

Linear Models: FA, ICA, NFA

Shikui Tu

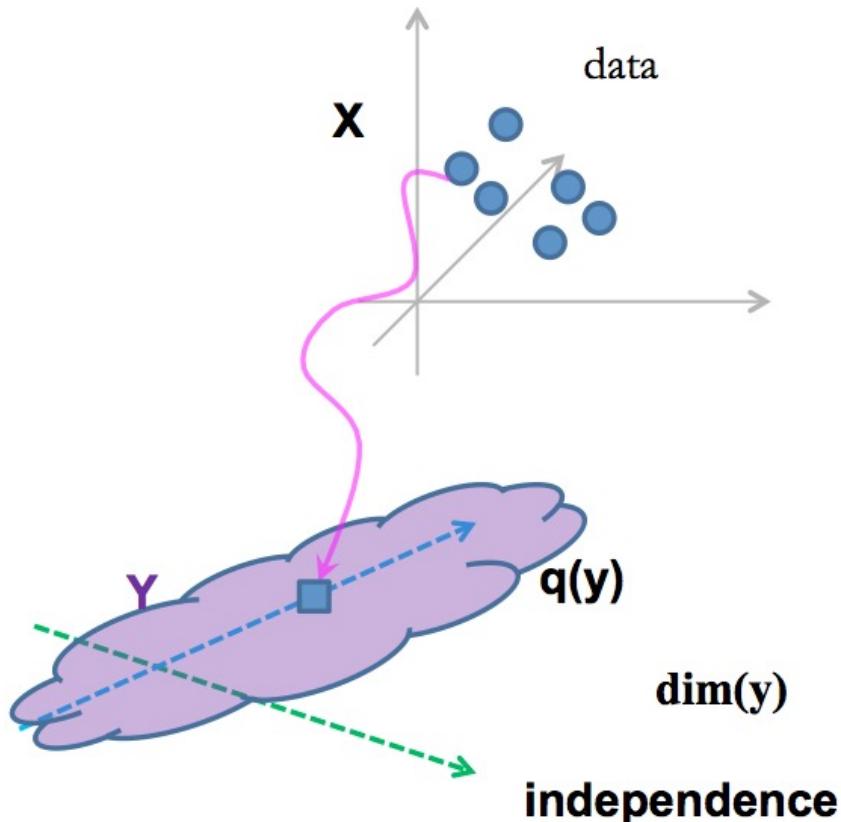
**Department of Computer Science and
Engineering, Shanghai Jiao Tong University**

2021-04-30

Outline

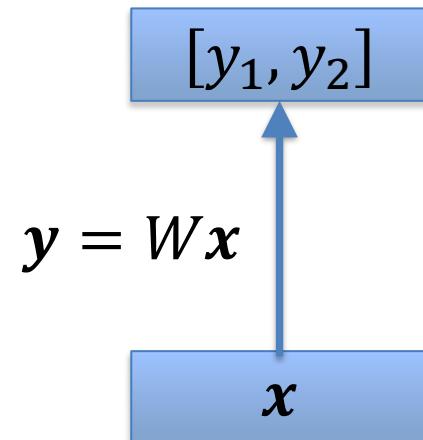
- Independent Component Analysis (ICA)
- Independent FA (IFA), Non-Gaussian FA (NFA)
- Recent papers related to PCA/ICA/GMM

Independent Component Analysis (ICA)



Find W such that

$$p(\mathbf{y}) = p(y_1) \cdots p(y_m)$$

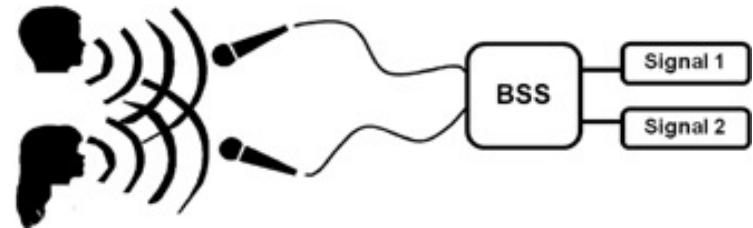


Blind Source Separation (BSS)

$$\mathbf{x} = \mathbf{A}\mathbf{s}$$

$$x_1(t) = a_{11}s_1 + a_{12}s_2$$

$$x_2(t) = a_{21}s_1 + a_{22}s_2$$



s_1 and s_2

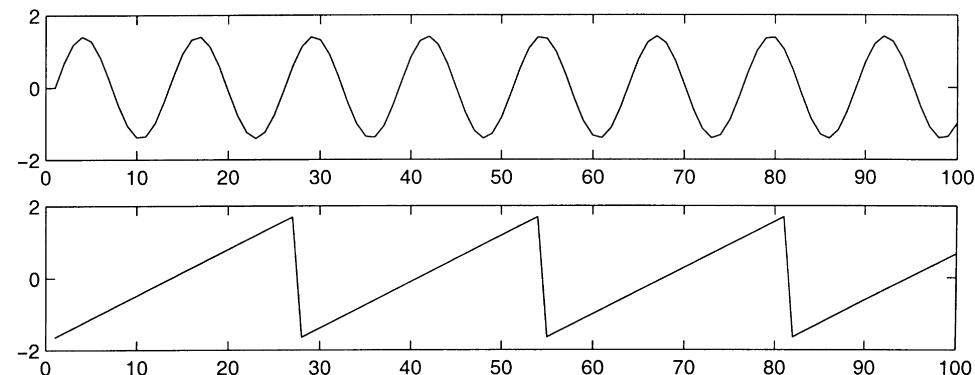


Fig. 1. The original signals.

x_1 and x_2

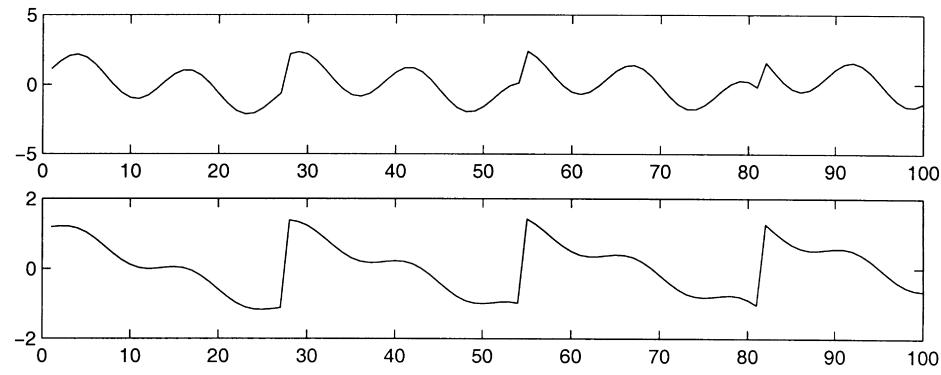


Fig. 2. The observed mixtures of the source signals in Fig. 1.

