A SINGLE-CLASS SVM BASED ALGORITHM FOR COMPUTING AN IDENTIFIABLE NMF

Slim Essid

Institut Telecom / Telecom ParisTech, CNRS-LTCI - 37, rue Dareau - 75014 Paris, France

ABSTRACT

The geometric interpretation of Nonnegative Matrix Factorisation (NMF) as the problem of determining a convex cone that "well describes" the data under analysis has been key for addressing a major shortcoming of the "mainstream" NMF algorithms, that is the non-identifiability of the factorisation. On the basis of such geometric motivations, this paper proposes a novel algorithm that makes use of single-class support vector machines to recover the targeted NMF components. Not only does this new approach alleviate the NMF ill-posedness issue, but also it allows for automatically estimating the number of relevant NMF components, as demonstrated through experiments described in the paper. Moreover, it is readily kernelised thus opening the way for non-linear factorisations of the data.

Index Terms— nonnegative matrix factorisation, single-class support vector machines, identifiability.

1. INTRODUCTION

Nonnegative Matrix Factorisation (NMF) is a more and more popular "data decomposition" technique that has proven successful in various application domains, for instance audio and music processing [1, 2], audiovisual document structuring [3], or text mining, etc. By NMF, a set of n positive vector observations $\{v_1, \cdots, v_n\}$, with coefficients $v_{fi} \geq 0$, $(f, i) \in \{1, \cdots, F\} \times \{1, \cdots, n\}$, are "explained" as positive linear combinations of positive basis vectors (also called dictionary elements). This is accomplished by determining a low-rank approximation of the matrix V, assembled by stacking the observations column-wise: $V = (v_{fi})$, under the form $V \approx WH$; where $W = (w_{fk})$, with $w_{fk} \geq 0 \, \forall (f, k) \in \{1, \cdots, F\} \times \{1, \cdots, K\}$, is a rank-$K$ matrix whose columns w_k are the *basis vectors*; and where $H = (h_{ki})$ is a $K \times n$ matrix whose positive elements are the so-called *activation coefficients* or *regressors*.

W and H are usually obtained by minimizing a *cost function* $D(V|WH)$ (also referred to as *measure of fit*), that is generally a divergence [2], while imposing the positivity of W and H, which is approached as a constrained optimisation problem. This problem (which is non-convex in (W, H)) is classicly solved for using an iterative scheme whereby multiplicative update rules are alternately applied to iterates of W and H [4]. Though satisfactory solutions can thus be found (especially with a proper choice of measure of fit), these solutions are not unique: they highly depend on the initial iterates chosen. In fact, the NMF problem is known to be ill-posed as the factorization is not identifiable [5, 6].

As pointed out in the latter works, this ill-posedness issue is best understood through the geometric interpretation of the NMF problem as the one of determining a *simplicial convex cone* $\mathcal{C}_W = \left\{\sum_{k=1}^{K} \lambda_k w_k \ : \ \lambda_k \geq 0\right\}$, hence generated by the columns of W (called *vertices* of the cone), that contains the data points v_i. As can be seen in Figure 1, the identifiability problem is not confined to the indeterminacy of the basis-vectors scale, which is easily alleviated by requiring these vectors to be unit-length, *i.e.* $||w_k|| = 1$, as will be assumed hereafter. Rather, the main difficulty is that without any further constraints, there visibly often exists many possible cones containing the data points v_i.

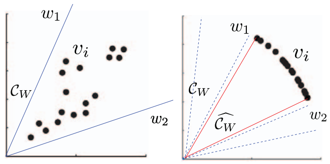

Fig. 1. Geometric interpretation of NMF in $\mathbb{R}^2$. Left panel depicts original data points, while right panel shows the same data after normalisation to unit-length. There exists many simplicial cones $\mathcal{C}_W$ containing the observations, the vertices w_k of which are given in blue dashed lines, the smallest is depicted in red continuous lines.

