Learning the IHMM through iterative map-reduce

Sébastien Bratières, Jurgen van Gael, Andreas Vlachos, Zoubin Ghahramani
University of Cambridge

Context

Web-scale datasets frequently cannot do without distributed learning. Indeed, parallelization and acceleration methods are not sufficient to make learning tractible in a realistic amount of time.

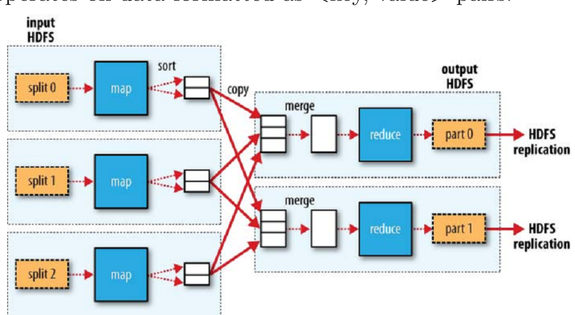
Non-parametric models, like Dirichlet process-based models, the Indian buffet process, or the infinite HMM used here, are very attractive from a theoretical point of view: their inference does not require model selection in order to fit "capacity" parameters, such as number of states or clusters – these are learned just like any other parameter.

Here, we selected one such model, the infinite HMM; applied it to a task, part-of-speech tagging; implemented it on the map-reduce platform Hadoop; and examined execution times.

Our learning algorithm is based on Gibbs sampling, thus requires several thousands of iterations. It turns out that the per-iteration-overhead required by Hadoop makes it prohibitive to iterate as often as this. Map-reduce implementations targeting precisely this requirement, like Twister, currently still under development, are expected to bring a real advantage.

Map-reduce with Hadoop

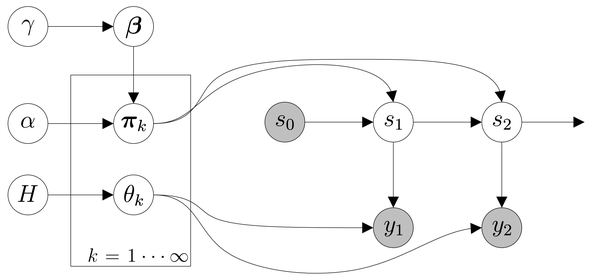
Hadoop is an open-source Apache project which implements the map-reduce distribution paradigm. Map-reduce operates on data formatted as <key, value> pairs.



Steps:

- split input data (<K1, V1>) into chunks
- execute a task on each chunk (the map task): obtain intermediate data <K2, V2>
- sort on K2
- send each set of K2 to one reducer
- perform the reduce task: obtain final data <K3, V3>
- (possibly sort again, on K3)

The infinite HMM



- State sequence $s_1, s_2, ..., s_t$. A state stems from a collection of K states.
- Observation sequence $y_1, y_2, ..., y_T$, with observations stemming from a vocabulary of size V.
- Transition matrix $\boldsymbol{\pi}$ consisting of rows $\boldsymbol{\pi}_k$, with $\pi_{kl} = p(s_{t+1} = l | s_t = k)$
- Emission vector θ_k for each state, of length V. $p(y_t | s_t, \theta_{s_t})$ is a simple categorical distribution.

The prior for $\boldsymbol{\pi}_k$ is a Dirichlet distribution with concentration α and base $\boldsymbol{\beta}$, drawn from a symmetric Dirichlet distribution parameterized by γ. Emission vectors have as prior a symmetric Dirichlet distribution whose parameter is H.

Here is the outline of the inference algorithm for the IHMM, applied to PoS tagging, with individual steps implemented as map-reduce jobs:

- count transitions (map over sentences)
- draw each $\boldsymbol{\pi}_k$ from a Dirichlet($\boldsymbol{\beta}$+counts) (map over states)
- count emissions (map over sentences)
- draw each θ_k from the posterior, a Dirichlet(symmetric H + counts) (map over vocabulary)
- sample auxiliary variables (used for beam sampling, an instance of auxiliary variable MCMC) from each sequence of two states in sentences (map over sentences)
- with reference to the Chinese Restaurant representation of the Dirichlet Process, sample the number of tables used by the state transitions (considering each transition in the counts a new customer) (map over elements in the transition matrix)
- based on table counts, resample $\boldsymbol{\beta}$
- expand $\boldsymbol{\beta}$, $\boldsymbol{\pi}_k$, θ_k
- run the dynamic program to sample new state assignments (map over sentences)
- clean up (prune unused states) $\boldsymbol{\beta}$, $\boldsymbol{\pi}_k$, θ_k (map over sentences twice: first take not of used states, then remove unused ones)
- ... and iterate

Experiments

We ran different versions of the same algorithm on different sizes of learning corpus, and measured iteration duration. The learning corpus consisted of Wall Street Journal sentences: we used subsets of 1e3, 1e4 and 1e5 words, the entire corpus of 1e6 words, and created an artificial data set of size 1e7 by duplicating the 1e6 data set.

Configurations were as follows:

parallel implementation of the iHMM in .NET which uses multithreading on a quad core 2.4 GHz machine with 8GB of RAM

hadoop-1-* Hadoop version, running on a cluster, where each iteration contains 9 map-reduce jobs; * stands for the number of slave nodes; hardware: Amazon "small" type, i.e. 32-bit platforms with one CPU equivalent to a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor, 1.7 GB of memory, with 160 GB storage.