Task: Find W , and compute $\mathbf{s} = W\mathbf{x}$

Principles of ICA estimation

- “Non-Gaussian is independent”
 - Kurtosis or the fourth order cumulant
 - Negentropy
 - Mutual information

Outline

- FA and PCA
- Independent Component Analysis (ICA)
- **Independent FA (IFA), Non-Gaussian FA (NFA)**
- Recent papers related to PCA/ICA/GMM

Generative model for ICA with noise

- Generate independent, non-Gaussian source signals $\mathbf{y} = [y_1, \dots, y_k]$;
- Generate $\mathbf{x} = \Lambda\mathbf{y} + \mathbf{e}$, where $\mathbf{e} \sim G(\mathbf{e}|\mathbf{0}, \Psi)$ is Gaussian noise.

How to represent a non-Gaussian distribution?

$$\mathbf{x} = \Lambda\mathbf{y} + \mathbf{e}$$

$$x_j = \sum_{i=1}^k \lambda_{ji} y_i + e_j$$

$$p(\mathbf{y}) = \prod_{i=1}^k f(y_i)$$

$$p(\mathbf{x}|\mathbf{y}) = G(\mathbf{x}|\Lambda\mathbf{y}, \Psi)$$

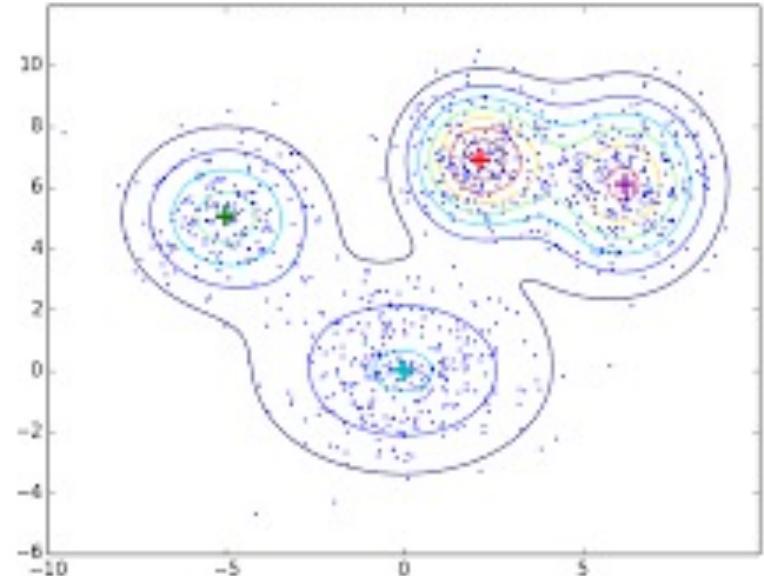
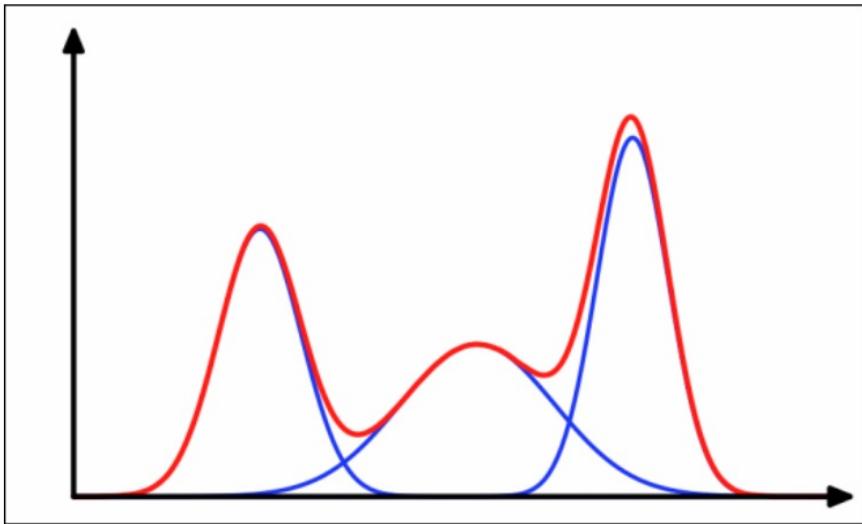
Use GMM!

$$f(y_i) = \sum_{q_i=1}^{m_i} w_{i,q_i} \mathcal{N}(\mu_{i,q_i}, \nu_{i,q_i})$$

Latent variables:

- Factors $\mathbf{y} = [y_1, \dots, y_k]$;
- Allocation variable $\mathbf{z} = [0, \dots, 0, 1, 0, \dots, 0]$

GMM for density estimation



$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

$$0 \leq \pi_k \leq 1$$

$$\sum_k \pi_k = 1$$

GMM can be used to estimate any probability density functions.

Equivalently as a GMM model

The distribution of \mathbf{x} :

$$\begin{aligned} f(\mathbf{x}|\Theta) &= \sum_{\mathbf{z}} \int f(\mathbf{x}, \mathbf{y}, \mathbf{z}|\Theta) d\mathbf{y} \\ &= \sum_{\mathbf{z}} \int f(\mathbf{z}|\Theta) f(\mathbf{y}|\mathbf{z}, \Theta) f(\mathbf{x}|\mathbf{y}, \mathbf{z}, \Theta) d\mathbf{y} \\ &= \sum_{\mathbf{z}} f(\mathbf{z}|\Theta) f(\mathbf{x}|\mathbf{z}, \Theta) \end{aligned}$$

Θ denotes the set of parameters.

$$\begin{aligned} f(\mathbf{x}|\mathbf{y}, \mathbf{z}, \Theta) &= \mathcal{N}(\Lambda\mathbf{y}, \Psi) \\ f(\mathbf{y}|\mathbf{z}, \Theta) &= \mathcal{N}(\boldsymbol{\mu}_{\mathbf{z}}, \mathbf{V}_{\mathbf{z}}) \end{aligned} \quad \left\{ \begin{array}{l} \boldsymbol{\mu}_{\mathbf{z}} = \left[\prod_{q_1=1}^{m_1} \mu_{1,q_1}^{z_{1,q_1}}, \dots, \prod_{q_k=1}^{m_k} \mu_{k,q_k}^{z_{k,q_k}} \right] \\ \mathbf{V}_{\mathbf{z}} = \text{diag} \left[\prod_{q_1=1}^{m_1} \nu_{1,q_1}^{z_{1,q_1}}, \dots, \prod_{q_k=1}^{m_k} \nu_{k,q_k}^{z_{k,q_k}} \right] \end{array} \right.$$



$$f(\mathbf{x}|\mathbf{z}, \Theta) = \mathcal{N}(\Lambda\boldsymbol{\mu}_{\mathbf{z}}, \Lambda\mathbf{V}_{\mathbf{z}}\Lambda^T + \Psi)$$

Then, $f(x|\Theta)$ is a GMM with the number of Gaussians: $m = \prod_{i=1}^k m_i$