In order for the columns of W to be uniquely defined, Klingenberg *et al.* suggest to seek the smallest cone $\widehat{\mathcal{C}_W}$ containing the (normalised) data points v_i [6], where the expression "smallest cone" implicitly refers to the cone with minimum aperture (the aperture being defined as the maximum angle between two generators of the cone), which defines the *conic hull* of the data. This approach is shown to be successful in revealing the "true" data generation process, as soon as some observations are close enough to the vertices of the generating cone, which is likely to happen especially when the observations are sparse combinations of the underlying components. The smallest cone $\widehat{\mathcal{C}_W}$ is found in [6] by the *Extreme Vector Algorithm* (EVA) which (iteratively) seeks the cone whose generators (to be selected from V columns), are the "furthest away one from another in angular sense".

In this work, we propose an alternative new algorithm for computing an identifiable NMF, based on the single-class SVM technique, with numerous advantages over the EVA algorithm, notably that i) our proposal can be easily kernelised to allow for non-linear factorisations of the data matrix, and ii) the number of components K, which is often a critical parameter not know in advance, can be automatically determined from the data. This will become clear in Section 3 after the former technique is recalled in Section 2. Our approach is experimentally validated on two data configurations in Section 4, before we suggest conclusions and an outlook in Section 5.

2. SINGLE-CLASS SVM

Single-class support vector machines (SC-SVM) is a nonparametric density-support estimation technique that relies on a kernel-based

This work is supported by project ANR-09-JCJC-0073-01 TANGERINE (Theory and applications of nonnegative matrix factorization). The author would also like to thank Stéphan Clémençon and Cédric Févotte for fruitful discussions that have strongly inspired this work.

"maximum-margin style" set-up [7, chap. 8]. Here we briefly recall its principle and properties of interest for the computation of NMF.

Let $\mathcal{X} = \{x_1, \cdots, x_n\}$ be a set of n training vectors in an *input space* $\mathcal{X}$ (a subset of $\mathbb{R}^F$) and $\Phi : \mathcal{X} \to \mathcal{H}$ a map into a feature space $\mathcal{H}$ where the inner product is given through a kernel $\kappa(x, y) = \langle \Phi(x), \Phi(y) \rangle$, $(x, y) \in \mathbb{R}^F \times \mathbb{R}^F$. The single-class SVM technique consists in determining a function f_s, describing a hyperplane $\mathcal{P}$ in the feature space, whose sign is positive in a region, as small as possible, that captures most of the data. This is achieved by determining the hyperplane $\mathcal{P} : \langle a, \Phi(x) \rangle - \rho$, defined by the normal vector $a \in \mathcal{H}$, and the offset $\rho \geq 0$, that separates the feature vectors from the origin with maximum margin and letting f_s to be $f_s(x) = \text{sgn}\left(\langle a, \Phi(x) \rangle - \rho\right)$, with sgn the sign function, which indicates whether a feature vector is on the positive or negative side of the hyperplane.

To this end one may solve the following quadratic program: $\min_{a,\xi,\rho} \frac{1}{2}||a||^2 + \frac{1}{\nu n}\sum_i \xi_i - \rho$, s.t. $\langle a, \Phi(x_i) \rangle \geq \rho - \xi_i$, $\xi_i \geq 0$, $\rho \geq 0$, where ξ_i are slack variables introduced to account for outliers, $1 \leq i \leq n$, and ν is a positive penalization parameter used to allow for a trade-off, between margin maximisation and training errors, similarly to the usual bi-class ν-SVM set-up [7, chap. 8]. The solution is given by: $f_s(x) = \text{sgn}\left(\sum_{i=1}^n \alpha_i \kappa(x_i, x) - \rho\right)$, where α_i are Lagrange multipliers, verifying $0 \leq \alpha_i \leq \frac{1}{\nu n}$, most of which are zero. Vectors x_i with $0 < \alpha_i < \frac{1}{\nu n}$ are the *margin support vectors* (that lie on $\mathcal{P}$), those with $\alpha_i = \frac{1}{\nu n}$ are *non-margin support vectors* or outliers, and the remaining vectors of $\mathcal{X}$ have zero valued Lagrange multipliers.

Parameter ν is full of meaning and plays a central role. In fact, it is easily proven (using the KKT conditions) [7, app. A.2] that it is both an upper bound on the fraction of margin errors and a lower bound on the fraction of support vectors. Moreover, ν is asymptotically equal to both the fraction of outliers and the number of support vectors, under mild conditions on the form of the data distribution and the kernel [7, chap. 8].

