A New Framework to Learn Neural Networks: Hybrid Orthogonal Projection and Estimation (HOPE)

Hui Jiang
Department of Electrical Engineering and Computer Science
York University, Toronto, Canada

Joint work with Mr. Shiliang Zhang, Prof. Lirong Dai,
University of Science and Technology of China (USTC)

Outline

- Introduction
- Hybrid Orthogonal Projection and Estimation (HOPE)
- Link HOPE to Neural Networks (NN) and Deep Learning
  - Unsupervised Learning of NN under HOPE
  - Supervised Learning of NN under HOPE
  - HOPE for Deep Learning
- Experiments on MNIST and TIMIT
- Conclusions and What's next

Deep Learning Boom: Neural Networks Resurgence

- Neural networks (NN) resurge as deep learning *(Hinton et. al. 2006)*
- Deep neural networks (DNN) for speech recognition *(Seide et. al. 2011)*
- Convolutional neural networks (CNN) for image recognition *(Krizhevsky et al. 2012)*
- Recurrent neural networks (RNN) for natural language processing, speech recognition, …. *(Mikolov et. al. 2010; Graves, et. al 2014)*
- NN has achieved the new state-of-the-art performance for (almost) all supervised learning tasks:
  - *more data + bigger models + powerful GPUs*

Lessons We Have Learned so far

- Neural networks are powerful ***and learnable***…
- Need a powerful model for big-data in real applications …
- No longer afraid of ***complex optimization*** problems …

  Optimization of neural networks is high-dimension and non-convex
- Supervised learning of neural networks is ***(almost) solved*** …

  the so-called ***end-to-end*** learning …
- Neural networks can learn good ***feature representations*** …

Some Open Questions

- Why neural networks (NN) work?
- Why we need a deep structure?
- Why the simple SGD works so well for NNs?
- How to do unsupervised learning?
- How to interpret NN features?

Some Open Questions

- **Why neural networks (NN) work?**
- **Why need a deep structure?**
- Why the simple SGD works so well in learning NNs?
- **How to do unsupervised learning?**
- **How to interpret NN features?**

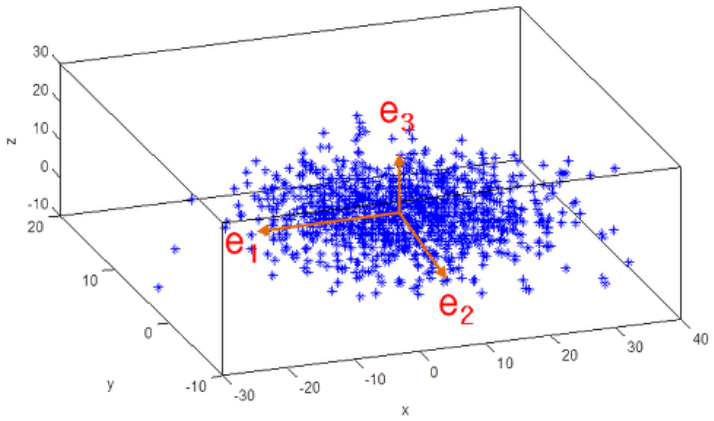
Outline

- Introduction
- **Hybrid Orthogonal Projection and Estimation (HOPE)**
- Link HOPE to Neural Networks (NN) and Deep Learning
  - Unsupervised Learning of NN under HOPE
  - Supervised Learning of NN under HOPE
  - HOPE for Deep Learning
- Experiments on MNIST and TIMIT
- Conclusions

