BILEVEL SPARSE MODELS FOR POLYPHONIC MUSIC TRANSCRIPTION

Tal Ben Yakar,[1] Roee Litman,[1] Pablo Sprechmann,[2] Alex Bronstein,[1] and Guillermo Sapiro[2]
[1]Tel Aviv University, [2]Duke University
talby10@gmail.com, roeelitm@post.tau.ac.il, bron@eng.tau.ac.il
{pablo.sprechmann,guillermo.sapiro}@duke.edu

ABSTRACT

In this work, we propose a trainable sparse model for automatic polyphonic music transcription, which incorporates several successful approaches into a unified optimization framework. Our model combines unsupervised synthesis models similar to latent component analysis and nonnegative factorization with metric learning techniques that allow supervised discriminative learning. We develop efficient stochastic gradient training schemes allowing unsupervised, semi-, and fully supervised training of the model as well its adaptation to test data. We show efficient fixed complexity and latency approximation that can replace iterative minimization algorithms in time-critical applications. Experimental evaluation on synthetic and real data shows promising initial results.

1. INTRODUCTION

The goal of automatic music transcription (AMT) is to obtain a musical score from an input audio signal. AMT is particularly difficult when the audio signal is polyphonic [12], as the harmonic relations and interactions in music signals challenges the detection of multiple concurrent pitches. Polyphonic AMT is still considered an open problem and the state-of-the-art solutions are far from the level of precision required in many applications. We refer the reader to [4] for a detailed description of the open questions and challenges in polyphonic AMT.

Work partially supported by BSF, ONR, NGA, NSF, ARO, and AFOSR.

1.1 Prior work

In what follows, we briefly review two main families of approaches recently used for AMT, which are particularly relevant for the present work. Reviewing all existing AMT methods is beyond the scope of this paper; for recent surveys, we refer to the reader to [4, 12] and references therein.

Being essentially a classification task, music transcription has been addressed by classification-based approaches. These techniques define a set of meaningful features for pitch and onset detection, and feed them to generic classification schemes such as neural networks [6, 15], deep believe networks [16], and support vector machines [17]. In [16], the features themselves were learned form the data. In [17], the authors argue that prior knowledge (such as harmonicity) is not strictly necessary for achieving levels of transcription accuracy comparable to the one obtained with competing approaches, and that such assumptions can be substituted with discriminative learning. While being feasible, the lack of insight makes such pure learning-based systems hard to train, since they need to infer from the training data all possible variations and combinations of pitches. In general, this translates into long off-line training times and requires huge training sets.

Another family of recent approaches is based on spectrogram factorization techniques, such as non-negative matrix factorization (NMF) [13], and its probabilistic counterpart – probabilistic latent component analysis (PLCA) [20]. The basic idea, first introduced in [19], aims at factorizing a spectral representation of the signal $\mathbf{X} \in \mathbb{R}^{n\times k}$, into a product of non-negative factors, $\mathbf{X} \approx \mathbf{DZ}$, where the $n \times p$ non-negative dictionary $\mathbf{D}$ contains templates of the individual pitches, and the $p \times k$ non-negative factor $\mathbf{Z}$ contains the corresponding activations for each frame. Ideally one would expect $\mathbf{Z}$ to resemble a piano-roll and reveal the active notes in each spectral frame. Unfortunately, this is not enough in practice. In order to overcome this problem, many approaches have proposed to regularize the factorization by including

sparsity [1], harmonicity, and smoothness [5, 22]. In [2], the authors propose a shift-invariant version of PLCA, where the dictionary contains note templates from multiple orchestral instruments. Including such a regularizations usually significantly improves the results but also translates into slower coding schemes. This contrasts with the discriminative approaches that, after training, have very light computational costs. On one hand, the generative nature of factorization approaches allows them to handle the spectral superposition of harmonically related pitches in a natural way. On the other hand, however, such generative approaches seem less flexible and more difficult to adapt to specific settings compared to their discriminative counterparts.

