Minutes for meeting 20/12

Ideas for ways of implementing a decision tree classifier to come up with recommendations was discussed.

The main Idea proposed suggests finding the most influential features in the data set and basing the decision trees on those.

A decision tree will be built for each product, per customer, which will predict if a customer is likely to have that product or not, and a confidence percentage.

From that list, you subtract away the product the customer already has (no point already recommending products to a customer that already has them.) and take the 7 most likely products as the recommendations.

The main fears with this idea is the processing overhead of creating so many decision trees per customer. Work-arounds suggested have been to either only use a portion of the data-set for training purposes (a month or 2 from the data set) or only building decision trees for the products the customer does not currently have.

There also suggestions that the decision tree method is too simple and too computationally expensive to build an efficient recommender system, so one area of research will be looking into other, more suitable algorithms to use.

Ali and Aky to build decision tree recommendation engine

Felix and Datinder to research other algorithms we can use

Georgios, Manuel, Hemesh and Shanta are on holiday.

Lawrence has suggested starting off with a non-supervised clustering of the customers and finding the common products between them, a collaborative filtering approach which Datinder and Felix are now working on and have aimed to make a simple working model by the end of the week.

Ali and Aky have produced a script that cleans the whole data-set and runs a machine learning algorithm to find the most influential features affecting the product purchases.

This will then be used to produce a simple recommender system by the end of the week.

Useful links for today:

<http://subs.emis.de/LNI/Proceedings/Proceedings165/170.pdf>

[https://datajobs.com/data-science-repo/Recommender-Systems-[Zhang-and-Iyengar].pdf](https://datajobs.com/data-science-repo/Recommender-Systems-%5bZhang-and-Iyengar%5d.pdf)

<https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm>

http://scikit-learn.org/stable/modules/neighbors.html