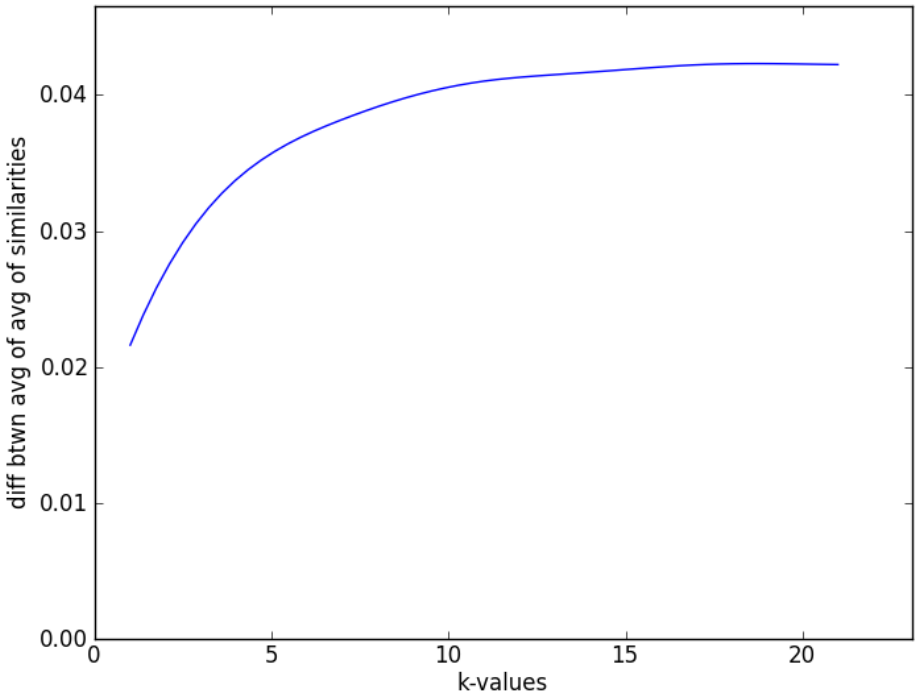
Performance of LSH Compared to Brute-force With Respect to Average Similarity and Running Time

Daniel Alabi and Cody Wang

We evaluated the performance by calculating the difference between the average of the average similarities using the brute-force approach and the LSH approach (Δs). We plotted these values against each of the four different independent variables: the number of nearest neighbors, k; the number of rows in each band, r; the number of rows in the signature matrix; and the dataset size, while keeping the other three factors constant.

We also evaluated their performances by calculating the running time for each.