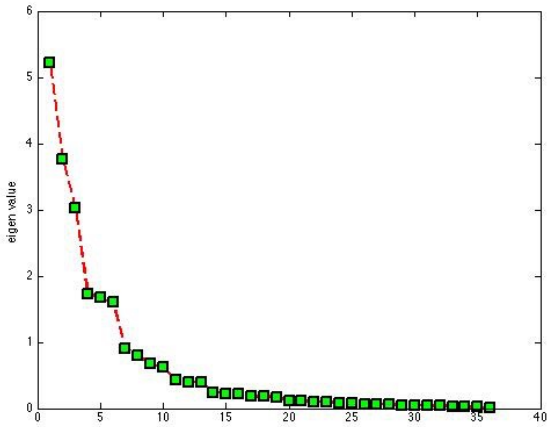
Linear Orthogonal Projection

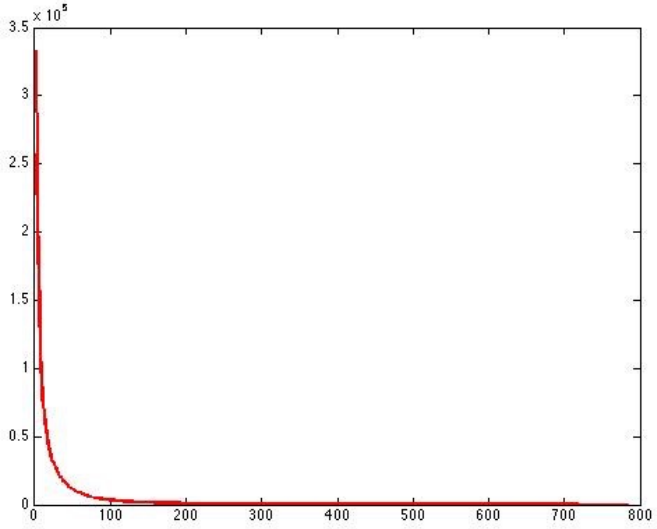
PCA is a special case of linear orthogonal projection

high-dimension data



variance (energy) along dimensions after PCA





MNIST

Hybrid Orthogonal Projection and Estimation (HOPE)

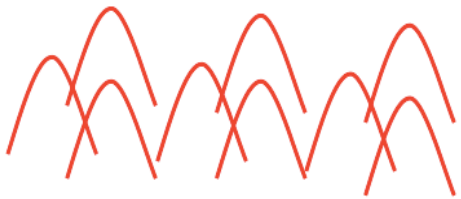
HOPE = a generative model (directed graphical model)
(linear orthogonal projection + a finite mixture model)

$$\mathbf{x} = \underbrace{(\mathbf{x}\cdot\mathbf{u}_1)\,\mathbf{u}_1 + \cdots + (\mathbf{x}\cdot\mathbf{u}_M)\,\mathbf{u}_M}_{signal\ \ component\ \ \tilde{\mathbf{x}}} + \underbrace{(\mathbf{x}\cdot\mathbf{u}_{M+1})\,\mathbf{u}_{M+1} + \cdots + (\mathbf{x}\cdot\mathbf{u}_D)\,\mathbf{u}_D}_{noise\ \ component\ \ \bar{\mathbf{x}}}$$

Linear Orthogonal Projection

$$\mathbf{z} = \mathbf{U}\mathbf{x}$$

$\mathbf{n}$ $\quad \mathbf{n}^T\mathbf{n}=(\mathbf{x}-\mathbf{U}^T\mathbf{z})^T(\mathbf{x}-\mathbf{U}^T\mathbf{z})$



(1)

GMM: $p(\mathbf{z}) = \sum_{k=1}^{K} \pi_k \cdot f_k(\mathbf{z}|\boldsymbol{\theta}_k) = \sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(\mathbf{z} \mid \boldsymbol{\mu}_k, \Sigma_k)$

(2) $p(\mathbf{n}) \sim \mathcal{N}(\mathbf{n} \mid \mathbf{0}, \sigma^2\mathbf{I})$

MovFM: $p(\mathbf{z}) = \sum_{k=1}^{K} \pi_k \cdot f_k(\mathbf{z}|\boldsymbol{\theta}_k) = \sum_{k=1}^{K} \pi_k \cdot \mathcal{C}_M(|\boldsymbol{\mu}_k|) \cdot e^{\mathbf{z}\cdot\boldsymbol{\mu}_k}$

(3) $p(\mathbf{x}) = |\hat{\mathbf{U}}^{-1}| \cdot p(\mathbf{z}) \cdot p(\mathbf{n})$

(4) s.t. $\mathbf{U}\mathbf{U}^T = \mathbf{I}$

Finite Mixture Models

- A choice of any finite mixture model:

  e.g., finite mixture of **exponential family** distributions

$$p(\mathbf{z}) = \sum_{k=1}^{K} \pi_k \cdot f_k(\mathbf{z}|\boldsymbol{\theta}_k)$$

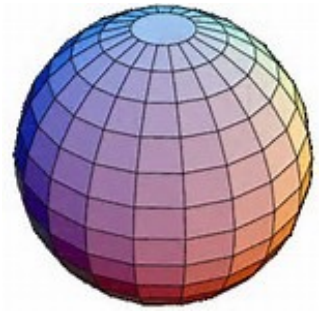
- Examples:
  - Gaussian mixture models (GMMs)
  - Mixture of ***von Mises-Fisher*** distributions (movFMs)

Mixture of von Mises-Fisher (movFM)

- **von Mises-Fisher (vFM)** distribution: a generalized normal distribution defined on high-dimension spherical surface.

$$f(\mathbf{z}) = \mathcal{C}_M(|\boldsymbol{\mu}|) \cdot e^{\mathbf{z}\cdot\boldsymbol{\mu}} \quad \text{s. t.} \quad |\mathbf{z}| = 1$$

with $$\mathcal{C}_M(\kappa) = \frac{\kappa^{M/2-1}}{(2\pi)^{M/2} I_{M/2-1}(\kappa)}$$