Finally, it is worth mentioning that the above optimisation can be solved efficiently using one of the variants of the so-called *subset selection methods* [7, chap. 10], making it manageable on very large data sets.

Suppose, without loss of generality that the kernel $\kappa(x, y)$ is such that $\kappa(x, x) = ||\Phi(x)||^2 = 1, \forall x \in \mathbb{R}^F$. In fact, this is already true for radial kernels, that is kernels that only depend on $x - y$, and can be otherwise obtained, merely by transforming the original kernel $\kappa(x, y)$ into $\kappa'(x, y) = \frac{\kappa(x,y)}{\sqrt{\kappa(x,x)\kappa(y,y)}}, \forall (x, y) \in \mathbb{R}^F_* \times \mathbb{R}^F_*$. Consequently all the (non-zero norm) data points in the feature space lie on a unit hypersphere as shown in Figure 2.

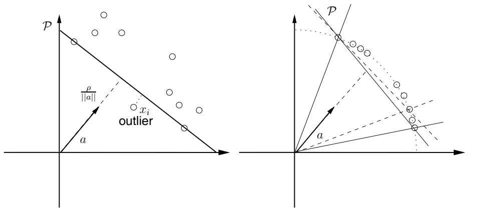

Fig. 2. Single-class support vector machine. Data points which lie on the hyperplane are the margin support vectors. With normalised kernels (right side) all points in the feature space are located on a unit hypersphere.

3. COMPUTING NMF USING SC-SVM

We start by describing the procedure to find the basis-vector matrix W before we explain how the weights in H can be recovered. Subsequently, we focus on a kernel-based version of our NMF algorithm.

3.1. Finding W

Following [6] we determine W by seeking the conic hull of the data, though using a completely different procedure. Our approach is motivated by the fact that since the observations are assumed to lie on a unit hypersphere, seeking the smallest cone containing the data is equivalent to looking for the hyperplane that separates the data from the origin with maximum margin, that is exactly the SC-SVM solution, as can be seen in Figure 2. In this Figure, two different cones are shown that correspond to two different SVM solutions, that is the hyperplanes shown respectively in dashed and continuous lines. The different solutions typically correspond to distinct ν values. Therefore, in order to obtain W, we apply the ν-SC-SVM algorithm on the data to select the margin support vectors as basis vectors w_k, since the former are then the vertices of $\widehat{\mathcal{C}_W}$. A nice consequence of this, is that the number of components K is thus automatically determined and is equal to the number of margin support vectors.

This approach has several advantages compared to the one proposed in [6] and more generally to "standard" NMF techniques, namely:

- the proposed algorithm can be straightforwardly kernelised (as will be shown hereafter), hence allowing one to achieve nonlinear factorisations of the data, and to incorporate in the kernel function prior knowledge on the data invariances and adequate similarity measures;
- the choice of K (the rank of W), often not known *a priori* but to be determined from the data, is not required, rather, one needs to choose a proper ν, that can be interpreted either as an upper-bound on the expected fraction of outliers in the set $\mathcal{V} = \{v_1, \cdots, v_n\}$ or a lower-bound on K;
- in marked contrast to the EVA algorithm whose time complexity is $O(K^4)$, our approach is computationally efficient thanks to the existence of various light-weight algorithms for solving the SVM optimisation problems, for instance variants of [7], that are readily available in many widely-used toolboxes, yielding complexities that can be as low as $O(n)$;
- the algorithm can be straightforwardly adapted to *online* processing, where the data is collected on the fly, building upon previous works on online SVM techniques;
- finally, our algorithm is applicable on any data configuration, as it does not require the restrictive assumption that the observations must satisfy the *extreme data property* [6], thanks to its ability to exclude some data points that can be considered as outliers.

3.2. Computing H

Once W has been determined, H can be found by solving a classic linear regression problem with positivity constraints. In the case where the rank of V is actually K, H can be found merely by solving the linear system $V = WH$. The h_i are then guaranteed to be positive for non-outlier data points (which lie inside the cone $\widehat{\mathcal{C}_W}$).