Variant: k

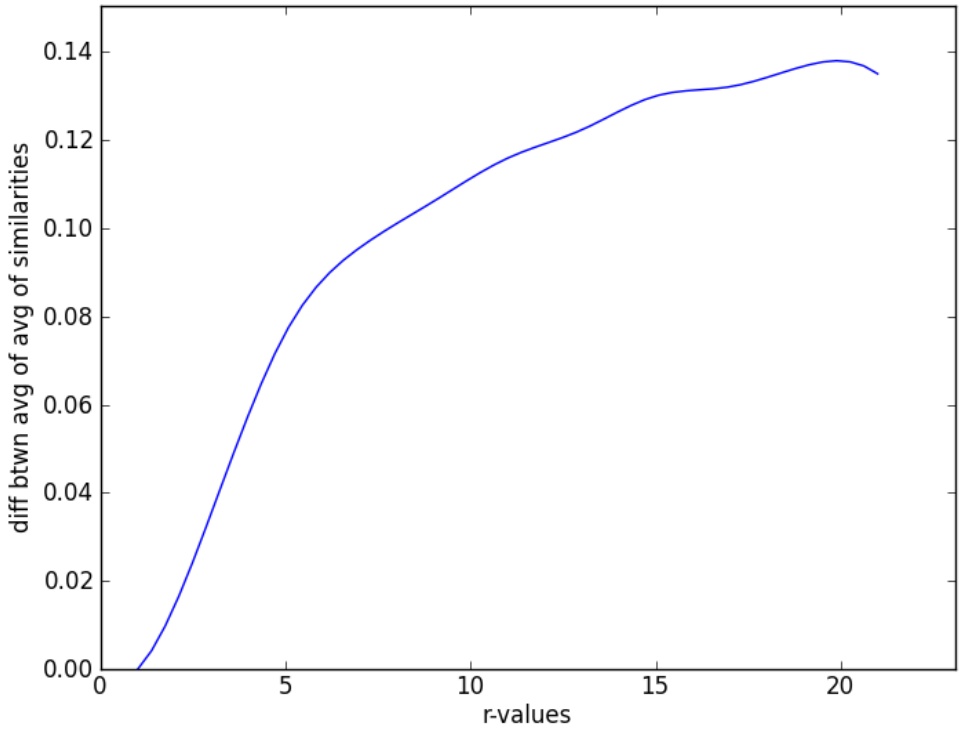


Δs increases as the number of nearest neighbors used increases, although starting at k > 10, this value seems to flatten out.

The reason for this is that as k increases and the other variables stay constant, the more neighbors will be chosen at random for a document in the

LSH approach, while the number of true nearest neighbors found from the candidates stays constant. Thus the LSH method becomes less accurate, so Δs increases. As the number of random neighbors increase to a certain amount, the average becomes more consistent because the neighbors are mostly randomly chosen.

Variant: r



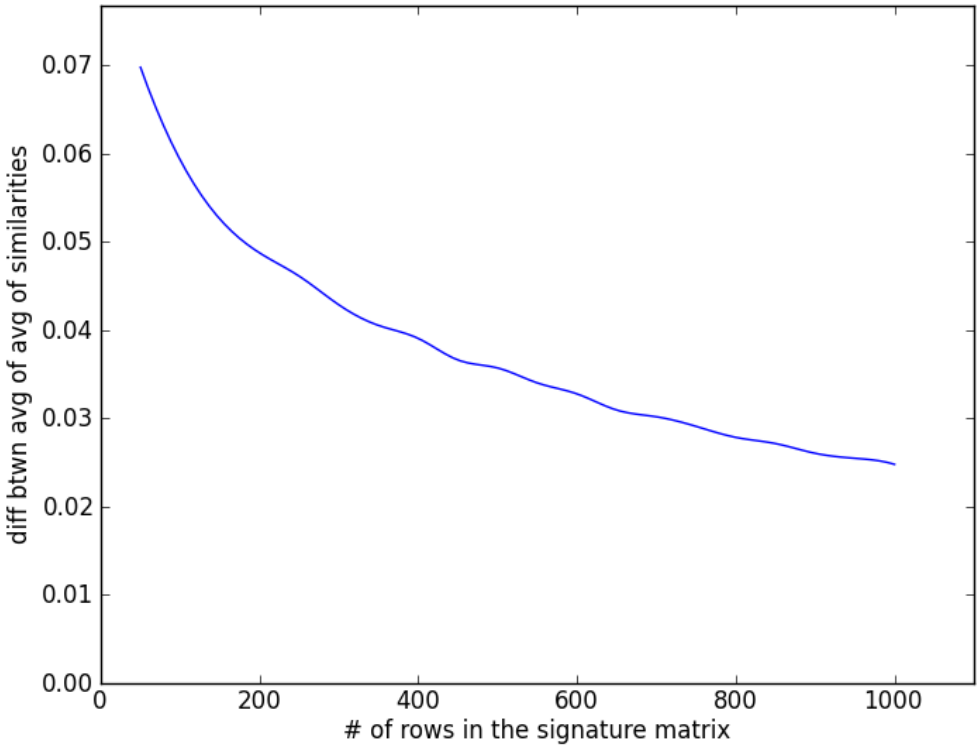
Δs increases as the number of rows in each band increases.