EM for maximum likelihood

Likelihood function: $f(\mathbf{x}|\Theta) = \sum_{\mathbf{z}} \int f(\mathbf{z}, \mathbf{y}, \mathbf{x}|\Theta) d\mathbf{y},$

$$f(\mathbf{x}, \mathbf{y}, \mathbf{z}|\Theta) = f(\mathbf{x}|\mathbf{y}, \Theta) f(\mathbf{y}|\mathbf{z}, \Theta) f(\mathbf{z}|\Theta)$$

E-Step: Compute the posterior $f(\mathbf{z}, \mathbf{y}|\mathbf{x}, \Theta) = f(\mathbf{y}|\mathbf{x}, \mathbf{z}, \Theta) f(\mathbf{z}|\mathbf{x}, \Theta)$

$$f(\mathbf{y}|\mathbf{x}, \mathbf{z}) = \mathcal{N}(\rho_{\mathbf{z}}(\mathbf{x}), \Sigma_{\mathbf{z}})$$

$$\Sigma_{\mathbf{z}} = (\Lambda^T \Psi^{-1} \Lambda + \mathbf{V}_{\mathbf{z}}^{-1})^{-1}$$

$$f(\mathbf{z}|\mathbf{x}, \Theta) \propto f(\mathbf{x}|\mathbf{z}, \Theta) f(\mathbf{z}|\Theta)$$

$$\rho_{\mathbf{z}}(\mathbf{x}) = \Sigma_{\mathbf{z}} (\Lambda^T \Psi^{-1} \mathbf{x} + \mathbf{V}_{\mathbf{z}}^{-1} \mu_{\mathbf{z}})^{-1}$$

$$E[\mathbf{y}^T|\mathbf{x}], \quad E[\mathbf{y}\mathbf{y}^T|\mathbf{x}]$$

M-Step: $\arg \max_{\Theta} E_{\mathbf{z}, \mathbf{y}|\mathbf{x}, \Theta'} [\ln f(\mathbf{z}, \mathbf{y}, \mathbf{x}|\Theta)]$

$$\hat{\Lambda} = \mathbf{x} E[\mathbf{y}^T|\mathbf{x}] E[\mathbf{y}\mathbf{y}^T|\mathbf{x}]^{-1} \quad \hat{\Psi} = \mathbf{x}\mathbf{x}^T - \mathbf{x} E[\mathbf{y}^T|\mathbf{x}] \Lambda^T$$

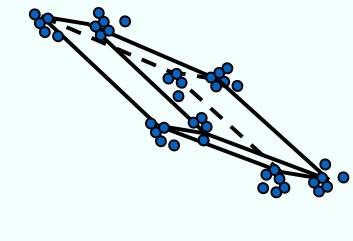
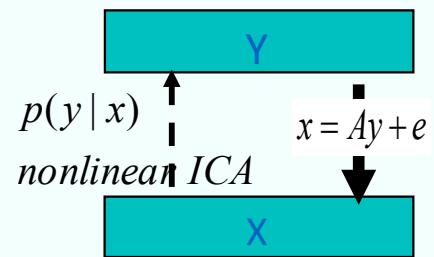
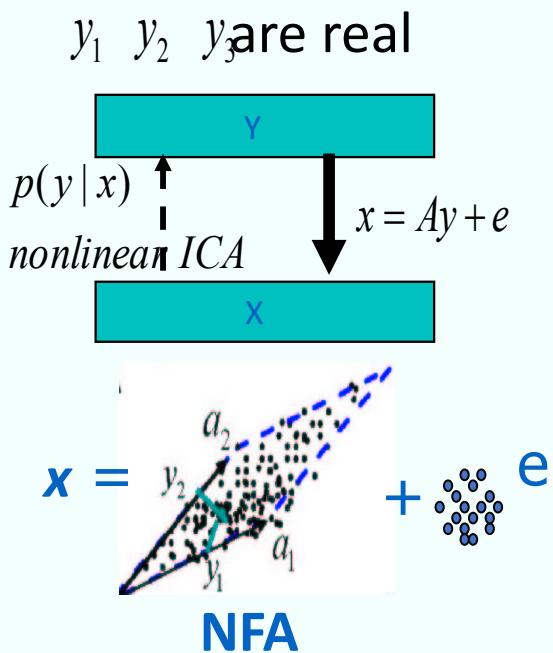
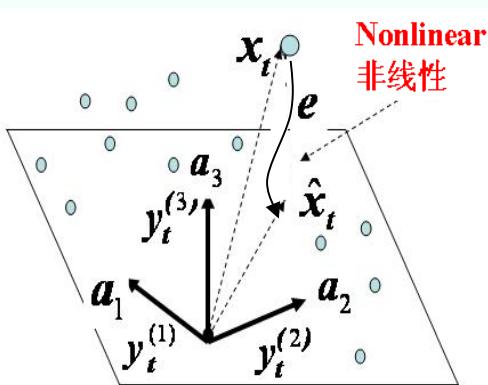
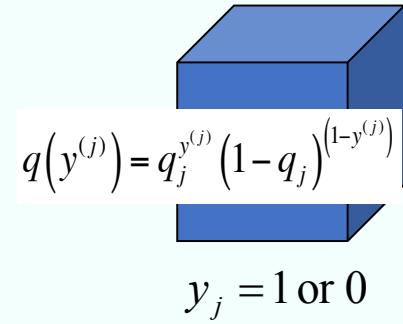
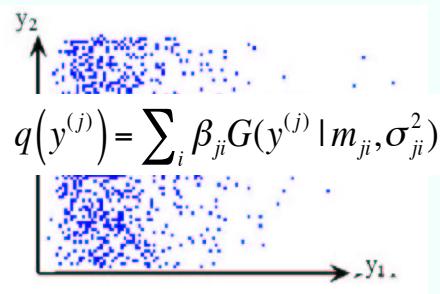
$$\hat{\mu}_{i,q_i} = \frac{f(z_i|\mathbf{x}) E[y_i|z_i, \mathbf{x}]}{f(z_i|\mathbf{x})} \quad \hat{\nu}_{i,q_i} = \frac{f(z_i|\mathbf{x}) E[y_i^2|z_i, \mathbf{x}]}{f(z_i|\mathbf{x})} - \hat{\mu}_{i,q_i}^2$$

$$\hat{w}_{i,q_i} = f(z_i|\mathbf{x})$$

Summation w.r.t. \mathbf{x} was ignored for simplicity.