In general one seeks partial-rank factorisations ($K < F$) for which the solutions to $v_i = Wh_i$ are not unique, and not guaranteed to be positive. Therefore, we determine H by solving the nonnegative least-squares problem [8]: $\min_{h_i} C(h_i) = ||v_i - Wh_i||^2$, s.t. $h_{k,i} \geq 0$, $k \in \{1, \cdots, K\}$, for $1 \leq i \leq n$.

Solving for H in a regression setting is quite advantageous as it allows one to straightforwardly introduce further constraints on the solution. For instance more sparsity on H can be enforced by considering shrinkage methods, typically by introducing L1 penalties in the optimisation.

3.3. Kernel-based NMF

We now show how our NMF algorithm can be kernelised. Our proposal is not the first instance of kernel-NMF which has been previously introduced in a *convex NMF* setting [9]. However, the latter still has the same defects that other NMF algorithms have (particularly ill-posedness).

Kernel-based NMF seeks a positive factorisation of transformed observations: $\Phi(V) \approx WH$, with the constraint that W and H coordinates be nonnegative, where we re-use the notations of Section 2 taking $\mathcal{X} = \mathcal{V}$. The basis vectors w_k are here elements of the feature space $\mathcal{H}$, which are merely found in our approach using the kernel-based version of the single-class SVM, similarly to the linear case. Thus, W contains the Φ-transformed support vectors: $W = \left[\ \Phi(v_{\sigma_1}),\ \ldots\ ,\Phi(v_{\sigma_K})\ \right]$, where v_{σ_k} are the support vectors.

H is determined by solving the problem $\min_{h_i} C_\Phi(h_i) = \frac{1}{2}||\Phi(v_i) - Wh_i||^2_{\mathcal{H}}$, s.t. $h_{ki} \geq 0, k \in \{1, \cdots, K\}$, for $1 \leq i \leq n$. Using the kernel trick it can be easily shown that:

$$\frac{\partial C_\Phi(h_i)}{\partial h_{ki}} = \sum_{l=1}^{K} h_{li}\kappa(v_k, v_l) - \kappa(v_i, v_l) \qquad (1)$$

$$\frac{\partial^2 C_\Phi(h_i)}{\partial h_{li}\partial h_{ki}} = \kappa(v_k, v_l). \qquad (2)$$

Therefore, the Hessian matrix is exactly the Gram matrix which is positive definite for positive definite kernels. As a consequence, the previous optimisation problem is convex for any such kernels and can be solved using state-of-the-art optimisation techniques.

4. EXPERIMENTAL VALIDATION

In the following we validate the proposed algorithm by applying to simulated image and audio analysis problems. We start with an example inspired by one initially presented in [6].

4.1. An image analysis example

A set of gray-level images of size 32 × 32 is generated by linear combination of 3 component-images that are black circles on a very light background (almost white, but actually low-amplitude random noise). The components are depicted in Figure 3 together with an example of a synthesised image.

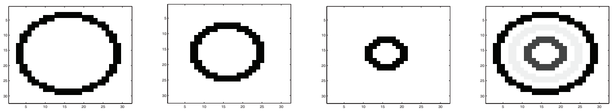

Fig. 3. Generating basis vectors with an observation example.

The images are represented as 1024-coefficient column vectors. The basis vectors are stacked in a 1024×3 matrix B. 1500 observations are formed in a 1024×1500 matrix V as $V = BC$, with coefficients of C drawn uniformly in the range $[0, 1]$.

The goal is to recover the generating "basis images" from the set of observations. To this end, single-class SVM based NMF, with $\nu = 0.001$, is applied on the data in two variations: i) using $V = [BC, B]$ and ii) using $V = BC$, to study the behavior of the algorithm when the targeted components are not among the observations.

The basis vectors found by the algorithm in each case are depicted in Figure 4. First, it is observed that when the basis vectors are among the data they are easily recovered by the algorithm. Second, when they are not necessarily among the data (*i.e.* when $V = BC$), the vectors selected by the algorithm are good approximations to them as can be seen in the second row of images of Figure 4, despite the fact that the activations (drawn uniformly) are not particularly sparse. If a simple post-processing were applied that would consist in "zeroing" all pixels whose values were below a low threshold, the result would be again perfect.