1.2 Contributions

In this paper, we present an attempt to inject the generative properties of factorization approaches into a discriminative setting. We aim at establishing a bridge between pure learning and factorization-based methods. Specifically, we propose the coupled training of a set of classifiers that detect the presence of a given pitch in a frame of audio, taking as the input the activations produced by a generative matrix factorization scheme. Instead of constraining the factorization algorithm, we design a very simple factorization method trained to produce the optimal input to a binary classifier in the sense of the classification performance. Once trained, the simplicity of the proposed factorization scheme allows to use fast approximation of sparse encoders, resulting in computation complexity comparable to that of pure discriminative models. With this implementation, the proposed method bridges between factorization and classification methods, designing a neural network that solves a meaningful factorization problem.

We formulate our model as a bilevel optimization program, generalizing supervised dictionary learning devised for sparse coding schemes [14]. We also incorporate elements of metric learning into this supervised sparse NMF setting in order to increase its discriminative power. The proposed approach is naturally amenable to semi-supervised training regimes, in which unlabeled data can be used to adapt the system. Finally, the output of these classifiers is temporally smoothed as is normally done in AMT [2, 17].

In Section 2 we present the proposed model coupling the codes with the pitch classifiers. Then, in Section 3, we formulate its supervised variant as a bilevel optimization problem. In Section 4, we describe how the proposed method can be significantly accelerated. Section 5 shows how to incorporate the proposed scheme into higher-level temporal models. Experimental evaluation is reported in Section 6. Finally, Section 7 concludes the paper.

2. NON-NEGATIVE SPARSE MODEL

Like the majority of music and speech analysis techniques, music transcription typically operates on the magnitude of the audio time-frequency representation such as the short-time Fourier transform or constant-Q transform (CQT) [8], as adopted in this work. Given a spectral frame $\mathbf{x} \in \mathbb{R}^n_+$ at some time, the transcription problem consists of producing a binary label vector $\mathbf{y} \in \{-1, +1\}^p$, whose i-th element indicates the presence $(+1)$ or absence (-1) of the i-th pitch at that time. We use $p = 88$ corresponding to the span of the standard piano keyboard (MIDI pitches $21 - 108$).

In the proposed model, the output vector is produced by applying a simple linear classifier $\mathbf{y} = \operatorname{sign}(\mathbf{W}\mathbf{z} + \mathbf{a})$, parametrized by the $p \times m$ matrix $\mathbf{W}$ and $p \times 1$ vector $\mathbf{a}$, to the m-dimensional feature vector $\mathbf{z}$ obtained by solving the following non-negative sparse representation pursuit problem

$$\mathbf{z}(\mathbf{x}) = \arg\min_{\mathbf{z} \geq \mathbf{0}} \frac{1}{2}\|\mathbf{M}(\mathbf{x} - \mathbf{D}\mathbf{z})\|_2^2 + \lambda_1\|\mathbf{z}\|_1 + \lambda_2\|\mathbf{z}\|_2^2. \quad (1)$$

Here, $\mathbf{D}$ is an $n \times m$ over-complete ($m > n$) non-negative dictionary, whose columns represent different templates for each of the individual pitches, and $\mathbf{M}$ is a $r \times n$ metric matrix ($r \leq n$).

The first data fitting term requires the data to be well-approximated by a sparse non-negative combination of the atoms of $\mathbf{D}$, expressing the assumption that at each time, only a few pitches are simultaneously present. Replacing the standard Euclidean fitting term by a more general Mahalanobis metric parametrized by the matrix $\mathbf{M}$ allows to give different weights to different frequencies, as frequently practiced in music processing. The second term, whose relative importance is governed by the parameter λ_1, actually promotes the sparsity of the solution vector, while the third term is added for regularization.

Pursuit problem (1) is a strictly convex optimization problem, which can be solved efficiently using (among other alternatives) a family of optimization techniques called proximal methods. We adopt a non-negative variant of the iterative shrinkage-thresholding algorithm (ISTA) [9], summarized in Algorithm 1. While faster versions of this fixed-step proximal method can be used to reach linear convergence rates, the discussion of these extensions is beyond of the scope of this paper.