Non-linear, non-Gaussian FA

$$q(y) = \prod_{j=1}^k q(y^{(j)})$$



BFA

Outline

- FA and PCA
- Independent Component Analysis (ICA)
- Independent FA (IFA), Non-Gaussian FA (NFA)
- Recent papers related to PCA/ICA/GMM

Deep Mixtures of Factor Analysers

Yichuan Tang

Ruslan Salakhutdinov

Geoffrey Hinton

Department of Computer Science, University of Toronto, Toronto, Ontario, CANADA
Abstract

An efficient way to learn deep density models that have many layers of latent variables is to learn one layer at a time using a model that has only one layer of latent variables. After learning each layer, samples from the posterior distributions for that layer are used as training data for learning the next layer. This approach is commonly used with Restricted Boltzmann Machines, which are *undirected* graphical models with a single hidden layer, but it can also be used with Mixtures of Factor Analysers (MFAs) which are *directed* graphical models. In this paper, we present a greedy layer-wise learning algorithm for Deep Mixtures of Factor Analysers (DMFAs). Even though a DMFA can be converted to an equivalent shallow MFA by multiplying together the factor loading matrices at different levels, learning and inference are much more efficient in a DMFA and the sharing of each lower-level factor loading matrix by many different higher level MFAs prevents overfitting. We demonstrate empirically that DMFAs learn better density models than both MFAs and two types of Restricted Boltzmann Machine on a wide variety of datasets.

Let $\mathbf{x} \in \mathbb{R}^D$ denote the D -dimensional data, $\{\mathbf{z} \in \mathbb{R}^d : d \leq D\}$ denote the d -dimensional latent variable, and $c \in \{1, \dots, C\}$ denote the component indicator variable of C total components. The MFA is a directed generative model, defined as follows:

$$p(c) = \pi_c, \quad \sum_{c=1}^C \pi_c = 1, \quad (1)$$

$$p(\mathbf{z}|c) = p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I}), \quad (2)$$

$$p(\mathbf{x}|\mathbf{z}, c) = \mathcal{N}(\mathbf{x}; \mathbf{W}_c \mathbf{z} + \boldsymbol{\mu}_c, \boldsymbol{\Psi}_c), \quad (3)$$

$$p(\mathbf{x}|c) = \int_{\mathbf{z}} p(\mathbf{x}|\mathbf{z}, c)p(\mathbf{z}|c)d\mathbf{z} = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_c, \Gamma_c) \quad (4)$$

$$\Gamma_c = \mathbf{W}_c \mathbf{W}_c^T + \boldsymbol{\Psi}_c$$

$$p(\mathbf{x}) = \sum_{c=1}^C \pi_c \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_c, \Gamma_c). \quad (5)$$

$$p(\mathbf{z}|\mathbf{x}, c) = \mathcal{N}(\mathbf{z}; \mathbf{m}_c, \mathbf{V}_c^{-1}), \quad (8)$$

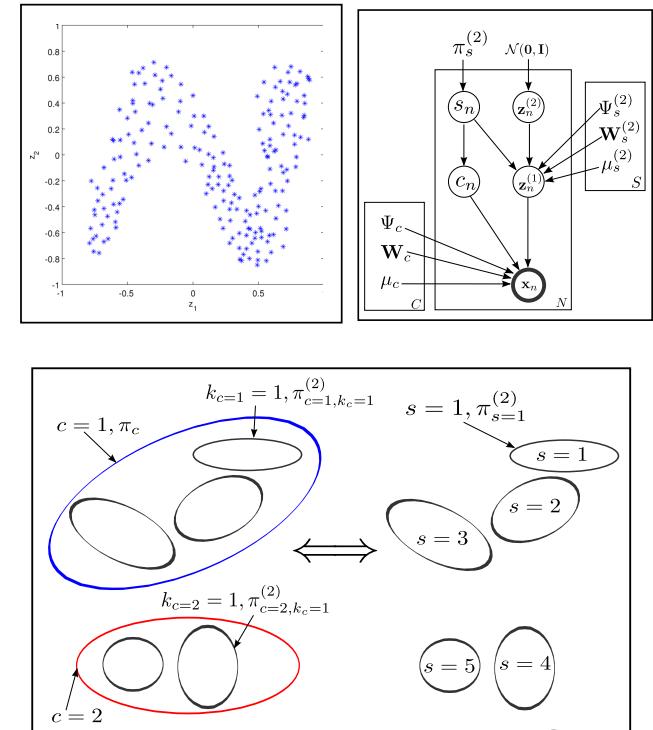
where

$$\begin{aligned} \mathbf{V}_c &= \mathbf{I} + \mathbf{W}_c^T \boldsymbol{\Psi}_c^{-1} \mathbf{W}_c, \\ \mathbf{m}_c &= \mathbf{V}_c^{-1} \mathbf{W}_c^T \boldsymbol{\Psi}_c^{-1} (\mathbf{x} - \boldsymbol{\mu}). \end{aligned}$$

TANG@CS.TORONTO.EDU

RSALAKHU@CS.TORONTO.EDU

HINTON@CS.TORONTO.EDU



Rethinking LDA: Moment Matching for Discrete ICA

Anastasia Podosinnikova Francis Bach Simon Lacoste-Julien
INRIA - École normale supérieure Paris

Abstract

We consider moment matching techniques for estimation in latent Dirichlet allocation (LDA). By drawing explicit links between LDA and discrete versions of independent component analysis (ICA), we first derive a new set of cumulant-based tensors, with an improved sample complexity. Moreover, we reuse standard ICA techniques such as joint diagonalization of tensors to improve over existing methods based on the tensor power method. In an extensive set of experiments on both synthetic and real datasets, we show that our new combination of tensors and orthogonal joint diagonalization techniques outperforms existing moment matching methods.



Michael I. Jordan

[FOLLOW](#)

Professor of EECS and Professor of Statistics, [University of California, Berkeley](#)

Verified email at cs.berkeley.edu - [Homepage](#)

[machine learning](#) [statistics](#) [computational biology](#) [artificial intelligence](#) [optimization](#)

Cited by

[VIEW ALL](#)

All

Since 2013

Citations

132283

64236

h-index

144

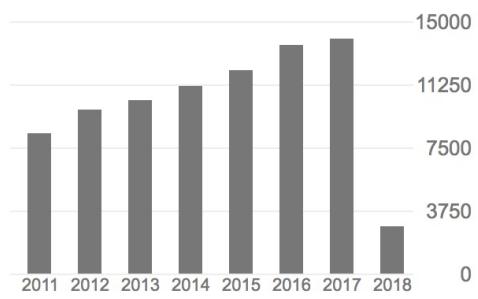
98

i10-index

456

358

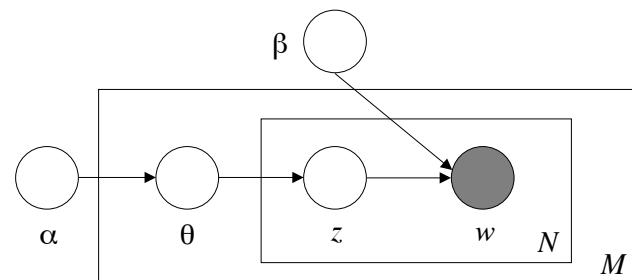
TITLE	CITED BY	YEAR
Latent dirichlet allocation DM Blei, AY Ng, MI Jordan Journal of machine Learning research 3 (Jan), 993-1022	22234	2003
On spectral clustering: Analysis and an algorithm AY Ng, MI Jordan, Y Weiss Advances in neural information processing systems, 849-856	6489	2002
Adaptive mixtures of local experts	3683	1991



Latent Dirichlet allocation (LDA) is a generative probabilistic model of a corpus. The basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words.¹

LDA assumes the following generative process for each document w in a corpus D :

1. Choose $N \sim \text{Poisson}(\xi)$.
2. Choose $\theta \sim \text{Dir}(\alpha)$.
3. For each of the N words w_n :
 - (a) Choose a topic $z_n \sim \text{Multinomial}(\theta)$.
 - (b) Choose a word w_n from $p(w_n | z_n, \beta)$, a multinomial probability conditioned on the topic z_n .