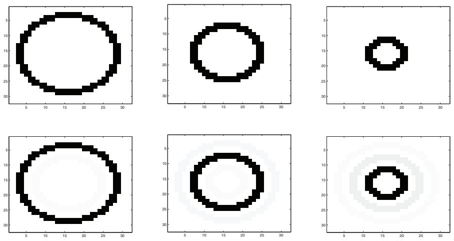

Fig. 4. Basis vectors found by ν-SC-SVM, with $\nu = 0.001$. Top row: using $V = [BC, B]$. Bottom row: $V = BC$.

Thus, this example is a good illustration of configurations where the fact that, in geometric approaches like ours, the "components" w_k are selected among the observations (given that they are support vectors), may not be too restrictive as soon as some data points are close to the basis vectors to be recovered. This is actually often verified for numerous applications, especially when the activations are sparse, or when the goal behind the use of NMF is to perform some form of (soft) clustering, which is typically the target of *convex NMF* approaches [9].

We now look at the tuning of the ν parameter. Figure 5 displays the behavior of the number of basis vectors selected (the margin support vectors SV) as a function of νn (recall that it is a lower bound on the number of SVs), compared to the total number of support vectors (including outliers). It is visible that the SC-SVM NMF algorithm always finds the appropriate number of components. When νn increases, outliers are created but the number of components remains fixed.

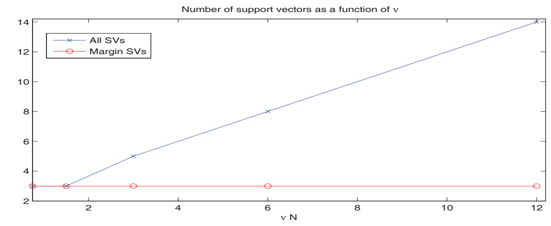

Fig. 5. Number of support vectors and basis vectors (that is margin support vectors) as a function of νn with $n = 1500$ observations.

4.2. An audio analysis example

We now turn to a synthetic audio example where we test the SV-NMF algorithm (that is our proposal) for a musical note transcription task. The synthetic audio piece to be analyzed (the spectrogram of which is given in Figure 6) is assembled from harmonic sinusoidal mixtures, each simulating a musical note m according to: $s_m(t) = \sum_{p=1}^{P} \frac{a_m}{p} cos(2\pi p f_m t)$, with $P = 10$ partials, f_m being the fundamental frequency of note m and a_m the amplitude of the first partial which is decreased linearly on the following ones. The excerpt is composed as follows: a C major chord is first built progressively by adding a new note every second, following the sequence C5, C5+E5, C5+E5+G5, C5+E5+G5+C6; then a much shorter single-note sequence (simulating an *arpeggio*) is added that consists of C5, E5, G5, C6, each played for 1/8 second, creating the situation where some observations are close to the targeted components. The signal to be analyzed is created according to $s(t) = \sum_m s_m(t) r_m(t) + b(t)$, where $r_m(t)$ is a rectangular window delimiting the activations of note m across time (see Figure 6), and $b(t)$ is a standard Gaussian noise whose power is calibrated to simulate a Signal to Noise Ratio (SNR) of 6dB. The amplitudes are chosen arbitrarily in the range $[0.5, 0.9]$ as show in Figure 6. The sampling frequency is 16kHz. The goal is to obtain a (non-supervised) transcription of this audio signal into notes using NMF. The task is not really straightforward due to the presence of noise and the fact that the partials of the chosen notes overlap in frequency (especially between C5 and C6).