We observe that given a collection of spectral frames $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_k)$, the solution of the pursuit problem aims at finding a non-negative factoriza-

input : Data $\mathbf{x}$, dictionary $\mathbf{D}$, metric matrix $\mathbf{M}$, parameters λ_1, λ_2, step size α.
output: Non-negative sparse code $\mathbf{z}$.
Define $\mathbf{H} = (1 - \frac{\lambda_2}{\alpha})\mathbf{I} - \frac{1}{\alpha}\mathbf{M}^{\mathrm{T}}\mathbf{D}^{\mathrm{T}}\mathbf{D}\mathbf{M}$,
$\mathbf{G} = \frac{1}{\alpha}\mathbf{M}^{\mathrm{T}}\mathbf{D}^{\mathrm{T}}$, $\mathbf{t} = \frac{1}{\alpha}\lambda_1$.
Initialize $\mathbf{z}^1 = \mathbf{0}$ and $\mathbf{b}^1 = \mathbf{G}\mathbf{x}$.
for $k = 1, 2, \ldots$ *until convergence* **do**
  $\mathbf{z}^{k+1} = \sigma_{\mathbf{t}}(\mathbf{b}^k)$
  $\mathbf{b}^{k+1} = \mathbf{b}^k + \mathbf{H}(\mathbf{z}^{k+1} - \mathbf{z}^k)$
end

Algorithm 1: Non-negative iterative shrinkage-thresholding algorithm (ISTA). $\sigma_{\mathbf{t}}(\mathbf{b}) = \max\{\mathbf{0}, \mathbf{b} - \mathbf{t}\}$ denotes element-wise single-sided soft thresholding.

tion of $\mathbf{X}$ into $\mathbf{DZ}$, thus being essentially an instance of a non-negative matrix factorization (NMF) problem with a fixed left factor $\mathbf{D}$. The approach proposed in the paper can be essentially viewed as a supervised version of NMF.

Denoting the parameters of the sparse model as $\mathbf{\Theta} = \{\mathbf{D}, \mathbf{M}\}$, and those of the linear classifier as $\mathbf{\Phi} = \{\mathbf{W}, \mathbf{a}\}$, the proposed pitch transcription system can be expressed as $\mathbf{y} = \mathbf{y}_{\mathbf{\Phi}}(\mathbf{z}_{\mathbf{\Theta}}(\mathbf{x}))$, where $\mathbf{z}_{\mathbf{\Theta}}$ denotes the non-linear map produced by solving (1), and $\mathbf{y}_{\mathbf{\Phi}}$ refers to the application of the classifier. In what follows, we will address how to train and adapt the parameters $\mathbf{\Theta}$ and $\mathbf{\Phi}$ for the AMT task.

Dictionary initialization. The initial dictionary is constructed to contain spectral templates for each possible pitch. The training of the dictionary is done by learning a set of small sub-dictionaries, one per pitch, minimizing

$$\min_{\mathbf{D}\in\mathcal{D},\mathbf{Z}\geq\mathbf{0}} \frac{1}{2}\|\mathbf{M}(\mathbf{X} - \mathbf{DZ})\|_{\mathrm{F}}^2 + \lambda_1\|\mathbf{Z}\|_1 + \lambda_2\|\mathbf{Z}\|_{\mathrm{F}}^2, \quad (2)$$

where $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm, $\mathcal{D}$ is the space of appropriately sized non-negative matrices with unit columns, and $\mathbf{M} = \mathbf{I}$. Additional constrains such as harmonicity can be included by changing $\mathcal{D}$ to be more restrictive. The initial dictionary can be constructed for a specific instrument or for multiple instruments as in [2], via simple concatenation.

Classifier initialization. Once the initial dictionary has been trained, we can learn the classifier parameters $\mathbf{\Phi}$. To that end, we construct a training set $\mathcal{X}$ containing pairs of the form $(\mathbf{x}, \mathbf{y})$, of spectral frames with the corresponding groundtruth pitch labels. Here, unlike in the unsupervised dictionary training, the best performance of the classifier is obtained when the training set contains representative examples of chords and pitch combinations.