PCANet: A Simple Deep Learning Baseline for Image Classification?

Tsung-Han Chan, *Member, IEEE*, Kui Jia, Shenghua Gao, Jiwen Lu, *Senior Member, IEEE*, Zinan Zeng, and Yi Ma, *Fellow, IEEE*

Abstract—In this paper, we propose a very simple deep learning network for image classification that is based on very basic data processing components: 1) cascaded principal component analysis (PCA); 2) binary hashing; and 3) blockwise histograms. In the proposed architecture, the PCA is employed to learn multistage filter banks. This is followed by simple binary hashing and block histograms for indexing and pooling. This architecture is thus called the PCA network (PCANet) and can be extremely easily and efficiently designed and learned. For comparison and to provide a better understanding, we also introduce and study two simple variations of PCANet: 1) RandNet and 2) LDANet. They share the same topology as PCANet, but their cascaded filters are either randomly selected or learned from linear discriminant analysis. We have extensively tested these basic networks on many benchmark visual data sets for different tasks, including Labeled Faces in the Wild (LFW) for face verification; the MultiPIE, Extended Yale B, AR, Facial Recognition Technology (FERET) data sets for face recognition; and MNIST for hand-written digit recognition. Surprisingly, for all tasks, such a seemingly naive PCANet model is on par with the state-of-the-art features either prefixed, highly hand-crafted, or carefully learned [by deep neural networks (DNNs)]. Even more surprisingly, the model sets new records for many classification tasks on the Extended Yale B, AR, and FERET data sets and on MNIST variations. Additional experiments on other public data sets also demonstrate the potential of PCANet to serve as a simple but highly competitive baseline for texture classification and object recognition.

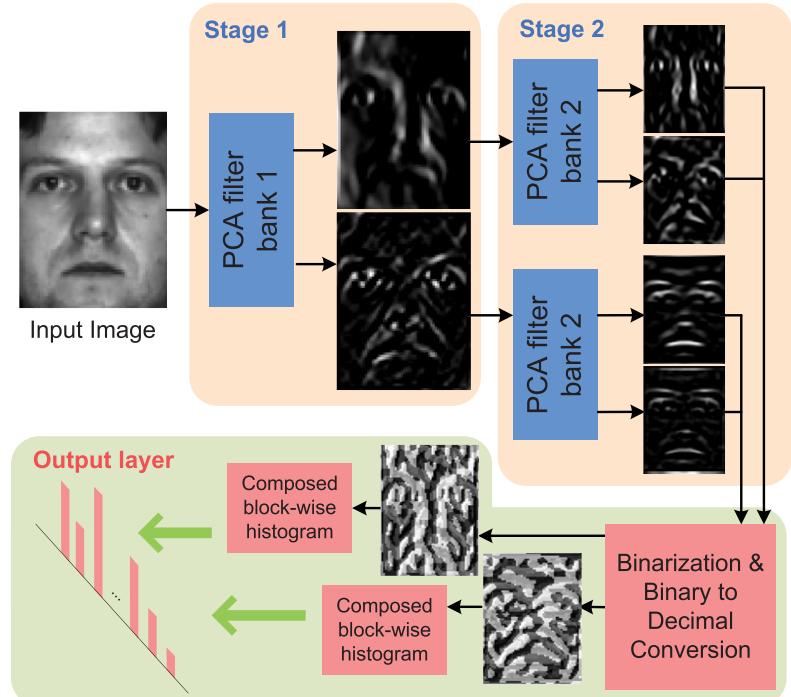


Fig. 1. Illustration of how the proposed PCANet extracts features from an image through the three simplest processing components: PCA filters, binary hashing, and histograms.

Global Analysis of Expectation Maximization for Mixtures of Two Gaussians

Ji Xu

Columbia University

jixu@cs.columbia.edu

Daniel Hsu

Columbia University

djhsu@cs.columbia.edu

Arian Maleki

Columbia University

arian@stat.columbia.edu

Abstract

Expectation Maximization (EM) is among the most popular algorithms for estimating parameters of statistical models. However, EM, which is an iterative algorithm based on the maximum likelihood principle, is generally only guaranteed to find stationary points of the likelihood objective, and these points may be far from any maximizer. This article addresses this disconnect between the statistical principles behind EM and its algorithmic properties. Specifically, it provides a global analysis of EM for specific models in which the observations comprise an i.i.d. sample from a mixture of two Gaussians. This is achieved by (i) studying the sequence of parameters from idealized execution of EM in the infinite sample limit, and fully characterizing the limit points of the sequence in terms of the initial parameters; and then (ii) based on this convergence analysis, establishing statistical consistency (or lack thereof) for the actual sequence of parameters produced by EM.

1.1 Expectation Maximization

Among the algorithms mentioned above, Expectation Maximization (EM) has attracted more attention for the simplicity of its iterations, and its good performance in practice (Dempster et al., 1977; Redner and Walker, 1984). EM is an iterative algorithm for climbing the likelihood objective starting from an initial setting of the parameters $\hat{\boldsymbol{\eta}}^{(0)}$. In iteration t , EM performs the following steps:

$$\text{E-step:} \quad \hat{Q}(\boldsymbol{\eta} | \hat{\boldsymbol{\eta}}^{(t)}) \triangleq \sum_{\mathbf{z}} f(\mathbf{z} | \mathcal{Y}; \hat{\boldsymbol{\eta}}^{(t)}) \log f(\mathcal{Y}, \mathbf{z}; \boldsymbol{\eta}), \quad (1)$$

$$\text{M-step:} \quad \hat{\boldsymbol{\eta}}^{(t+1)} \triangleq \arg \max_{\boldsymbol{\eta}} \hat{Q}(\boldsymbol{\eta} | \hat{\boldsymbol{\eta}}^{(t)}), \quad (2)$$

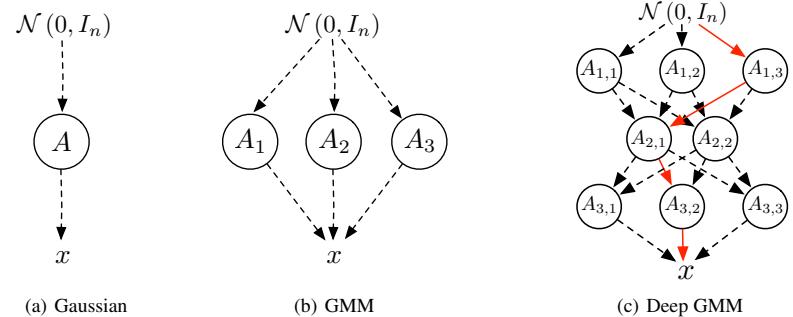
Factoring Variations in Natural Images with Deep Gaussian Mixture Models