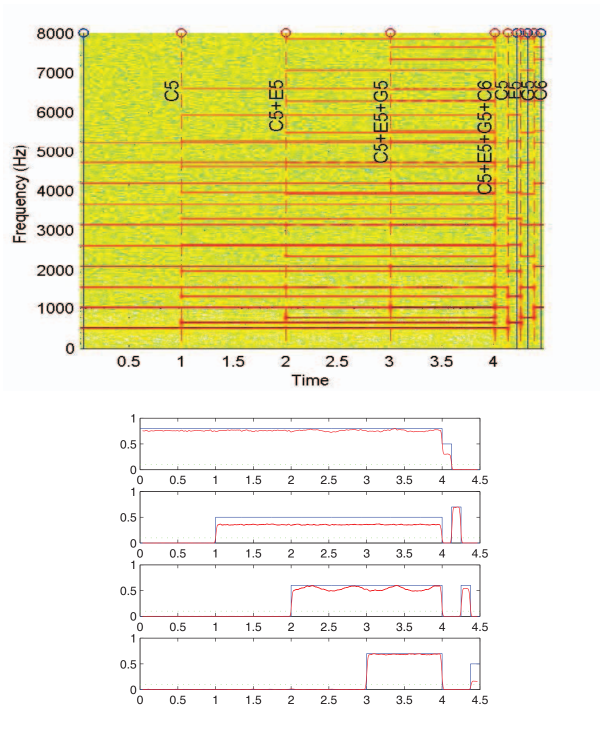

Fig. 6. *Top panel*: Synthetic audio signal. Red dashed vertical lines indicate ground-truth note onsets. Blue continuous vertical lines indicate the basis vectors that are selected by the algorithm. *Bottom panel*: Original note activations (a_m values) in blue, along with rescaled estimated activations in red. Decision threshold in green.

The power spectrogram of the signal is computed using 1024-sample length windows with a hopsize of 1000 samples and matrix V is formed by stacking the successive local observations of the power spectrum column-wise. SV-NMF is then executed on V with the parameter ν being fixed to an arbitrarily small values, here $\nu = 10^{-6}$ as no outliers are really expected to be observed.

The basis vectors found by the algorithm are highlighted in Figure 6 by continuous (blue) vertical lines (one nearby the origin of time, and three more at the end). As can be seen, SV-NMF has succeeded at automatically determining the relevant number of generating notes ($K = 4$). Additionally, the basis vectors selected are actually also relevant as they have been selected from segments where only one note is played at a time, and each one corresponds to a different note.

The estimated activations are shown in red in Figure 6 (after global rescaling to match the maximum amplitude of the original activations, for visual convenience). The automatic transcription of the signal is obtained merely by thresholding the obtained activations, which here presents no difficulty, yielding perfect transcription results.

5. CONCLUSIONS AND FUTURE WORK

A new geometric NMF algorithm, that makes use of the single-class SVM technique, has been introduced. The new algorithm combines a number of ideal features, notably that it is readily kernelised, it is able to automatically determine the number of relevant components, it is computationally efficient, and amenable to online processing scenarios.

Future work will consider real-world applications for this algorithm making use of its kernel-based version that has a great potential for incorporating prior knowledge on the data under analysis.

6. REFERENCES

[1] P. Smaragdis and J.C. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *Applications of Signal Processing to Audio and Acoustics, 2003 IEEE Workshop on.*, 2003, pp. 177 – 180.

[2] C. Fevotte, N. Bertin, and J.L. Durrieu, "Nonnegative matrix factorization with the Itakura-Saito divergence. with application to music analysis," *Neural Computation*, vol. 21, no. 3, Mar. 2009.

[3] S. Essid and C. Fevotte, "Nonnegative matrix factorization for unsupervised audiovisual document structuring," Tech. Rep. hal-00605886, HAL, 2011.

[4] Daniel D. Lee and H. Sebastian Seung, "Algorithms for non-negative matrix factorization," in *NIPS*, 2000, pp. 556–562.

[5] H. Laurberg, M. G Christensen, M. D Plumbley, L. K Hansen, and S. H Jensen, "Theorems on positive data: On the uniqueness of NMF," *Computational Intelligence and Neuroscience*, vol. 2008, pp. 1–9, 2008.

[6] Bradley Klingenberg, James Curry, and Anne Dougherty, "Non-negative matrix factorization: Ill-posedness and a geometric algorithm," *Pattern Recognition*, vol. 42, no. 5, pp. 918–928, May 2009.

[7] B. Shölkopf and A. J. Smola, *Learning with kernels*, The MIT Press, Cambridge, MA, 2002.

[8] C.L. Lawson and R.J. Hanson, *Solving least squares problems*, Classics in applied mathematics. SIAM, 1995.

[9] Chris H.Q. Ding, Tao Li, and Michael I. Jordan, "Convex and Semi-Nonnegative matrix factorizations," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 1, pp. 45–55, 2010.