The classifier is trained by minimizing

$$\min_{\mathbf{\Phi}} \frac{1}{|\mathcal{X}|} \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{X}} \ell(\mathbf{y}_{\mathbf{\Phi}}(\mathbf{z}), \mathbf{y}) \quad (3)$$

on the outputs $\mathbf{z} = \mathbf{z}_{\mathbf{\Theta}}(\mathbf{x})$ of the pursuit algorithm. Here, ℓ denotes a loss function penalizing for the mismatch between the ground truth labels and the actual output of the classifier. We use the logistic regression loss function $\ell(\mathbf{y}', \mathbf{y}) = \log(1 + e^{-\mathbf{y}^{\mathrm{T}}\mathbf{y}'})$. The minimization of (3) can scale to very large training sets by using (projected) stochastic gradient descent (SGD) techniques [7], which we adopt in all our experiments.

3. BILEVEL SPARSE MODEL

A striking disadvantage of the two-stage training described so far is the fact that the training of the dictionary $\mathbf{D}$ aims at reducing the data reconstruction error $\|\mathbf{X} - \mathbf{DZ}\|_{\mathrm{F}}$ rather than reducing the classification error (3). Consequently, the dictionary trained in the initial unsupervised regime is suboptimal in the sense of (3); furthermore, there is no natural way to train the metric matrix $\mathbf{M}$. The ultimate way to perform supervised training of the entire system would be therefore by minimizing

$$\min_{\mathbf{\Theta},\mathbf{\Phi}} \frac{1}{|\mathcal{X}|} \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{X}} \ell(\mathbf{y}_{\mathbf{\Phi}}(\mathbf{z}_{\mathbf{\Theta}}(\mathbf{x})), \mathbf{y}) + \mu||\mathbf{\Phi}||_{\mathrm{F}}^2 \quad (4)$$

not only with respect to the parameters $\mathbf{\Phi}$ of the classifier, but also with respect to the parameters $\mathbf{\Theta}$ of the pursuit. This leads to a *bilevel* optimization problem, as we need to optimize the loss function ℓ, which in turn depends on the minimizer of (1). Note that, as is standard practice in machine learning, a regularization term on the classifier parameters is added to prevent over-fitting.

In particular, one would need to compute the gradients of the loss with respect to the parameters $\mathbf{\Theta} = \{\mathbf{D}, \mathbf{M}\}$ of the pursuit. Fortunately, $\mathbf{z}$ is almost everywhere differentiable with respect to $\mathbf{D}$ and $\mathbf{M}$ [14]. Denoting by Λ the active set of $\mathbf{z}$ (i.e., the set of indices at which it attains non-zero values), we define

$$\boldsymbol{\beta}_{\Lambda} = (\mathbf{D}_{\Lambda}^{\mathrm{T}}\mathbf{M}^{\mathrm{T}}\mathbf{M}\mathbf{D}_{\Lambda} + \lambda_2\mathbf{I}_{\Lambda})^{-1}(\nabla_{\mathbf{z}}\ell(\mathbf{y}_{\mathbf{\Phi}}(\mathbf{z}), \mathbf{y}))_{\Lambda},$$

where $\nabla_{\mathbf{z}}\ell$ is the gradient of the loss function with respect to $\mathbf{z}$. The elements of $\boldsymbol{\beta}$ outside Λ are set to zero. The gradients of $\ell(\mathbf{y}_{\mathbf{\Phi}}(\mathbf{z}), \mathbf{y})$ with respect to $\mathbf{D}$ and $\mathbf{M}$ can be expressed as

$$\begin{aligned} \nabla_{\mathbf{D}}\ell &= \mathbf{M}^{\mathrm{T}}\mathbf{M}((\mathbf{x} - \mathbf{D}\mathbf{z}^*)\boldsymbol{\beta}^{\mathrm{T}} - \mathbf{D}\boldsymbol{\beta}\mathbf{z}^*) \quad (5)\\ \nabla_{\mathbf{M}}\ell &= \mathbf{M}\mathbf{D}_{\Lambda}\boldsymbol{\beta}_{\Lambda}(\mathbf{x} - \mathbf{D}\mathbf{z}^*)^{\mathrm{T}} - \mathbf{M}(\mathbf{x} - \mathbf{D}\mathbf{z}^*)\mathbf{D}_{\Lambda}^{\mathrm{T}}. \end{aligned}$$

We omit the derivation details due to lack of space, and refer the reader to [14] for a related discussion.

