Online algorithms for Nonnegative Matrix Factorization with the Itakura-Saito divergence

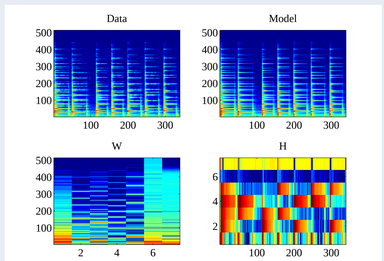
Augustin Lefèvre Francis Bach Cédric Févotte
INRIA (SIERRA project-team)/CNRS LTCI;Telecom ParisTech

Nonnegative matrix factorization

- Given an observed power spectrogram $V \in \mathbb{R}_+^{F\times N}$ and a dictionary of template spectra $W \in \mathbb{R}_+^{F\times K}$, Itakura-Saito nonnegative matrix factorization aims at solving :

$$\min_{\forall(f,k,n)\, H_{kn}\geq 0\ W_{fk}\geq 0} d_{IS}(V, WH)$$

- In Eq. (1), either W is a fixed dictionary (wavelets, gammatones, etc.) or it may be optimized to yield minimum error.
- Unsupervised learning of W provides meaningful data representations for simple signals ...



- ... but W may not be used in other tracks (poor generalization).
- Learning W on larger datasets may improve generalization power.

Bottlenecks in batch NMF

- Time : $O(FKN)$ (computation of $W \times H$).
- Memory :$O(FN + FK + KN)$(storing V, $W \times H$, W and H).
- As $N \to +\infty$, impossible to store V and $W \times H$ entirely.
- Stochastic gradient is unstable (gradient descent is not adapted to Itakura-Saito NMF).
- We propose to optimize a sequence of functions that converges to the objective function.

Idea

- Multiplicative updates are derived from an auxiliary function

$$d_{IS}(V, WH) \leq \sum_{f,k} A_{fk}^{(t)} \frac{1}{W_{fk}} + B_{fk}^{(t)} W_{fk}\,.$$

- $A_{fk}^{(t)} = (W_{fk}^{(t)})^2 \sum_{n=1}^N \frac{V_{fn}}{(W^{(t)}H^{(t)})_{fn}^2} H_{kn}^{(t)}$ $\quad B_{fk}^{(t)} = \sum_{n=1}^N \frac{1}{(W^{(t)}H^{(t)})_{fn}^2} H_{kn}^{(t)}$ in $O(FKN)$.
- Compute A_{fk},B_{fk} recursively in $O(FK)$.

Batch Algorithm (sketch)

Input training set of N samples, $W^{(0)}$, $A^{(0)}$, $B^{(0)}$.
$t \leftarrow 0$
for $t = 1 \ldots$ number of iterations
  for $n = 1 \ldots N$

$$h_n^{(t)} \leftarrow h_n^{(t-1)} \cdot \frac{W^\top \left(v_n \cdot (W h_n^{(t-1)})^{-2}\right)}{W^\top (W h_n^{(t-1)})^{-1}}$$

  end for

$$A^{(t)} \leftarrow \sum_{n=1}^N \left(\frac{v_n}{(W^{(t)}h_n)^2} h_n^\top\right) \cdot (W^{(t)})^2 \qquad B^{(t)} \leftarrow \sum_{n=1}^N \frac{1}{W^{(t)}h_n} h_n^\top$$

$$W^{(t+1)} \leftarrow \sqrt{\frac{A^{(t)}}{B^{(t)}}}$$

  end for
end for

Online Algorithm (sketch)

Input training set of N samples, $W^{(0)}$, $A^{(0)}$, $B^{(0)}$.
$t \leftarrow 0$
for *epoch* $= 1 \ldots$ number of iterations
  Randomly permute training set
  for $n = 1 \ldots N$
    $t \leftarrow t + 1$
    $h^{(t)} \leftarrow \arg\min_h d_{IS}(v_n, W^{(t)}h)$ (1)

$$a^{(t)} \leftarrow \left(\frac{v_n}{(W^{(t)}h_t)^2} h_t^\top\right) \cdot (W^{(t)})^2 \qquad b^{(t)} \leftarrow \frac{1}{W^{(t)}h_t} h_t^\top$$

$$A^{(t)} \leftarrow A^{(t-1)} + a^{(t)} \qquad B^{(t)} \leftarrow B^{(t-1)} + b^{(t)}$$

$$W^{(t+1)} \leftarrow \sqrt{\frac{A^{(t)}}{B^{(t)}}}$$

  end for
end for

Bibliography

S. Bucak and B. Gunsel. Incremental subspace learning via non-negative matrix factorization. *Patt. Recog.*, 42(5):788–797, 2009.

Matthew D. Hoffman, David M. Blei, and Francis Bach. Online learning for latent dirichlet allocation. In *Adv. in Neur. Inf. Proc. Sys. (NIPS)*, 2010.

J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *J. Mach. Learn. Res.*, 11:19–60, 2010.

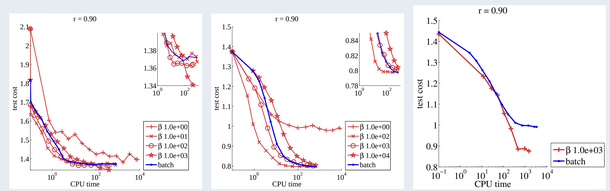
Main difference

In online NMF, the auxiliary function depends on all $(W^{(s)})_{1\leq s\leq t}$:
$A_{fk}^{(t)} = \sum_{s=1}^t \left(\frac{v_n}{(W^{(s)}h_s)^2} h_s^\top\right) \cdot (W^{(s)})^2$ $\quad B_{fk}^{(t)} = \sum_{s=1}^t \frac{1}{W^{(s)}h_s} h_s^\top$.

Practical issues

- Rescaling : rescale W, A, B at each update.
- Minibatch : Update W every β samples ($\beta = 10^3$).
- Minibatch : $\beta = 1$ is inefficient, $\beta = N$ requires too much memory.
- Early stopping in (1).
- Forgetting factor $A_{fk}^{(t)} = \rho A_{fk}^{(t-\beta)} + \sum_{s=t-\beta+1}^t a_{fk}^{(s)}$ $(\rho = 0.9)$ (idem $B^{(t)}$)
- Warning : unstable if $\beta = 1$ and $\rho < 1$.
- In practice, data sets are not infinite but may be too large to store : split data sets in several files so V is never loaded entirely.
- Warm restarts : warm restart h in (2) (at the cost of keeping H in memory).

How much faster ?



(a) 30 seconds' track (b) 4 minutes' track (c) 1 hour 20 minutes' track

Figure: Comparison of online and batch algorithm on short,medium and long audio tracks.

- Monitor convergence with a smaller test set : in practice descent of the test cost is observed.
- As long as memory is not an issue, storing H in memory (split in several files) and using warm restarts is preferrable.
- For large enough data sets, convergence of W is observed after a small number of passes through the data set.

Conclusion

- Our algorithm makes it possible to learn Itakura-Saito NMF on large data sets.
- It is possible to reformulate the same algorithm to deal with other β-divergences.
- Extension to sparse Itakura-Saito NMF is straightforward.