Aäron van den Oord, Benjamin Schrauwen

Electronics and Information Systems department (ELIS), Ghent University
`{aaron.vandenoord, benjamin.schrauwen}@ugent.be`

Abstract

Generative models can be seen as the swiss army knives of machine learning, as many problems can be written probabilistically in terms of the distribution of the data, including prediction, reconstruction, imputation and simulation. One of the most promising directions for unsupervised learning may lie in Deep Learning methods, given their success in supervised learning. However, one of the current problems with deep unsupervised learning methods, is that they often are harder to scale. As a result there are some easier, more scalable shallow methods, such as the Gaussian Mixture Model and the Student-t Mixture Model, that remain surprisingly competitive. In this paper we propose a new *scalable* deep generative model for images, called the Deep Gaussian Mixture Model, that is a straightforward but powerful generalization of GMMs to multiple layers. The parametrization of a Deep GMM allows it to efficiently capture products of variations in natural images. We propose a new EM-based algorithm that scales well to large datasets, and we show that both the Expectation and the Maximization steps can easily be distributed over multiple machines. In our density estimation experiments we show that deeper GMM architectures generalize better than more shallow ones, with results in the same ballpark as the state of the art.



NIPS 2016

Sparse PCA via Bipartite Matchings

Megasthenis Asteris

The University of Texas at Austin
megas@utexas.edu

Anastasios Kyrillidis

The University of Texas at Austin
anastasios@utexas.edu

Dimitris Papailiopoulos

University of California, Berkeley
dimitrisp@berkeley.edu

Alexandros G. Dimakis

The University of Texas at Austin
dimakis@austin.utexas.edu

Abstract

We consider the following multi-component sparse PCA problem: given a set of data points, we seek to extract a small number of sparse components with *disjoint* supports that jointly capture the maximum possible variance. Such components can be computed one by one, repeatedly solving the single-component problem and deflating the input data matrix, but this greedy procedure is suboptimal. We present a novel algorithm for sparse PCA that jointly optimizes multiple disjoint components. The extracted features capture variance that lies within a multiplicative factor arbitrarily close to 1 from the optimal. Our algorithm is combinatorial and computes the desired components by solving multiple instances of the bipartite maximum weight matching problem. Its complexity grows as a low order polynomial in the ambient dimension of the input data, but exponentially in its rank. However, it can be effectively applied on a low-dimensional sketch of the input data. We evaluate our algorithm on real datasets and empirically demonstrate that in many cases it outperforms existing, deflation-based approaches.

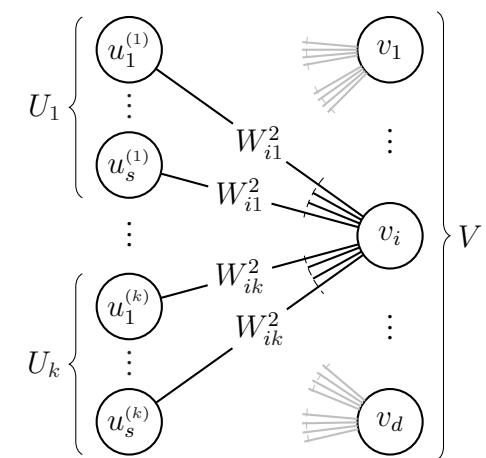


Figure 1: The graph G generated by Alg. 2. It is used to determine the support of the solution $\widehat{\mathbf{X}}$ in (6).

SPLICE: Fully Tractable Hierarchical Extension of ICA with Pooling

Jun-ichiro Hirayama^{1 2} Aapo Hyvärinen^{3 4} Motoaki Kawanabe^{2 1}

Abstract

We present a novel probabilistic framework for a hierarchical extension of independent component analysis (ICA), with a particular motivation in neuroscientific data analysis and modeling. The framework incorporates a general subspace pooling with linear ICA-like layers stacked recursively. Unlike related previous models, our generative model is fully tractable: both the likelihood and the posterior estimates of latent variables can readily be computed with analytically simple formulae. The model is particularly simple in the case of complex-valued data since the pooling can be reduced to taking the modulus of complex numbers. Experiments on electroencephalography (EEG) and natural images demonstrate the validity of the method.

2. Proposed Method

2.1. First-Layer Model

We begin with formulating the generative model for our SPLICE. Denote by \mathbf{x}_t observed data vectors ($t = 1, 2, \dots, n$), either real- or complex-valued, consisting of d entries x_{it} . Each of the d entries is given by a linear combination of the same number of unknown (first-layer) components or *sources*, collectively denoted as source vector \mathbf{s}_t . Here, we consider the fundamental case where \mathbf{x}_t and \mathbf{s}_t are independently and identically distributed (i.i.d.). Omitting sample index t for notational simplicity, we write

$$\mathbf{x} = \mathbf{As}, \quad (1)$$

where the coefficient matrix \mathbf{A} , called mixing matrix, is square and assumed to be invertible; the inverse $\mathbf{W} := \mathbf{A}^{-1}$ is called demixing matrix. For convenience, we as-

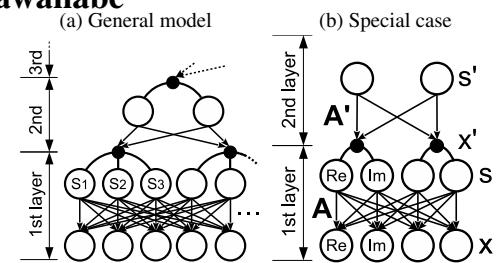


Figure 1: Generative model of SPLICE: (a) A higher layer directly gives the squared L_2 -norms of lower sources s within each subspace. (b) An important special case having one complex source s' per subspace.

Learning Independent Features with Adversarial Nets for Non-linear ICA

ICA [PDF](#)

Philemon Brakel, Yoshua Bengio

16 Feb 2018 ICLR 2018 Conference Blind Submission readers: everyone Show Bibtex Revisions

Abstract: Reliable measures of statistical dependence could potentially be useful tools for learning independent features and performing tasks like source separation using Independent Component Analysis (ICA). Unfortunately, many of such measures, like the mutual information, are hard to estimate and optimize directly. We propose to learn independent features with adversarial objectives (Goodfellow et al. 2014, Arjovsky et al. 2017) which optimize such measures implicitly. These objectives compare samples from the joint distribution and the product of the marginals without the need to compute any probability densities. We also propose two methods for obtaining samples from the product of the marginals using either a simple resampling trick or a separate parametric distribution. Our experiments show that this strategy can easily be applied to different types of model architectures and solve both linear and non-linear ICA problems.