- **Mixture of von Mises-Fisher (movFM)** model:

$$p(\mathbf{z}) = \sum_{k=1}^{K} \pi_k \cdot f_k(\mathbf{z}|\boldsymbol{\theta}_k) = \sum_{k=1}^{K} \pi_k \cdot \mathcal{C}_M(|\boldsymbol{\mu}_k|) \cdot e^{\mathbf{z}\cdot\boldsymbol{\mu}_k}$$

Maximum Likelihood Estimation for HOPE

- The data distribution based on HOPE:

$$p(\mathbf{x}) = |\hat{\mathbf{U}}^{-1}| \cdot p(\mathbf{z}) \cdot p(\mathbf{n})$$

- The log-likelihood function of HOPE:

$$\begin{aligned}
\mathcal{L}(\mathbf{U}, \boldsymbol{\Theta}, \sigma \mid \mathbf{X}) &= \sum_{n=1}^{N} \ln \Pr(\mathbf{x}_n) = \sum_{n=1}^{N} \Big[\ln \Pr(\mathbf{z}_n) + \ln \Pr(\mathbf{n}_n) \Big] \\
&= \underbrace{\sum_{n=1}^{N} \ln \left(\sum_{k=1}^{K} \pi_k \cdot f_k(\mathbf{U}\mathbf{x}_n | \boldsymbol{\theta}_k) \right)}_{\mathcal{L}_1(\mathbf{U},\boldsymbol{\Theta})} + \underbrace{\sum_{n=1}^{N} \ln \left(\mathcal{N}(\mathbf{n}_n \mid \mathbf{0}, \sigma^2 \mathbf{I}) \right)}_{\mathcal{L}_2(\mathbf{U},\sigma)}
\end{aligned}$$

- Maximum Likelihood Estimation for HOPE

$$\{\mathbf{U}^*, \boldsymbol{\Theta}^*, \sigma^*\} = \arg\max_{\mathbf{U},\boldsymbol{\Theta},\sigma} \quad \mathcal{L}(\mathbf{U}, \boldsymbol{\Theta}, \sigma \mid \mathbf{X})$$

subject to the orthogonal constraint:

$$\mathbf{U}\mathbf{U}^T = \mathbf{I}.$$

Dealing with Orthogonal Constraints

- Casting the constraint as a penalty term:

$$\mathcal{D}(\mathbf{U}) = \sum_{i=1}^{M} \sum_{j=i+1}^{M} \frac{|\mathbf{u}_i \cdot \mathbf{u}_j|}{|\mathbf{u}_i| \cdot |\mathbf{u}_j|}$$

- Converting into an unconstrained optimization:

$$\{\mathbf{U}^*, \mathbf{\Theta}^*, \sigma^*\} = \arg\max_{\mathbf{U},\mathbf{\Theta},\sigma} \left[\mathcal{L}(\mathbf{U}, \mathbf{\Theta}, \sigma \mid \mathbf{X}) - \beta \cdot \mathcal{D}(\mathbf{U}) \right]$$

SGD-based Unsupervised Learning of HOPE based on Maximum Likelihood Estimation

Algorithm 1 SGD-based Maximum Likelihood Learning Algorithm for HOPE

randomly initialize $\mathbf{u}_i$ $(i = 1, \cdots, M)$, π_k and $\boldsymbol{\theta}_k$ $(k = 1, \cdots, K)$

for *epoch* $= 1$ **to** T **do**

  for *minibatch* $\mathbf{X}$ **in** training set **do**

$$\mathbf{U} \leftarrow \mathbf{U} + \epsilon \cdot \left(\frac{\partial \mathcal{L}_1(\mathbf{U}, \boldsymbol{\Theta})}{\partial \mathbf{U}} + \frac{\partial \mathcal{L}_2(\mathbf{U}, \sigma)}{\partial \mathbf{U}} - \beta \cdot \frac{\partial \mathcal{D}(\mathbf{U})}{\partial \mathbf{U}} \right)$$

$$\boldsymbol{\theta}_k \leftarrow \boldsymbol{\theta}_k + \epsilon \cdot \frac{\partial \mathcal{L}_1(\mathbf{U}, \boldsymbol{\Theta})}{\partial \boldsymbol{\theta}_k} \quad (\forall k)$$

$$\pi_k \leftarrow \pi_k + \epsilon \cdot \frac{\partial \mathcal{L}_1(\mathbf{U}, \boldsymbol{\Theta})}{\partial \pi_k} \quad (\forall k)$$

$$\sigma^2 \leftarrow \frac{1}{N(D-M)} \sum_{n=1}^{N} \mathbf{n}_n^T \mathbf{n}_n$$