We perform the minimization of (4) again by using SGD alternating descents on $\mathbf{\Phi}$ keeping $\mathbf{\Theta}$ fixed, and on $\mathbf{\Theta}$ keeping $\mathbf{\Phi}$ fixed. $\mathbf{\Theta}$ and $\mathbf{\Phi}$ are initialized as described in Section 2. It is also worthwhile noting that the minimization of the discriminative loss (4) with respect to the matrix $\mathbf{M}$ can be viewed as a particular setting of *metric learning* – a family of problems that aim at designing task-specific metrics. In our case, we design a Mahalanobis metric $\mathbf{M}^{\mathrm{T}}\mathbf{M}$ such that the pursuit with respect to it minimizes the classification errors.

The purely discriminative objective of (4) is susceptible to over-fitting since the learned matrix $\mathbf{M}$ will not aim at producing faithful data reconstructions. In that case, the generative advantage of NMF would be lost. To avoid this problem, the minimization of (4) can be regularized by adding an data reconstruction term of the form $\|\mathbf{x} - \mathbf{D}\mathbf{z}_{\mathbf{\Theta}}(\mathbf{x})\|_2^2$.

We distinguish between two training regimes: in the *fully supervised* setting, all samples in the training set come with label information $\mathbf{y}$, and the training is performed as described above. Since label information is often difficult to obtain, in many practical cases only some of the samples in the training set are labeled. We call such a setting *semi-supervised*. Given a set of unlabeled data, $\mathcal{X}_u$, we can change the learning process by augmenting the discriminative loss (4) on the labeled data (eventually regularized with data reconstruction term), with the unsupervised term

$$\sum_{\mathbf{x}\in\mathcal{X}_u} \frac{1}{2}\|\mathbf{x}-\mathbf{D}\mathbf{z}_{\mathbf{\Theta}}(\mathbf{x})\|_2^2+\lambda_1\|\mathbf{z}_{\mathbf{\Theta}}(\mathbf{x})\|_1+\lambda_2\|\mathbf{z}_{\mathbf{\Theta}}(\mathbf{x})\|_2^2.$$

This new training scheme aims at producing a dictionary $\mathbf{D}$ that is good for classifying (and reconstructing) the labeled data but that can also used to sparsely represent the unlabeled data. Note that this regime can also be used as a way to *adapting* the system to unseen testing data. Both factorization- and classification-based approaches suffer a performance drop when the testing data are not well represented by the training samples. In this way, our system can be adapted to new unseen (and unlabeled) data.

4. FAST APPROXIMATION

The proposed approach relies on solving optimization problem (1) using an iterative method. One of the drawbacks of such iterative schemes is their relatively high computational complexity and latency, which is furthermore data-dependent. For example, non-negative ISTA typically requires hundreds of iterations to converge. However, while the classical optimization theory provides worst-case (data-independent) convergence rate bounds for many families of iterative algorithms, very little is known about their behavior on *specific* data, coming e.g., from a distribution supported on a low-dimensional manifold – properties often exhibited by real data. Common practice of sparse modeling concentrates on creating sophisticated data models, and then relies on computational and analytic techniques that are totally agnostic of the data structure.

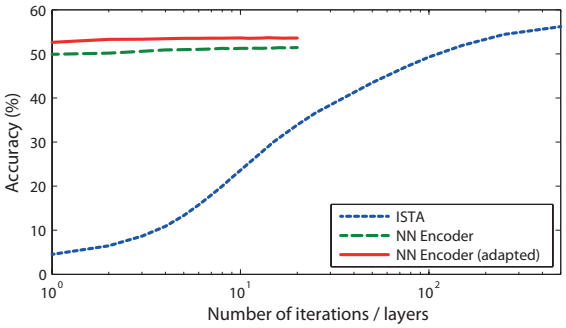

Figure 1. Accuracy of the optimization-based and neural network encoders as a function of the number of iterations or layers. Evaluation was performed on dataset from [17]. The networks were trained in the unsupervised regime.