Keywords: adversarial networks, ica, unsupervised, independence

ICLR 2018 Conference Acceptance Decision

ICLR 2018 Conference Program Chairs

30 Jan 2018 ICLR 2018 Conference Acceptance Decision readers: everyone

Decision: Reject

Comment: The paper proposes the use of GANs to match the joint distribution of features to the product totally plausible but reviewers have complaints about lack of rigor and analysis in terms of (i) mixing conditi approach will work, given that ICA is ill-posed for general nonlinear mixing (ii) comparison with prior work

Further, in most scenarios where GANs are used, one of the distributions is fixed (say, the real distribution trying to come close to the fixed distribution during optimization. In the proposed method, the discriminant product of marginals which are both dynamic during the learning. It might be useful to comment whether instability of training, etc.

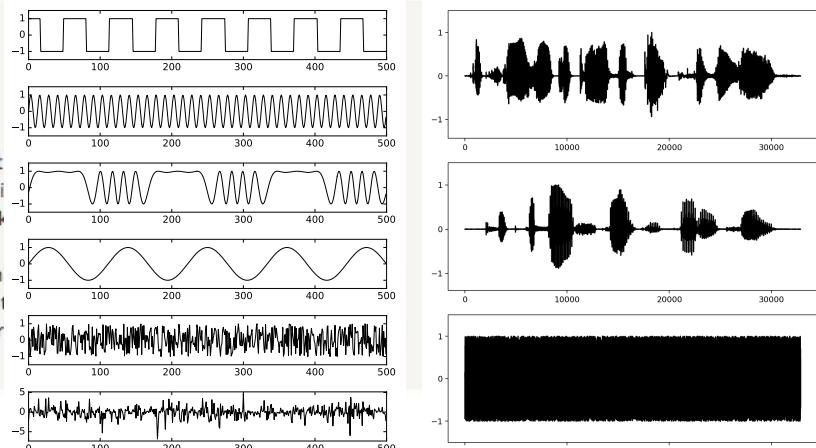


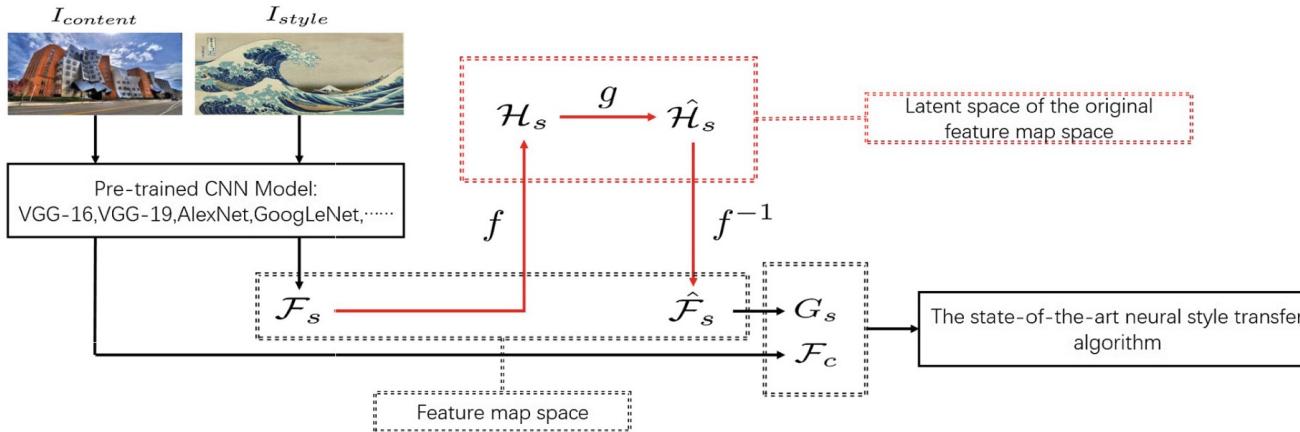
Figure 3: Source signals used in the experiments.

Computational Decomposition of Style for Controllable and Enhanced Style Transfer

Minchao Li, Shikui Tu^(✉), and Lei Xu^(✉)

Department of Computer Science and Engineering,
Center for Cognitive Machines and Computational Health (CMaCH),
Shanghai Jiao Tong University, Shanghai, China
`{marshal_lmc,tushikui,leixu}@sjtu.edu.cn`

