SSL via Locally Sensitive Hashing (LSH)

Judith Abécassis, Timothée Lacroix

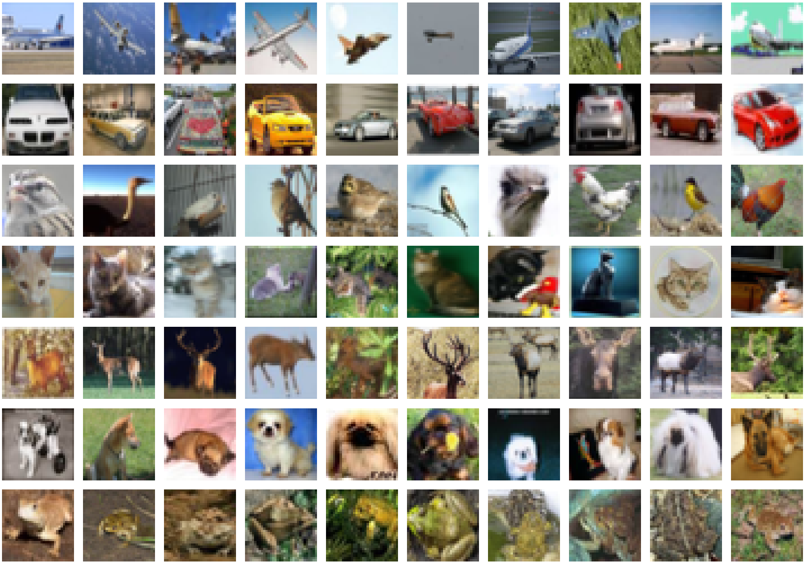
Graphs in Machine Learning

April 2015

Table of contents

1. Challenge
2. Semi-Supervised Learning Framework
3. Locality Sensitive Hashing
4. Theory
5. Experiments
6. Conclusion

Classify a lot of elements in a SSL context



Leverage information from unlabeled nodes in the graph

The harmonic solution

Issues arise with the size of the graph

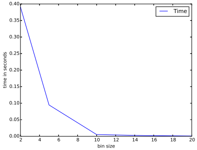
Presentation of the principle

Accuracy

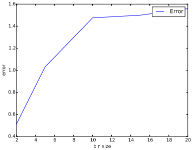
Limitations to obtaining a bound for LSH

Empirical results on LSH accuracy

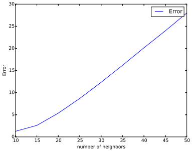
- Instead of getting a theoretical bound, we have explored in a practical setting the error made by LSH



(a) Average time for a nearest neighbor query, as the number of bucket increases



(b) Average normalized error for an approximated line of the Laplacian and an exact line, as the number of bucket increases



(c) Average normalized error for an approximated line of the Laplacian and an exact line, as the number of neighbor increases

The tinyImage dataset and preprocessing steps

- 80 million unlabeled images
- CIFAR-100 labels: 10 classes with 600 labeled images in each
- enriched with GIST descriptors (384 dimensions)
- PCA on the 80M images: down to 32 dimensions

Setting to assess performance

- we consider C=10 random classes
- for each class, we select t positive and t negative (in one of the remaining $100 - C$ classes, $t \in \{0, 1, 2, 3, 5, 8, 10, 16, 20, 40, 60, 100\}$. Those are labeled nodes from the training set.
- we compute the lerning step (by LSH+HFS or Fergus algorithm)
- we select 100 positive test images and 200 negative ones for each of the C classes (unlabeled)
- we measure precision at 15 % recall.

Problems with LSH and GRaphlab implementations

We have encountered several issued due to computation time while applying LSH+HFS method

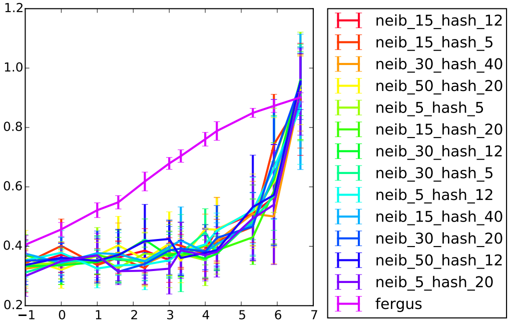
- computation of the graph is long (2600 sec)
- one propagation step is long too (1100 sec)

We have adapted the setting to save some computation time

- only positive and neutral labels (0 and 1s), to allow doing one propagation for the C classes and not C propagation steps
- only consider the first 25 classes, and not 100 (otherwise it is even longer)

Problem results are no longer comparable to the Fergus baseline

Results


1.2
1.0
0.8
0.6
0.4
0.2
−1
0
1
2
3
4
5
6
7
neib_15_hash_12
neib_15_hash_5
neib_30_hash_40
neib_50_hash_20
neib_5_hash_5
neib_15_hash_20
neib_30_hash_12
neib_30_hash_5
neib_5_hash_12
neib_15_hash_40
neib_30_hash_20
neib_50_hash_12
neib_5_hash_20
fergus

Conclusion

- we have encourageing results on LSH performance
- computational limitations of current implementation prevent from proper testing of LSH+HFS methodology

Future work

There are some variants of LSH method that learn hashing on data and could prove even more efficient.