$$\pi_k \leftarrow \frac{\pi_k}{\sum_j \pi_j} \quad (\forall k) \quad \text{and} \quad \mathbf{u}_i \leftarrow \frac{\mathbf{u}_i}{|\mathbf{u}_i|} \quad (\forall i)$$

  end for

end for

Related Work

- ***HOPE*** may be viewed as a combination of generalized PCA with generative model (finite mixture model) …
- Principal Component Analysis (PCA) *(Pearson, 1901; Hotelling, 1933, 1936)*
- *Linear Discriminant Analysis (LDA) (Fisher, 1936)*
- Probabilistic PCA (PPCA) *(Tipping & Bishop, 1999a)*
- Mixture of PPCA *(Tipping & Bishop, 1999b)*
- Heteroscedastic LDA (HLDA/HDA) *(Kumar & Andreous, 1998)*

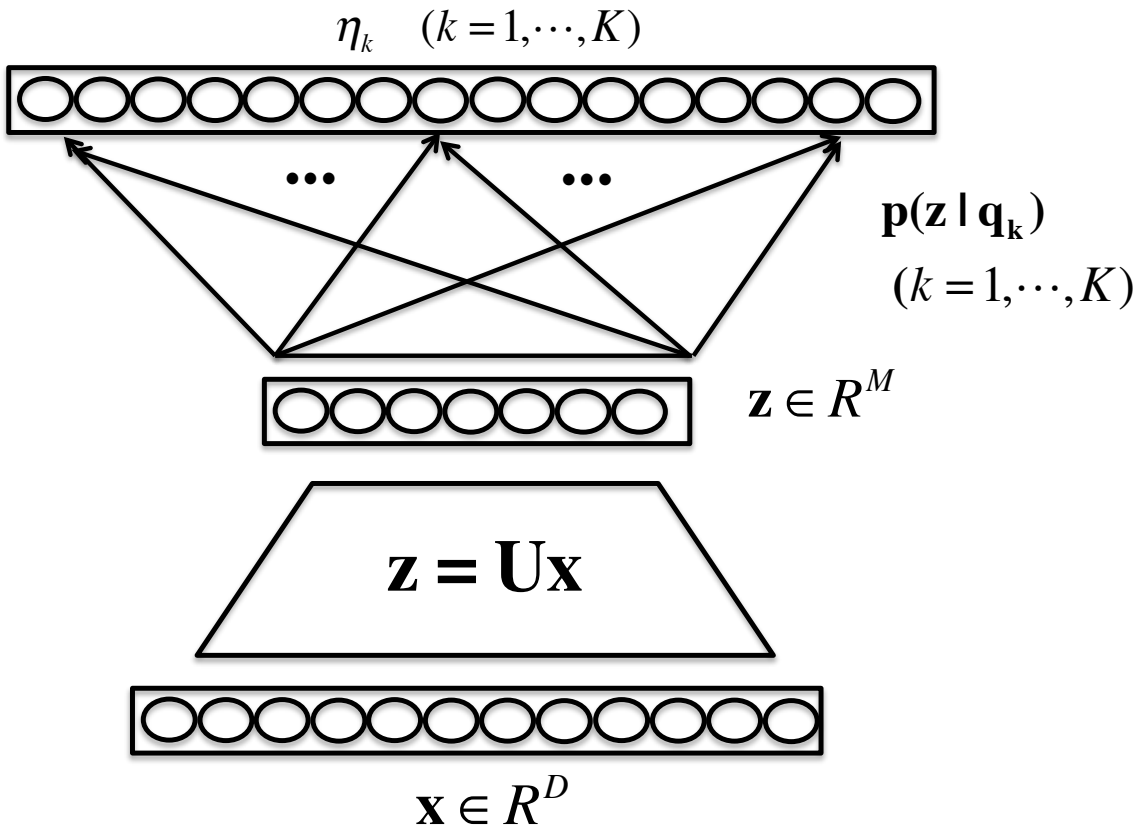
Outline

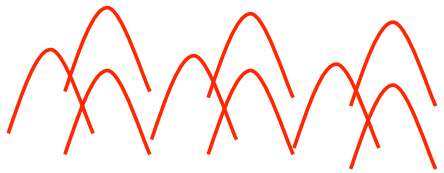
- Introduction
- Hybrid Orthogonal Projection and Estimation (HOPE)
- **Link HOPE to Neural Networks (NN) and Deep Learning**
  - **Unsupervised Learning of NN under HOPE**
  - **Supervised Learning of NN under HOPE**
  - **HOPE for Deep Learning**
- Experiments on MNIST and TIMIT
- Conclusions