From the perspective of the pursuit process, the minimization of (1) is merely a proxy to obtaining a highly non-linear map between the data vector $\mathbf{x}$ and the corresponding feature vector $\mathbf{z}$. Adopting ISTA as the iterative algorithm, such a map can be expressed by unrolling the iterations into the composition $f \circ f \circ \cdots \circ f(\mathbf{0}, \mathbf{G}\mathbf{x})$ of T elementary operations of the form $f : (\mathbf{z}, \mathbf{b}) \mapsto (\sigma_{\mathbf{t}}(\mathbf{b}), \mathbf{b} + \mathbf{H}(\sigma_{\mathbf{t}}(\mathbf{b}) - \mathbf{b}))$, where T is a sufficiently large number of iterations required for convergence. By fixing T, we obtain a fixed-complexity and latency encoder $\hat{\mathbf{z}}_{T,\mathbf{\Psi}}(\mathbf{x})$, parametrized by $\mathbf{\Psi} = \{\mathbf{H}, \mathbf{G}, \mathbf{t}\}$ (recall that ISTA defines the latter parameters as functions of $\mathbf{\Theta} = \{\mathbf{D}, \mathbf{M}\}$). Such an encoder can be thought of as a time-recurrent neural network, or a feed-forward network with T identical layers.

Note that for a sufficiently large T, $\hat{\mathbf{z}}_{T,\mathbf{\Psi}} \approx \mathbf{z}_{\mathbf{\Theta}}$. However, when complexity budget constraints require T to be truncated at a small fixed number, the output of $\hat{\mathbf{z}}_{T,\mathbf{\Psi}}$ is usually unsatisfactory, and the worst-case bounds provided by the classical optimization theory are of little use. However, within the family of functions $\{\hat{\mathbf{z}}_{T,\mathbf{\Psi}}\}$, there might exist better parameters for which $\hat{\mathbf{z}}$ performs better *on relevant input data*. These ideas advocated by [11], have been recently shown very effective in sound separation problems [21].

Adapted to our problem, the encoder $\hat{\mathbf{z}}_{T,\mathbf{\Psi}}$ can be trained in lieu of the iterative pursuit process in one of the discussed regimes, using the standard backpropagation techniques to compute the gradients of the network with respect to its parameters [11]. The training of the encoder can be achieved by minimize the dis-

criminative objective

$$\min_{\mathbf{\Psi},\mathbf{\Phi}} \frac{1}{|\mathcal{X}|} \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{X}} \ell(\mathbf{y}_{\mathbf{\Phi}}(\hat{\mathbf{z}}_{T,\mathbf{\Psi}}(\mathbf{x})),\mathbf{y}) + \mu||\mathbf{\Phi}||_{\mathrm{F}}^2 \quad (6)$$

Similar ideas can be used in the unsupervised setting. Figure 1 shows, as a function of T, the performance of the exact pursuit (ISTA truncated after T iterations), and its approximation using the neural networks. About two orders of magnitude of speedup is observed.

This perspective bridges between two popular approaches to music transcription: those based on explicit data modeling and relying on optimization to solve some kind of a representation pursuit problem, and those relying on pure learning of a neural network. However, while training $\hat{\mathbf{z}}_{T,\mathbf{\Psi}}$ is technically a pure learning approach, it is very much rooted into the underlying data model. First and most importantly, the training objective is solving a matrix factorization problem. Second, the architecture of the neural network is derived from an iterative process that is guaranteed to minimize a meaningful objective. Third, the network comes with a very good initialization of the parameters (as prescribed by ISTA). Since neural network training is a highly non-convex optimization problem, such an initialization is crucial.