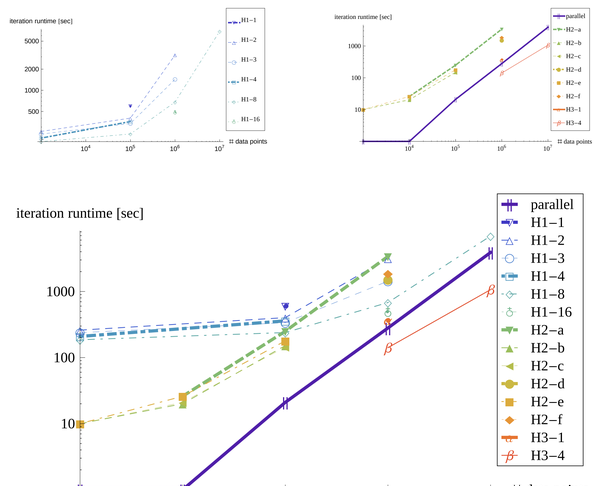
hadoop-2-* Hadoop version, where each iteration contains only one map-reduce job, the most CPU-intensive one (the dynamic programming step); same hardware, always just 1 slave node

hadoop-3-* same as *hadoop-2* but on more efficient hardware: Amazon "extra large" nodes, 64-bit platforms with 8 virtual cores, each equivalent to 2.5 times the reference 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor, 7 GB of memory, 1690 GB of storage.

Here are the 18 clusters obtained after 1160 iterations, starting from 10 clusters, learning from a 1e6 word corpus. A token appears in a cluster if for the last Gibbs iteration, it was assigned at least once to the corresponding state. Tokens are ranked inside each cluster according to a relevance metric, the log-likelihood ratio. Tokens may be repeated among clusters, as there is no deterministic token-to-cluster assignment.

```
SENTENCE_END
SENTENCE_START
., ?, !, ", %, quips
it, n't, he, which, they, be, ■, share, she, there, 's, more, who, not, Japan, them, expected, well, Congress, but
, of, to, in, and, the, 's, for, is, that, on, by, with, from, at, as, are, was, will, said
The, In, But, ", ", A, It, He, They, Mr., And, That, If, For, This, As, At, a, Some, While
coupled, crocidolite, success, blacks, 1965, 8.07, Arnold, Richebourg, conversations, new-home, smokers, ward,
worksheets, prove, Bell, Darkhorse, Italy, Panama, buses, insisted
the, a, $, its, of, an, and, their, Mr., his, New, no, other, any, this, these, recent, some, The, two
also, We, spokesman, Corp., I, addition, There, company, admitting, Computer, declined, Acquisition, L., Ross,
White, House, you, has, Inc., officials
%, million, ■ company, market, billion, York, trading, cents, year, president, Inc., ", funds, shares, months,
days, is, markets, yen
Yeargin, High, end, carrier, Hills, dollar, lot, Angeles, IRS, Orleans, principal, circuit, most, Mississippi,
buyer, Lane, Street, female, Senate, result
Co, N.J, 2645.90, Calif, Mass, UNESCO, fines, soon, pollen, Baltimore, fired, 1956, Conn, Danville, Egypt, anx-
ious, goodwill, privilege, severable, shelter
Trotter, home, 106, 301, Amadou-Mahtar, Appellate, Chadha, Confidence, Corrigan, Crew, Curt, Dakota, Express-
Buick, Federico, Fuentes, Glove, Harrison, Islamic, Jeremy, Judiciary
Always, Heiwado, N.V, Stung, Perhaps, 4, 5, 3, 2, )
River, Jr., States, says, School, Bridge, sheet, voice, Corps, Times, piece, Giuliani, Price, underwriter, Code,
Register, Tashi, accurate, condition, fills
new, of, same, own, vice, first, appropriations, U.S., financial, Japanese, executive, 10, Big, major, 100, good,
stock, previous, bad, joint
Earns
Haruki, Nomenklatura, abuzz, alumni, ambassadors, attended, backyard, boarding, celebrate, diminish, finite, less-
ening, milestones, preventative, profess, rebuild, relax, tapping, thin-lipped, cloth
Danube
```

Results



Next steps

1. Reproduce the experiment with Twister, a map-reduce framework built specifically for iterative algorithms. It leaves the mapper running from one iteration to the next, instead of restarting it. Twister is under development at the University of Indiana. Stay tuned !...

2. Scale from the Wall Street Journal corpus to the English Wikipedia (from 1e6 to 2e9 words). NLP evaluation cannot be against a golden standard, such as the WSJ labels, used so far, so we will resort to indirect methods.

References

[1] M. J. Beal, Z. Ghahramani, and C. E. Rasmussen. The Infinite Hidden Markov Model. *Advances in Neural Information Processing Systems*, 14:577 – 584, 2002.

[2] Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

[3] J. Van Gael, Y. Saatci, Y. W. Teh, and Z. Ghahramani. Beam Sampling for the Infinite Hidden Markov Model. In *Proceedings of the 25th International Conference on Machine learning*, Helsinki, 2008.

[4] Jurgen Van Gael, Andreas Vlachos, and Zoubin Ghahramani. The infinite HMM for unsupervised PoS tagging. In *Proceedings of 2009 Conference on Empirical Methods in Natural Language Processing*, pages 678–687, Singapore, 2009.