HOPE as two-layer network

- A HOPE model may be represented as a two-layer network:

$$\eta_k = \ln\left(\pi_k \cdot f_k(\mathbf{z}|\boldsymbol{\theta}_k)\right) = \ln \pi_k + \ln \mathcal{C}_M(|\boldsymbol{\mu}_k|) + \mathbf{z} \cdot \boldsymbol{\mu}_k$$





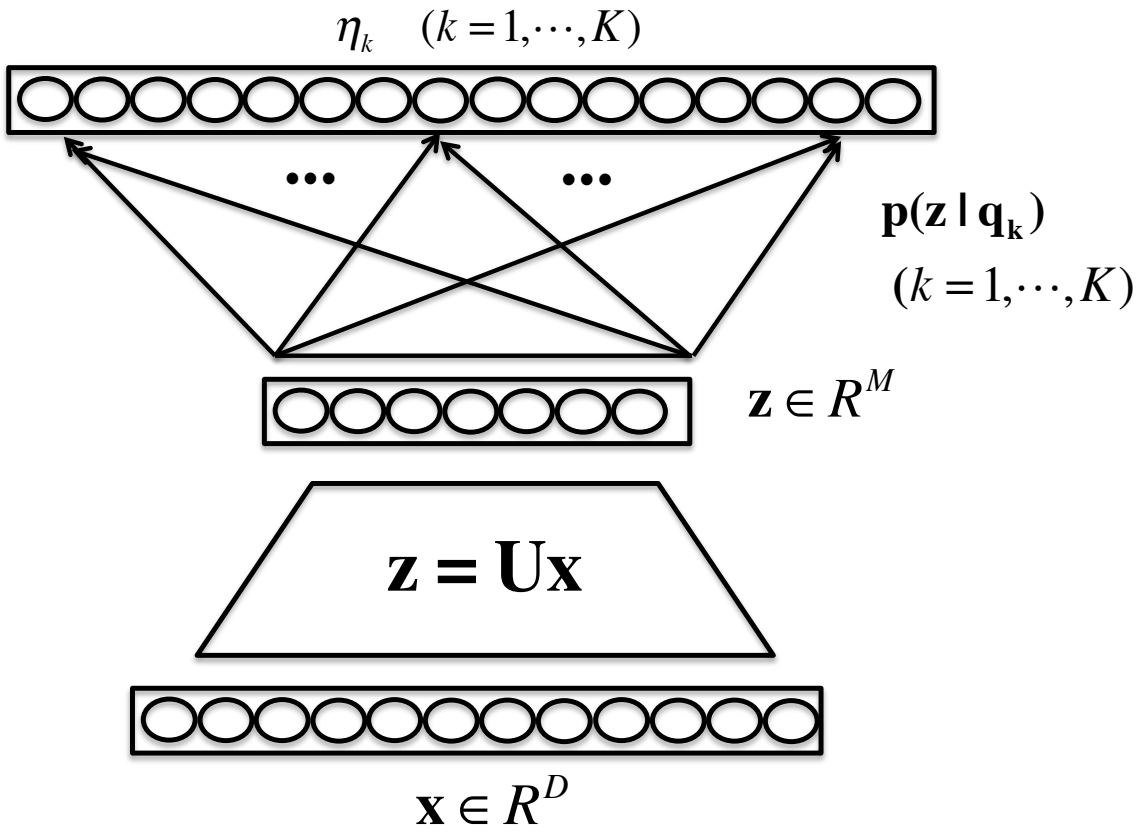
$$\ln p(\mathbf{z}) = \ln \sum_{k=1}^{K} \exp(\eta_k)$$

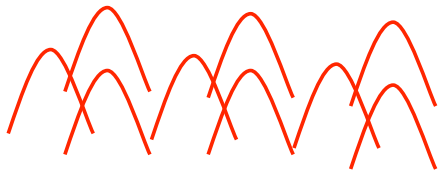
HOPE as Two-layer Networks

- A HOPE model may be represented as a two-layer network:

$$\eta_k = \max\left(0, \ln\left(\pi_k \cdot f_k(\mathbf{z}|\boldsymbol{\theta}_k)\right) - \varepsilon\right)$$

$$= \max\left(0, \ln \pi_k + \ln \mathcal{C}_M(|\boldsymbol{\mu}_k|) + \mathbf{z} \cdot \boldsymbol{\mu}_k - \varepsilon\right)$$

