Boltzmann Machine Neural Network

Michael McDermott

December 14, 2014

So you're out shopping at the mall, and accross the way you see an old girlfriend/boyfriend from highschool. Things didn't end well between the two of you, in fact you broke up with them by text message - needless to say you don't want to run into them if you can at all avoid it.

You really want to go to SportChek though to get your new fitbit, but you've lost track of him/her and now you're worried about bumping into them. Should you risk it and go into SportChek?

We can use a Boltzmann Machine to answer this question. Say the following data on shopper patterns at the mall is available:

Sears	Bay	Zellers	Old Navy	Gap	Running Room	SportChek
1	1	1	0	0	0	0
0	0	0	1	1	0	0
1	1	1	0	0	0	0
1	1	0	0	0	0	0
0	1	1	0	0	1	0
1	1	1	0	1	0	0
0	0	0	1	1	0	0
0	0	0	0	0	1	1
0	0	0	0	0	1	1
1	0	0	0	0	1	1
1	0	0	1	0	1	0
1	0	0	1	0	1	0
0	0	0	1	1	0	1
1	0	0	1	0	1	0

From this data you might suspect that shoppers fall into roughly three categories:

1. Customers who tend to shop at department stores (Sears, Bay, Zellers)
2. Customers who tend to shop at clothing stores (Old Navy, Gap)
3. Customers who tend to shop at sporting stores (Running Room, SportChek)

So you create 3 hidden units in your Boltzmann Machine. However, you notice that the last 4 rows don't really fall into either of these 3 categories. You're not sure what they correspond to but it doesn't matter for a Boltzmann Machine, you just add another hidden unit which can detect a new pattern (perhaps the underlying reason for the pattern is that Sears, Old Navy and Running Room are quite close to each other in the mall, but the point is that the reason doesn't matter to the Boltzmann network).

So you run the Boltzmann Machine (you can find the code at github.com/mbmcdermott/) for a Markov Chain length of 2 and you update the weight matrix 4000 times. You end up with the following weight matrix:

	Hidden 1	Hidden 2	Hidden 3	Hidden 4
Sears	-7.99677908	6.25901682	3.06646307	-5.49902637
Zellers	-3.61138119	-6.56738949	7.13570937	-2.86091173
Bay	-3.56465021	-9.95069775	2.13034382	-1.32520726
Old Navy	3.73486804	4.10231254	-5.32997538	-7.59285082
Gap	6.45487962	-3.86794454	-1.00969702	-4.76830704
Running Room	-6.58089255	3.48306101	-4.15908351	7.11534596
SportChek	0.17301091	-4.87537416	-8.41734985	6.48432907

Each hidden unit corresponds to a different feature of the data: 1. Captures the clothes shoppers 2. Captures the "proximity" shoppers (this was the unit that was added due to the noticed differing) 3. Captures the departmen store shoppers 4. Captures the sport store shoppers

Armed with the weights you can now finally answer your question of whether you should go into SportChek. You have observed that your ex is a total shopaholic and has gone to every store on the list. Most people would probably at this point assume that your ex will likely go to SportChek since they've been to every other store. But you have a Boltzmann Machine.

To get an estimate for the probability of going to SportChek one can take a large sample (20000) of hypothetical shoppers that went to every store and also of hypothetical shoppers that went to every store **except** SportChek (since the ex went to every other store and we don't yet know about SportChek). Then since one has the weights at equilibrium one can just simulate the system with those hypothetical visible unit vectors and take the average. Thus one basically does a reconstruction step (only one since we don't want to step away from the equilibrium we have attained) for every sample.

The result you find is that even though your ex has gone to every other store, she is only about 28% likely to go to SportChek (this makes sense from the data since it's quite rare to go to SportChek if shoppers have gone to department stores or clothing stores). You really want that fitbit, so you take your 2/3 chances and go to SportChek.