5. TEMPORAL REGULARIZATION

Independent analysis of spectral frames fails short of exploiting the temporal structures and dependencies of music signals. This prior knowledge can be incorporated by temporally regularizing the output of the classifiers. A popular way to achieve this is by adding a post-processing stage based on hidden Markov models (HMMs) [18]. Following [2, 17], in this work we smooth the classifier outputs using an independent two-state HMM for each pitch. We now think of the output of the sparse coding $\mathbf{Z} = (\mathbf{z}_1, \ldots, \mathbf{z}_k)$ as a sequence of k input observations. For each pitch p, the states are represented as a sequence of hidden variables $\mathbf{q}^p = (q_1^p, \ldots, q_k^p)$ that take the value of $+1$ in the presence, or -1 in the absence of the pitch, following the convention used throughout the paper. The HMM aims at finding for each pitch, the optimal sequence of states minimizing

$$\min_{\mathbf{q}^p} \; p(\mathbf{z}_1|\mathbf{q}_1^p)p(\mathbf{q}_1^p)\prod_{i=2}^{k} p(\mathbf{z}_i|\mathbf{q}_i^p)p(\mathbf{q}_i^p|\mathbf{q}_{i-1}^p), \quad (7)$$

where the initial probabilities $p(\mathbf{q}_1^p)$ and the transition probabilities $p(\mathbf{q}_i^p|\mathbf{q}_{i-1}^p)$ are learned from the data. The probability of observing a sparse code given a pitch state, $p(\mathbf{z}_i|\mathbf{q}_i^p)$, can be obtained naturally from the output of the classifiers. The logistic classifiers can be thought as generalized linear predictors for Bernoulli variables, leading to $p(\mathbf{z}_i|\mathbf{q}_i^p = y) = 1/(1+e^{-y\mathbf{w}^{\mathrm{T}}\mathbf{z}_i})$. Hence, maximizing the classifier's performance can be thought as maximizing the likelihood of the true state in the HMM. Problem (7) is solved using the Viterbi algorithm [18].

Table 1. Precision, recall, F1 and accuracy in percent on the test data presented in [17] for the proposed approach under different training regimes. For reference, the accuracy obtained for three recent alternative methods is: 57.6 % for [2], 56.5 % for [17], and 47.0 % for [3].

Training regime	Pre.	Rec.	F1	Acc.
Supervised	81.6	69.8	74.3	60.0
Supervised+Fitting	79.7	69.9	73.7	59.2
Semi-supervised	79.7	70.0	73.7	59.2
Semi-supervised+Fitting	82.0	70.9	75.1	61.0

6. EXPERIMENTAL EVALUATION

Similarly to the factorization-based methods, the proposed model can be trained to transcribe pieces containing mixtures of instruments by appropriately training the initial dictionaries. However, since the scope of the paper is rather a proof-of-concept than the design of a full-featured AMT system, we limit the experimental evaluation to piano recordings only.

Data. The system was tested on the Disklavier dataset proposed in [17]. For training of the initial dictionaries (pitch templates), two different piano types were used from the MAPS dataset [10]. Then, the classifier and the dictionary were trained in the supervised regime using the first 30 seconds of 50 songs from MAPS and the training data in [17]. Spectral frames were represented using CQT with 48 frequency bins per octave.

Performance measures. We adopted the frame-based accuracy measure proposed in [17], $\mathrm{Acc} = \mathrm{TP}/(\mathrm{FP}+\mathrm{FN}+\mathrm{TP})$, where TP (true positives) is the number of correctly predicted pitches, and FP (false positives) and FN (false negatives) are the number of pitches incorrectly transcribed as ON or OFF, respectively. We also include the standard frame-based precision, recall and F1 measures.

Evaluation. The proposed system was evaluated under different training regimes. Training the system by solving the bilevel optimization problem (4) is referred to as *Supervised*, and *Supervised+Fitting* when using the additional data fitting term described in Section 3. We also tested the capability of our system to use unlabeled data to unsupervisedly adapt to the test data (the *Semi-supervised+Fitting* settings). The obtained performance was compared against one successful representative approach from each of the

main existing philosophies. We used [2] to represent the factorization-based approaches, and [17] for the classification-based ones. Both of these methods include a post-processing stage very similar to the one used in the work. We also included a third method based on onset detection [3]. It is worth mentioning that the training of [2] does not use the Disklavier training data given in [17]; the authors train their system using samples of three different piano types of the MAPS dataset.

Table 1 summarizes the obtained results. The proposed method is competitive with the alternative approaches. The inclusion of unlabeled data allows a significant improvement in the system performance. Adding the fitting term seems to have more impact when unlabeled data are available. We attribute this to the reduction of over-fitting risks that such a regularization offers. In both semi- and fully-supervised regimes, the reconstruction properties of the dictionary are much better preserved with the mentioned fitting regularization.