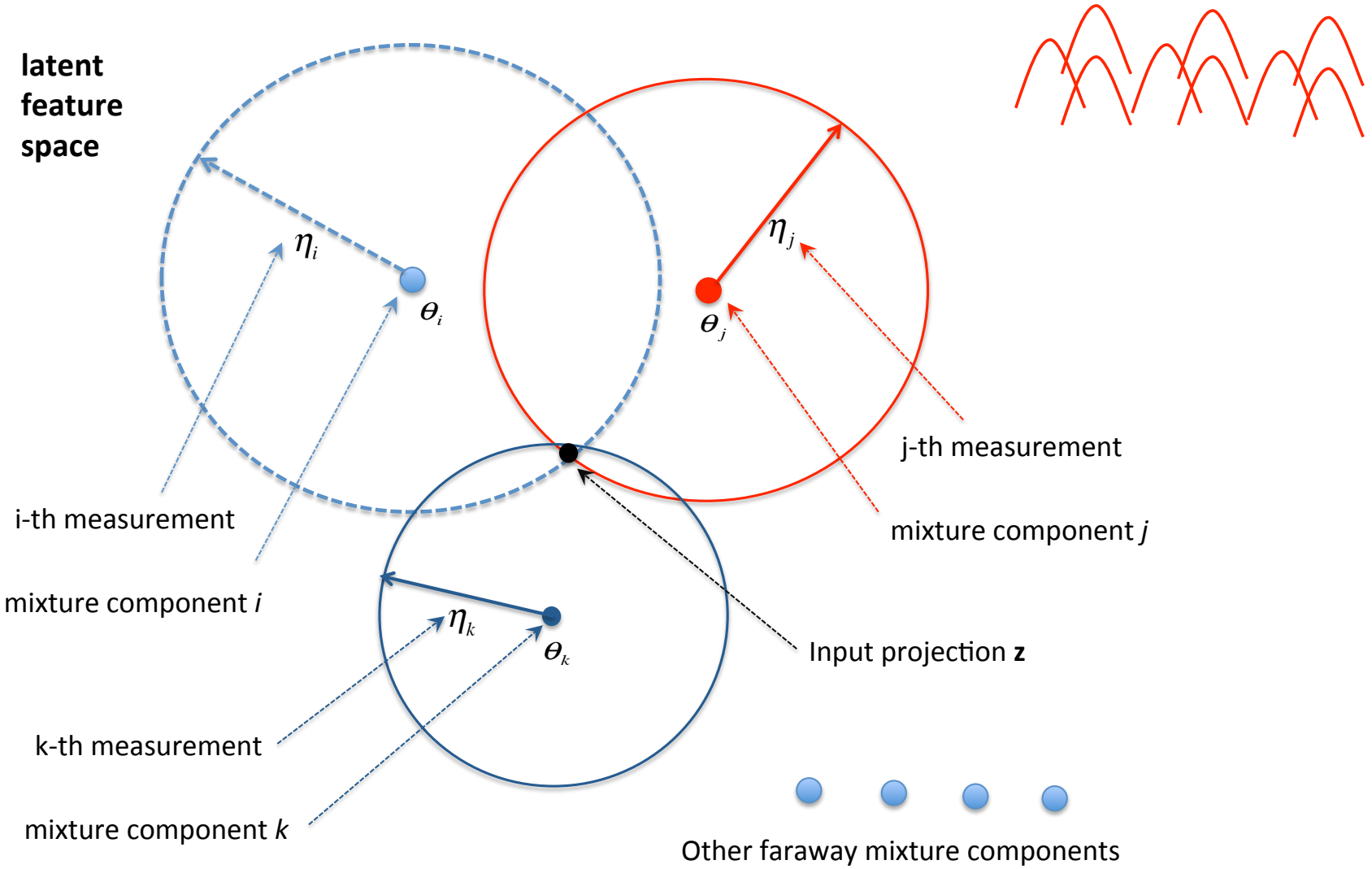




$$\ln p(\mathbf{z}) \approx \varepsilon + \ln \sum_{k=1}^{K} \exp(\eta_k)$$

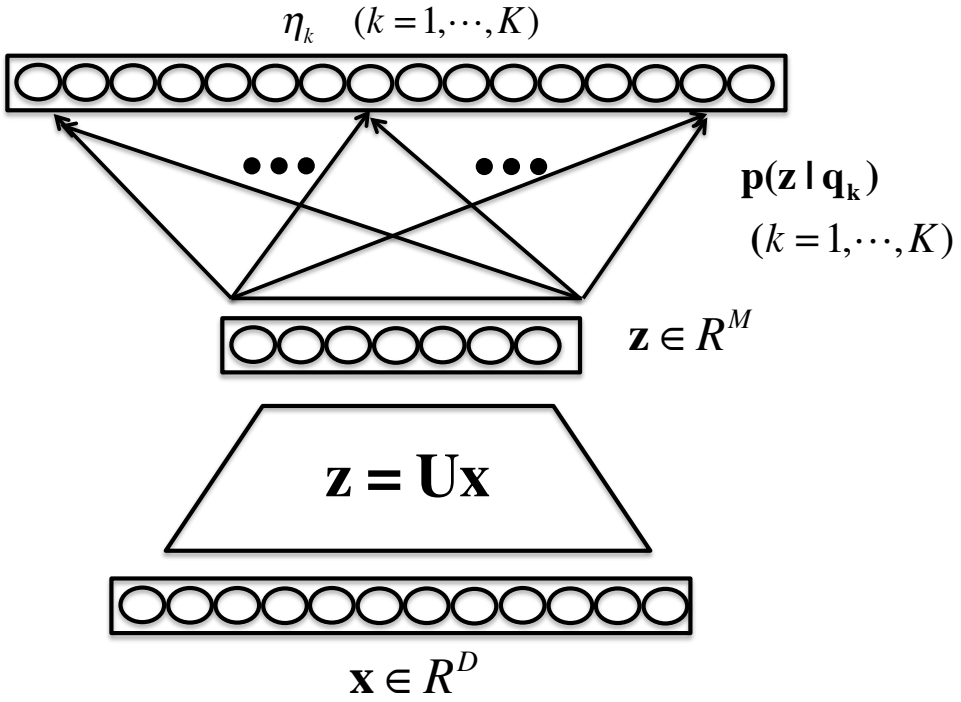
HOPE Features as Trilateration

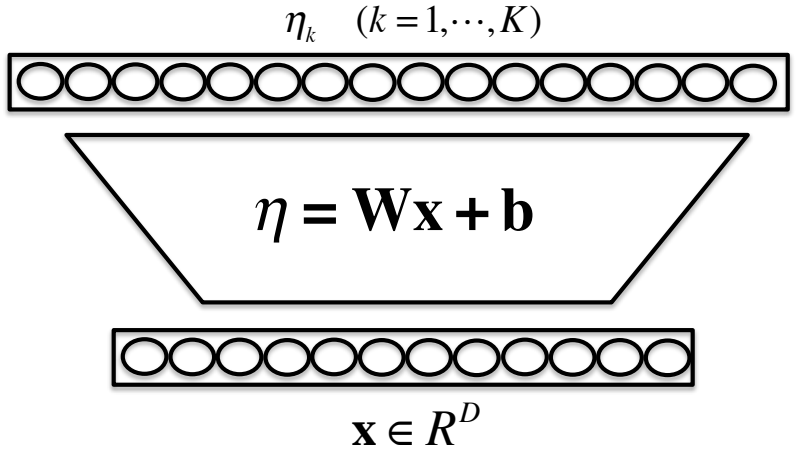
- **Interpreting HOPE features as a trilateration problem in the latent feature space**



Link HOPE to Neural Networks

- **A HOPE model = an NN hidden layer**





$$\mathbf{w}_k = \mathbf{U}^T \boldsymbol{\mu}_k$$