Z. Cui et al. (Eds.): IScIDE 2019, LNCS 11936, pp. 15–39, 2019.



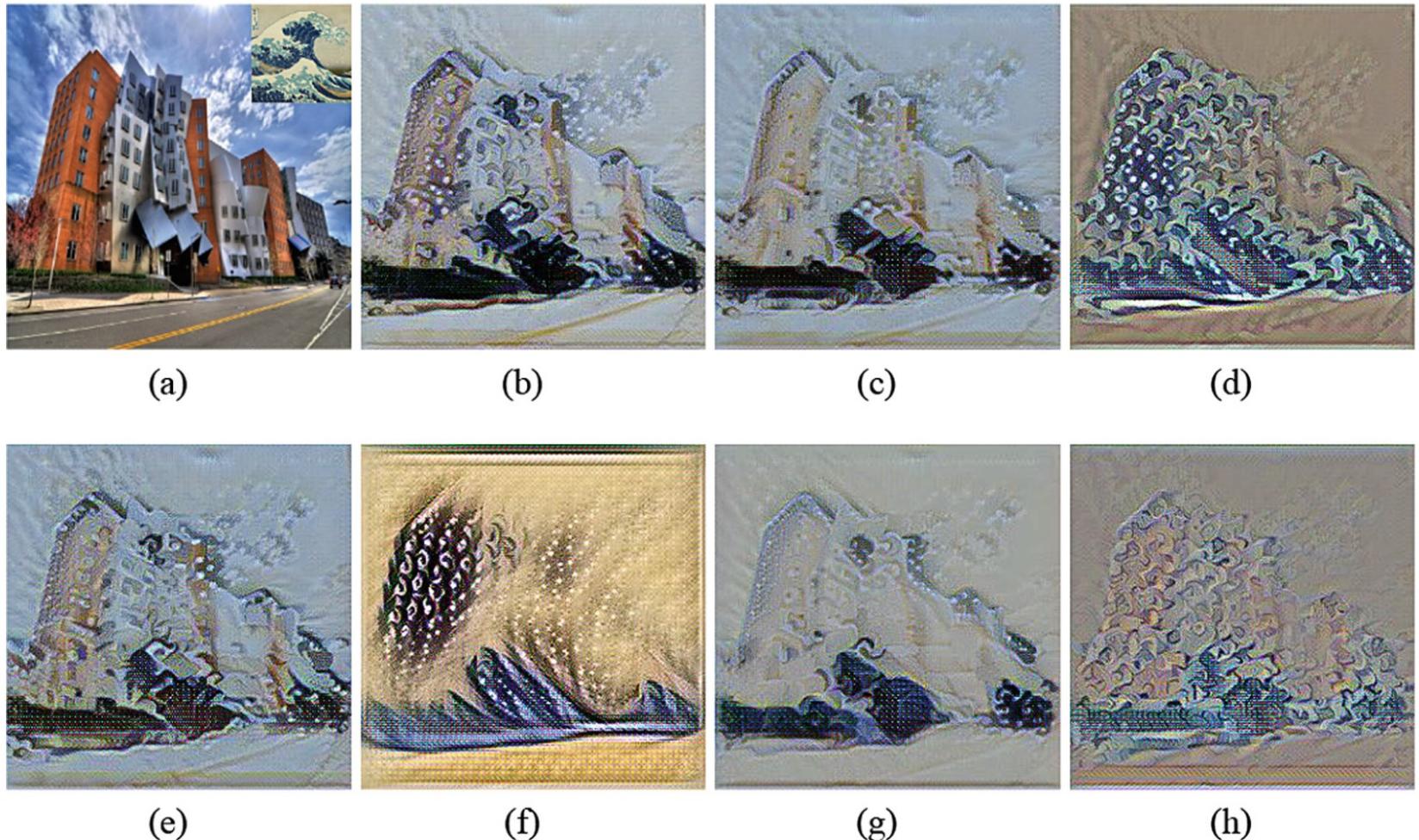


Fig. 3. (a) the original content and style images; (b) styled image by traditional neural style transfer; (c–h) results of preserving one style basis by different methods. Specifically, (c–d) FFT; (e–f) PCA; (g–h) ICA where (c, e, g) aim to transfer the color of style image while (d, f, h) aim to transfer the stroke of style image.

GANSpace: Discovering Interpretable GAN Controls



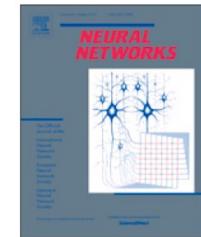
Figure 1: Sequences of image edits performed using control discovered with our method, applied to three different GANs. The white insets specify the edits using notation explained in Section 2.3.



Contents lists available at ScienceDirect

Neural Networks

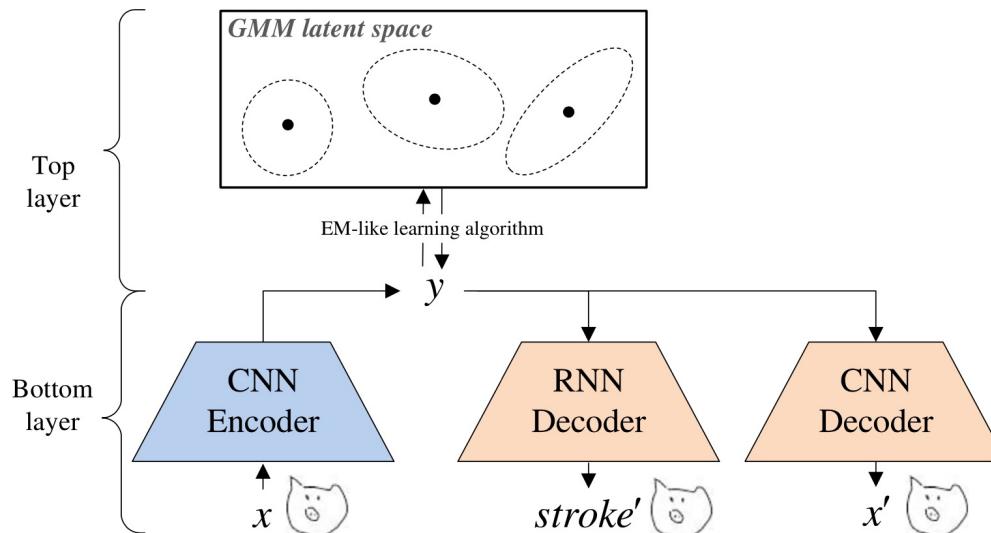
journal homepage: www.elsevier.com/locate/neunet



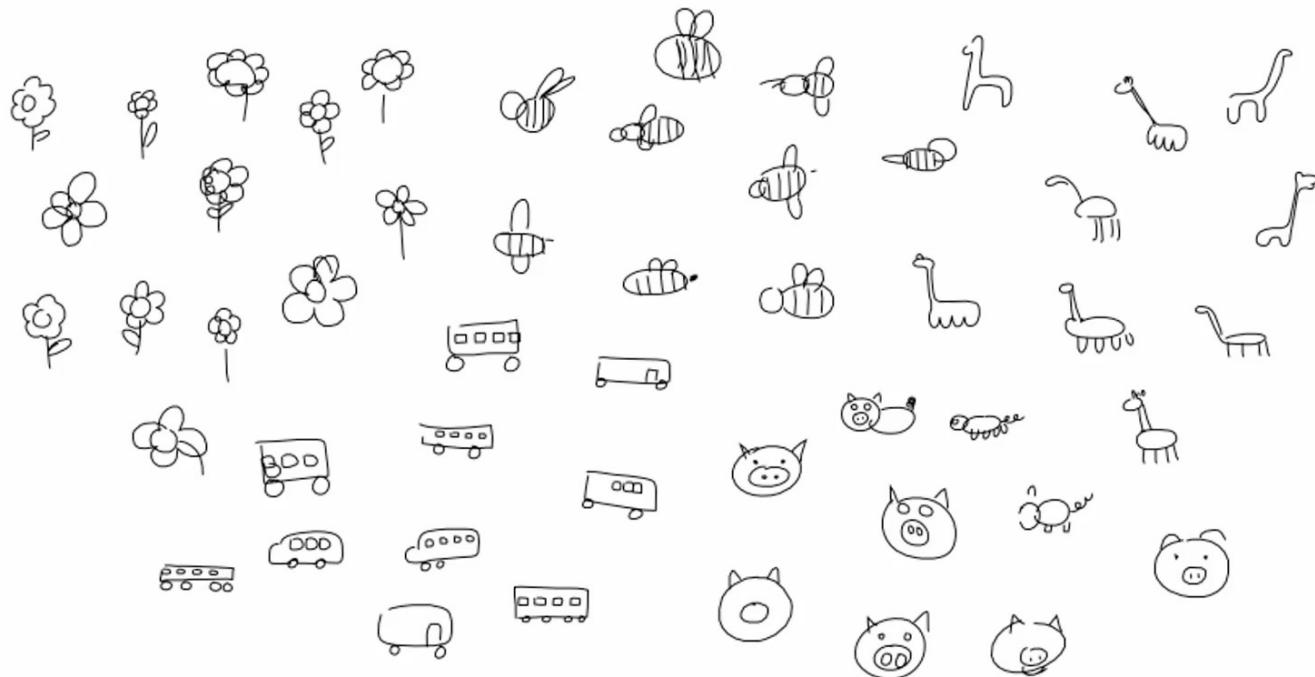
Controllable stroke-based sketch synthesis from a self-organized latent space

Sicong Zang, Shikui Tu ^{*}, Lei Xu ^{*}

Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China



Demo of RPCL-pix2seq



Thank you!