7. CONCLUSION

We showed a trainable bilevel non-negative sparse model model for polyphonic music transcription. Our model can be interpreted as a supervised variant of NMF as well as a flavor of metric learning. We also showed that the original iterative optimization-based approach can be efficiently approximated by fixed-complexity feed-forward architectures that give a two-order-of-magnitude speedup at little expense of accuracy. This creates an interesting relation between the optimization-based transcription methods, and those relying on pure learning, which are traditionally dealt with by two, practically disjoint, communities. The approach can naturally benefit from the inclusion of unlabeled data via a semi-supervised training scheme.

8. REFERENCES

[1] S. Abdallah and M. Plumbley. Polyphonic music transcription by non-negative sparse coding of power spectra. In *ISMIR*, pages 10–14, 2004.

[2] E. Benetos and S. Dixon. Multiple-instrument polyphonic music transcription using a convolutive probabilistic model. In *Sound and Music Computing Conference*, pages 19–24, 2011.

[3] E. Benetos and S. Dixon. Polyphonic music transcription using note onset and offset detection. In *ICASSP*, pages 37–40. IEEE, 2011.

[4] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri. Automatic music transcription: Breaking the glass ceiling. In *ISMIR*, 2012.

[5] N. Bertin, R. Badeau, and E. Vincent. Enforcing harmonicity and smoothness in Bayesian non-negative matrix factorization applied to polyphonic music transcription. *IEEE Trans. Audio, Speech, and Language Proc.*, 18(3):538–549, 2010.

[6] S. Bock and M. Schedl. Polyphonic piano note transcription with recurrent neural networks. In *ICASSP*, pages 121–124, 2012.

[7] L. Bottou. Large-scale machine learning with stochastic gradient descent. In *COMPSTAT*, pages 177–187, August 2010.

[8] J. C. Brown. Calculation of a constant Q spectral transform. *The Journal of the Acoustical Society of America*, 89:425, 1991.

[9] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57(11):1413–1457, 2004.

[10] V. Emiya, R. Badeau, and B. David. Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle. *IEEE Trans. Audio, Speech, and Language Proc.*, 18(6):1643–1654, 2010.

[11] K. Gregor and Y. Lecun. Learning fast approximations of sparse coding. In *ICML*, 2010.

[12] A. Klapuri. Multipitch analysis of polyphonic music and speech signals using an auditory model. *IEEE Trans. Audio, Speech, and Language Proc.*, 16(2):255–266, 2008.

[13] D. D. Lee and H. S. Seung. Learning parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[14] J. Mairal, F. Bach, and J. Ponce. Task-driven dictionary learning. *IEEE Trans. PAMI*, 34(4):791–804, 2012.

[15] M. Marolt. A connectionist approach to automatic transcription of polyphonic piano music. *IEEE Trans. Multimedia*, 6(3):439–449, 2004.

[16] J. Nam, J. Ngiam, H. Lee, and M. Slaney. A classification-based polyphonic piano transcription approach using learned feature representations. In *ISMIR*, 2011.

[17] G. E. Poliner and D. Ellis. A discriminative model for polyphonic piano transcription. *EURASIP J. Adv. in Sig. Proc.*, 2007, 2006.

[18] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, 77(2):257, 1989.

[19] P. Smaragdis and J. Brown. Non-negative matrix factorization for polyphonic music transcription. In *WASPAA*, pages 177–180, 2003.

[20] P. Smaragdis, B. Raj, and M. Shashanka. A probabilistic latent variable model for acoustic modeling. In *NIPS*, volume 148, 2006.

[21] P. Sprechmann, A. Bronstein, and G. Sapiro. Real-time online singing voice separation from monaural recordings using robust low-rank modeling. In *ISMIR*, 2012.

[22] E. Vincent, N. Bertin, and R. Badeau. Adaptive harmonic spectral decomposition for multiple pitch estimation. *IEEE Trans. Audio, Speech, and Language Proc.*, 18(3):528–537, 2010.