$$b_k = \ln \pi_k + \ln \mathcal{C}_M(|\boldsymbol{\mu}_k|) - \varepsilon$$

Why NNs can handle almost any types of data input?

Unsupervised Learning of NNs with HOPE

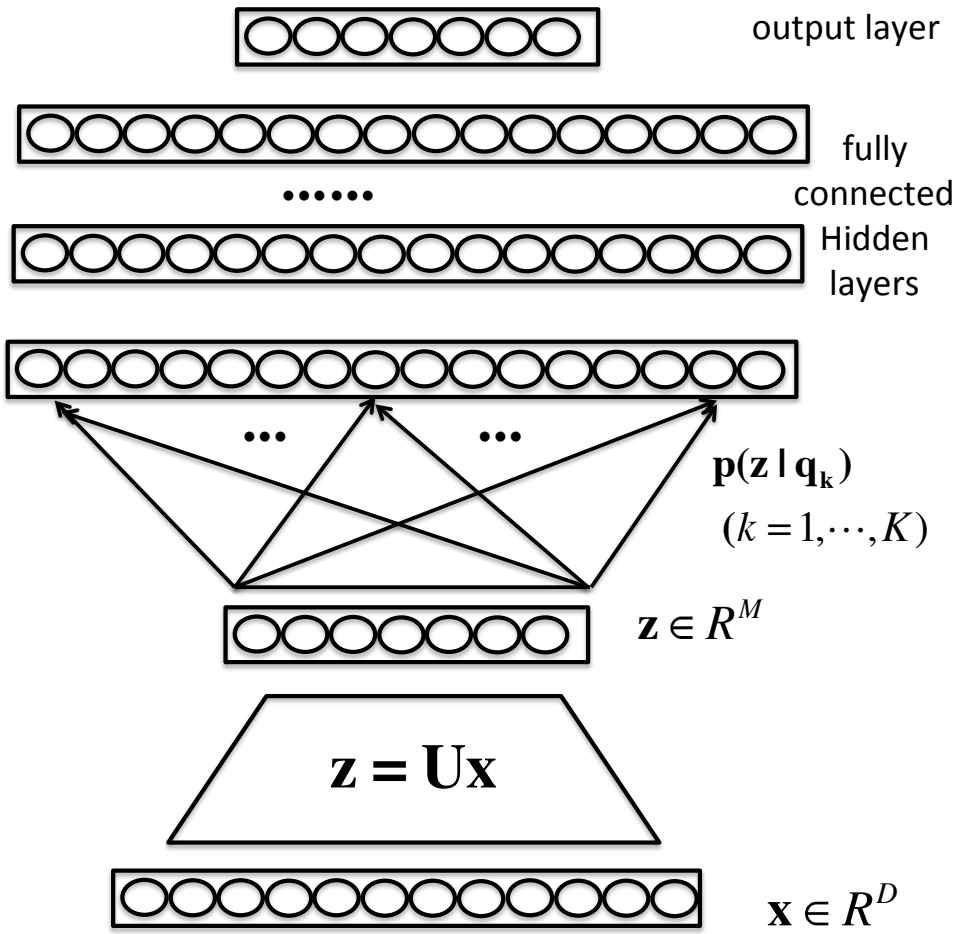
- View each hidden layer as a HOPE model
- Unsupervised learning of the HOPE model based on the maximum likelihood estimation
- Merge to generate the NN hidden layer (if you wish)
- Layer-wise learning for a deep structure
- Similar to the Hebbian learning *(Hebb, 1949)*, but with a well-defined objective function

