Binary Matrix Factorization

Ignacio Ramírez

December 22, 2016

introduction

We consider the problem of approximating a binary matrix $X \in \{0,1\}^{m \times n}$ as the product of two other binary matrices $U \in \{0,1\}^{m \times k}$ and $V \in \{0,1\}^{n \times k}$ plus a small residual,

$$X = UV^T + E. \tag{1}$$

Matrix Factorization The problem (1) has been addressed using various methods from different fields for over a hundred years. Two such fields are Signal Processing and Machine Learning under the so-called "Matrix Factorization" (MF) methods, (with non-negative matrix factorization (NMF) a very popular particular case,) which is very mature and has been developed largely for the case of real matrices.

Dictionary Learning A particular case of MF is the so-called "Dictionary Learning" problem which is particularly useful when $m \gg n$. In this case, contrary to the more general MF approach, the roles of U and V are quite different and, accordingly, both are estimated in quite different ways. More specifically, U is called a dictionary (and usually assigned the letter D), whereas $V^T = A$ is a matrix of "linear combination coefficients". The columns of the matrix D are called "atoms" and are supposed to embody typical patterns observed throughout the columns of X, while the columns of A, usually assumed sparse, specify the linear combination of columns of D that better approximates the corresponding columns of X.

Data Mining On the other hand, there is a large body of work from the Data Mining community on the subject of extracting general "patterns" from large matrices, in particular binary matrices. Some of those works are based on matrix factorization concepts, for example the Proximus algorithm [3],although most of them are quite heuristic.

Other ideas Initially I was interested in connecting the above techniques with other learning frameworks aimed at binary data. In particular, the Bidirectional Auto-associative Memory (BAM) model [2] was an early binary recurrent neural network.

Our approach The idea here is very simple: to apply a Dictionary Learning approach to the problem of binary matrix factorization for those cases where the matrix X is significantly "fat" $m \gg n$ (or, naturally, apply it to X^T when it is "tall", $m \ll n$).

0.1 General Dictionary Learning framework

The problem (1) is generally non-convex. There are very special cases in which it can be solved exactly, or it reduces to a tractable convex problem under particular conditions. Most DL approaches fall into the general setting where (1) is NP-hard and only local convergence (to a stationary point) can be guaranteed. This is usually achieved through alternate (block) minimization on D and A,

$$A^{(k+1)} = \arg\min_A \{f(D^{(k)} - A) + g(A)\} \quad (2)$$

$$D^{(k+1)} = \arg\min_D \{f(D - A^{(k+1)}) + g(A^{(k+1)})\}, \quad (3)$$

where $f(\cdot)$ and $g(\cdot)$ are *fitting* and *regularization* functions respectively. The first one is typically the squared ℓ_2 norm, $\|\cdot\|^2$, and the second one is an ℓ_p norm such as $\|\cdot\|_1$, with $0 \leq p \leq 2$ (when $p < 1$ it is not a norm).

[1]

Bibliography

[1] M. Aharon, M. Elad, and A. Bruckstein. k -svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, Nov 2006.

[2] Bart Kosko. Bidirectional associative memories. *IEEE Trans. Syst. Man Cybern.*, 18(1):49–60, January 1988.

[3] Mehmet Koyutrk and Ananth Grama. Proximus: A framework for analyzing very high dimensional discrete-attributed datasets. In *IN PROC. NINTH ACM SIGKDD INTL. CONF. ON KNOWLEDGE DISCOVERY AND DATA MINING (KDD 2003*, pages 147–156. IEEE Computer Society, 2003.