 at 12:29 pm](#)

Love the post. I got similar background and completed Andrew's Ng ML course last year. Still trying to figure out how to plug this knowledge to startup opportunity. Anything that I'm coming so far is a solution looking for a problem. It's frustrating as I got a feeling that there is something going on ML area in the last 5-7 years after 20 or so years of dead end.

Greg

[Reply](#)**Francisco J Martin** says:[April 20, 2012 at 5:02 am](#)

Great post! As you mention: "large players are using numerous machine learning techniques to enhance various aspects of their workflows" but a number of startups will soon democratize Machine Learning and offer extremely easy ways for small and midsize companies to learn from their data. We are executing on that vision at BigML!

[Reply](#)**laknath** says:[April 20, 2012 at 5:42 am](#)

Cool. I like how your solution has a custom model creation option, which differs from Google prediction API's generalized approach.

[Reply](#)**Marketing Agency Chicago** says:[July 12, 2012 at 2:29 pm](#)

Hmm, my brother is working on stuff like this for the government with regards to miniature submarines. Next thing you know we'll have robots with machine learning techniques. Terminator anyone?

[Reply](#)

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

Post Comment

[← Previous post](#) [Next post →](#)

ARCHIVES

- [April 2018](#)
- [September 2017](#)
- [January 2017](#)
- [March 2012](#)

CATEGORIES

- [.Net](#)
- [Backups](#)
- [Cricket](#)
- [Databases](#)

LAKNATH'S COLUMN

- [Is this a toxic comment?](#)
- [Notes from Quora duplicate question pairs finding Kaggle competition](#)

- [December 2011](#)
 - [October 2011](#)
 - [September 2011](#)
 - [December 2009](#)
 - [May 2009](#)
 - [January 2009](#)
 - [December 2008](#)
 - [October 2008](#)
 - [July 2008](#)
 - [June 2008](#)
 - [May 2008](#)
 - [February 2008](#)
 - [January 2008](#)
 - [December 2007](#)
 - [November 2007](#)
 - [August 2007](#)
 - [July 2007](#)
 - [June 2007](#)
 - [May 2007](#)
 - [April 2007](#)
 - [March 2007](#)
 - [February 2007](#)
 - [January 2007](#)
 - [December 2006](#)
- [Drupal](#)
 - [Eclipse](#)
 - [education](#)
 - [FIT](#)
 - [Flash](#)
 - [FOSS](#)
 - [Fun](#)
 - [Gnome](#)
 - [GSoC](#)
 - [inspiration](#)
 - [Java](#)
 - [javascript](#)
 - [machine learning](#)
 - [Maya](#)
 - [Movies](#)
 - [My Activities](#)
 - [NLP](#)
 - [Novels](#)
 - [PHP](#)
 - [politics](#)
 - [Python](#)
 - [saas](#)
 - [servers](#)
 - [Sri Lanka](#)
 - [ubuntu](#)
 - [Uncategorized](#)
 - [web](#)
 - [Web Designing](#)
- [The necessity of lifelong learning](#)
 - [Machine Learning in SaaS paradigm](#)
 - [Avoiding AWS potholes](#)