Prediction and Inverse Problems in Dynamical Systems

Joan Bruna, Pablo Sprechmann

August 4, 2014

1 Status/Planning

The problem is now well defined. We study the following general question: Given temporally coherent data, how to learn a nonlinear representation that linearizes dynamics?

In particular, we use two criteria to evaluate models. The first one is prediction error. Given an observed sequence x_t, we construct an estimate $\hat{x}_t$, either from the past observations (casual inference), or from past and future (sort of non-linear interpolation). Currently we measure performance with $\|x_t - \hat{x}_t\|$, which corresponds to a Gaussian likelihood around a point estimate.

The second is denoising or source separation. We have observations $y_t = x_{1,t} + x_{2,t}$, where $x_{i,t}$ $i = 1, 2$ correspond to different sources, or source and noise. Assuming we have training sets for each of the sources (or only one of them), we attempt to estimate $\hat{x}_{i,t}$ by exploiting the temporal dynamics of each source. We are starting with speech data, and then we will eventually try video data.

So far we have tested:

- NMF Kalman dynamics on each speaker separately, and on the joint speakers.
- Spatio-temporal group lasso on log spectrogram.
- Pooling operators (analysis) on the log spectrogram.
- "LISTA" predictors using only two past frames.
- Frequency scattering on the spectrogram, followed by linear prediction.

Conclusions so far:

- Linear dynamics on NMF synthesis coefficients is too strong: it cannot disentangle position and momentum.
- Non-linear decoders from pooling are unstable. The gains produced by the pooling are ofset by these unstablilities in our experiments so far.

- Reconstruction based dictionary learning converges too fast, suggesting that bi-level with prediction task is necessary.
- Having a Gaussian posterior for prediction might be too ambitious in general.

Next things to do:

- Construct simple synthetic data with oracle dynamics (smooth warpings, jitter, etc).
- Bi-level NMF with simple linear dynamics.
- Lista version.
- Implement RNNs as a matter of comparison (which ones? LSTMs, sigmoids).
- Bi-level group lasso where we put the dynamics on the modulus.
- Bilinear models (with exponential maps?)

Introduction

Extracting information from unlabeled data remains the main challenge of unsupervised learning. In this work we consider semi-supervised learning, in which one observes the evolution of a dynamical system and attempts to learn through the underlying dynamics. The main strategy to understand the dynamics is by linearizing them through an appropriate non-linear representation.

This work considers the setting of speech and temporal video data. There are two main taks we are interested in: prediction and source separation/denoising. Other works on NLP have shown that training systems to perform prediction is a very effective surrogate that can be applied to other tasks, such as recognition. On the other hand, source separation is a major application of speech representations that requires exploiting the temporal coherence of different sources.

Dynamics can be learnt with a variety of models. The simplest are Kalman Filter and Hidden Markov Models, which learn dynamics in the form of linear equations, assuming Gaussian distributions in the former and discrete variables in the latter. On the other hand, recent works on speech and natural language processing have developed Recurrent Neural Networks (RNNs), which the capacity to learn more complex nonlinear dynamics.

Our objective is to develop a model which progressively moves from linear to RNNs, which keeps the interpretability but also handles non-linear dynamics. Since we are interested in inverse problems, models need to be generative.

Given a temporally ordered sequence $x_t \in \mathbb{R}^N$, we can write $x_t = U_t x_{t-1}$, where U_t is any operator in $\mathbb{R}^{N \times N}$. Clearly, there is too much freedom in the choice of these dynamics. The goal is to estimate x_t by inferring U_t on a much restricted class of possible dynamics.

One particular class of dynamics is to write $U_t = DA\Phi$, where $x_t \approx D\Phi(x_t)$

We assume a dynamical system $z(t)$ governed by some dynamics $z(t+\Delta) = F(z(t), \partial z(t))$ in a high-dimensional space. Our observations $x(t)$ are obtained

Summary of contributions:

- bla
- bla

2 Linearizing Dynamics

2.1 Linear NMF Dynamics

This is the basic first model:

$$\min_{z_t, A, D} \|x_t - Dz_t\|^2 + \lambda \|z_t\|_1 + \mu \|z_t - Az_{t-1}\|^2 \tag{1}$$

Properties:

- + Easy to train, compact model with few parameters
- + Gives good results on source separation
- + It is easily interpreted as a sparse Kalman FIlter.
- - The model has a fundamental limitation in that the dynamics on the hidden states are linear.
- - All the uncertainty/inference power in the model is in the latent variables z_t. The temporal relationship is rigid and given by a single linear operator A. The model can capture different trajectories, but at the expense of making the dictionary D large and unstable.
- - Optimiziation is too "easy": there are many local minima, and the objective function is not directly optimizing what we want.

2.2 Position and Momentum Dynamics

We want to explicitly disentangle the representation of position, encoding the current state of the system, from the representation of momentum, encoding how the position is going to evolve in time:

$$\mu_{t+1} = \mathcal{F}(\mu_t, v_t) \ ,$$
$$v_{t+1} \approx v_t \ , \text{ or } \ v_{t+1} \approx Av_t \ ,$$

In that setting, our observations are typically obtained from μ_t through a linear decoder. The critical part of the model is the operator $\mathcal{F}$. If the μ_t are sparse, then it is natural to expect that the momentum will act by moving the active set. In that case, we want to model

$$\mathcal{F}(\mu, v)(k) = \mu(k - v(k)) \ .$$

Here, v can model a deformation field in the dictionary where μ is defined, or a more compact feature vector encoding only smooth vector fields.

Another model is to consider a bilinear operator:

$$\mathcal{F}(\mu, v)(k) = \sum_{i,j} v(i)\mu(j)F_{i,j,k} \ ,$$

And yet a simpler model is the linear version:

$$\mathcal{F}(\mu, v) = F_1\mu + F_2 v \ ,$$

The question is how these dynamics compare with non-linear models of RNNs:

$$[\mu_{t+1}, v_{t+1}] = \rho(F_\mu \mu_t + F_v v_t) \ .$$

2.3 The exponential mapping

2.4 Bilevel Linear NMF

A modification of the previous model is to train it discriminatively to predict the next frame:

If we denote

$$z *_t (D, A, x) = \arg\min_z \|x_t - Dz_t\|^2 + \|z_t\|_1 + \|z_t - Az_{t-1}\|^2 \ ,$$

we optimize

$$\min_{D,A,E} \|x_{t+1} - Ez *_t (D, A, x)\|^2 \ .$$

(Lista version) also

2.5 Linear Pooling NMF

2.6 Bi-Linear NMF

3 Examples: Newton Dynamics, Jitter

4 Pooling and Scattering

Optical FLow

5 Relationship to Previous work

Cedric et al. Linear dynamics on NMF.
RNN for music
RNN for speech recognition (Graves et al).
Yann's papers.

6 Experimental Results

Prediction
Source Separation.
Important points:

- deformation operators are phase modulations
- proximal operators
- link with optical flow estimation
- cascade: we must show at least two layers of prediciton.
- scattering; link with commutation error.
- deep network performs gradient steps of a proximal operator: this ressembles LISTA.
- causal vs non-causal
- relationship with RNN
- Consider a model: jitter, local deformations. Can I prove things there? What is the optimum system?