The reason for this is that as r increases and the other variables stay constant, the less likely potential candidates will be found for a document in the LSH approach. Thus the LSH method becomes less accurate, so Δs increases.

Variant: number of rows in the signature matrix

Δs decreases as the number of rows in the signature matrix increases.

The reason for this is that as the size of signature matrix increases and the other variables stay constant, the more likely potential neighbors will be chosen as candidates for a document in the LSH approach because there are

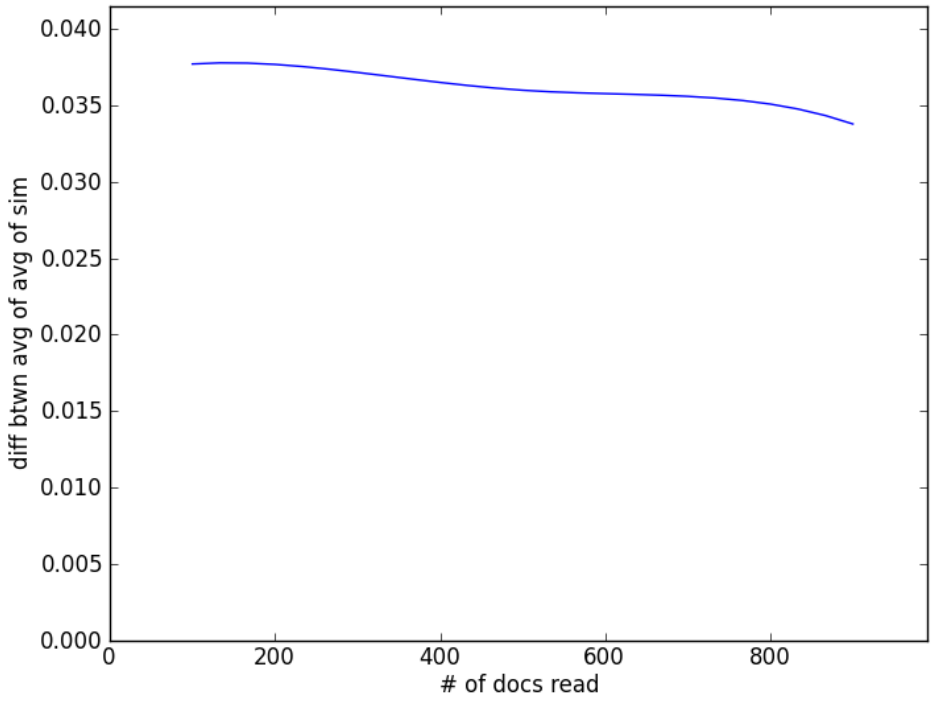


more bands (which is number of rows in the signature matrix / number of rows in each band). Thus the LSH method becomes more accurate, so Δs decreases.

Variant: dataset size

Δs slowly decreases as the dataset size increases.

The reason for this is that as the dataset size increases and the other variables stay constant, the more consistent the data will be (lower standard deviation from the mean), and Δs will show a less spread. Also more potential neighbors will be found for each document, so the LSH method becomes more accurate, thus Δs decreases.



Running Times:

The running time for LSH is almost always less than that of the brute- force approach. Below is the chart showing time elapsed for varying number of rows in each band (r) while holding k=10, dataset size=1000, and number of rows in the signature=200 constants.

	LSH	Brute-force
r=1	9.36	8.38
r=2	8.29	1.41
r=5	8.29	0.78
r=10	8.20	0.73
r=20	8.58	0.72

As the number of rows in each band increases, the running time for LSH decreases, and the difference between the running times of the two

approaches increases since the running time for the brute-force approach does not depend on this variable.

However, if the number of rows in each band is one, the running time for LSH becomes a little more than that of the brute-force approach because they use the same signature matrix, but LSH takes extra time for using the banding technique (computing the hashes to the buckets).

Overall, LSH outperforms the brute-force approach in running time, and can retain a rather accurate result by keeping the rows in each band relatively low.