Supervised Learning of NN with HOPE

- Reformulate each hidden layer as two HOPE layers:
  - A linear orthogonal projection layer (choose M)
  - A movMF layer
- Use other discriminative learning criterion, like cross-entropy …
- **SGD**: backpropagation + **orthogonal constraints**
- Related to the low-rank weight matrix factorization in speech recognition (*Sainath et. al., 2013; Xue et. al., 2013*).

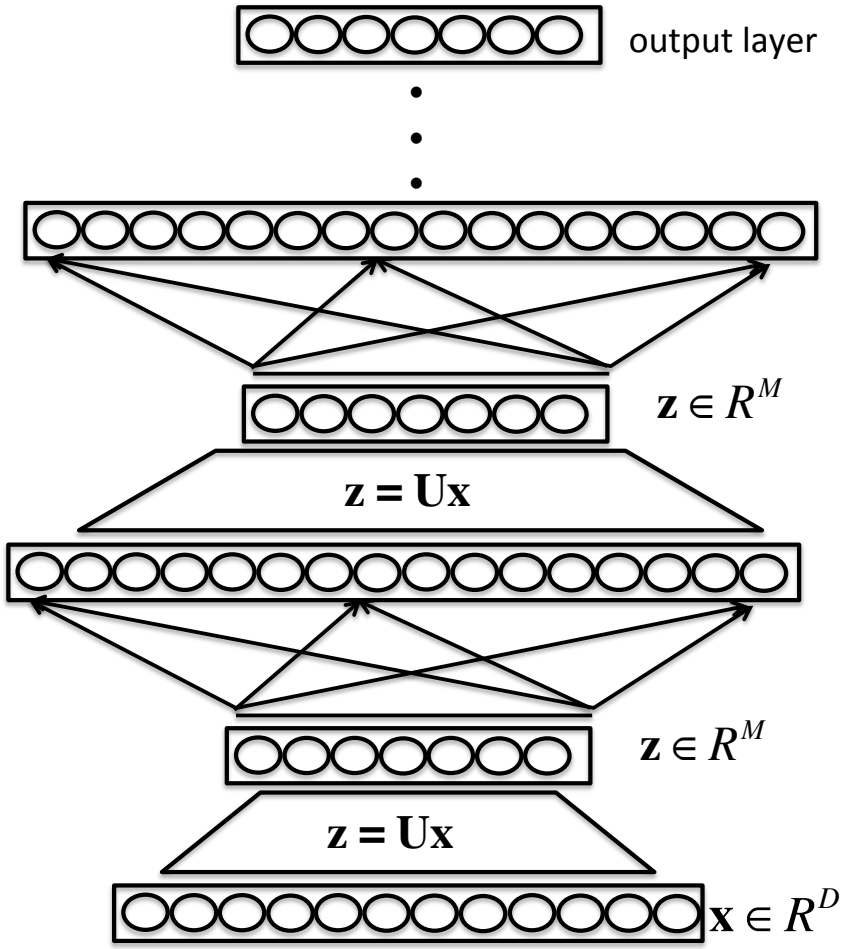
HOPE for Deep Learning (I)

- **Use HOPE to learn deep neural networks as feature extractors**
- **A HOPE + DNN**



HOPE for Deep Learning (II)

- **Using HOPE to learn deep neural networks: stacking multiple HOPE models**



Outline

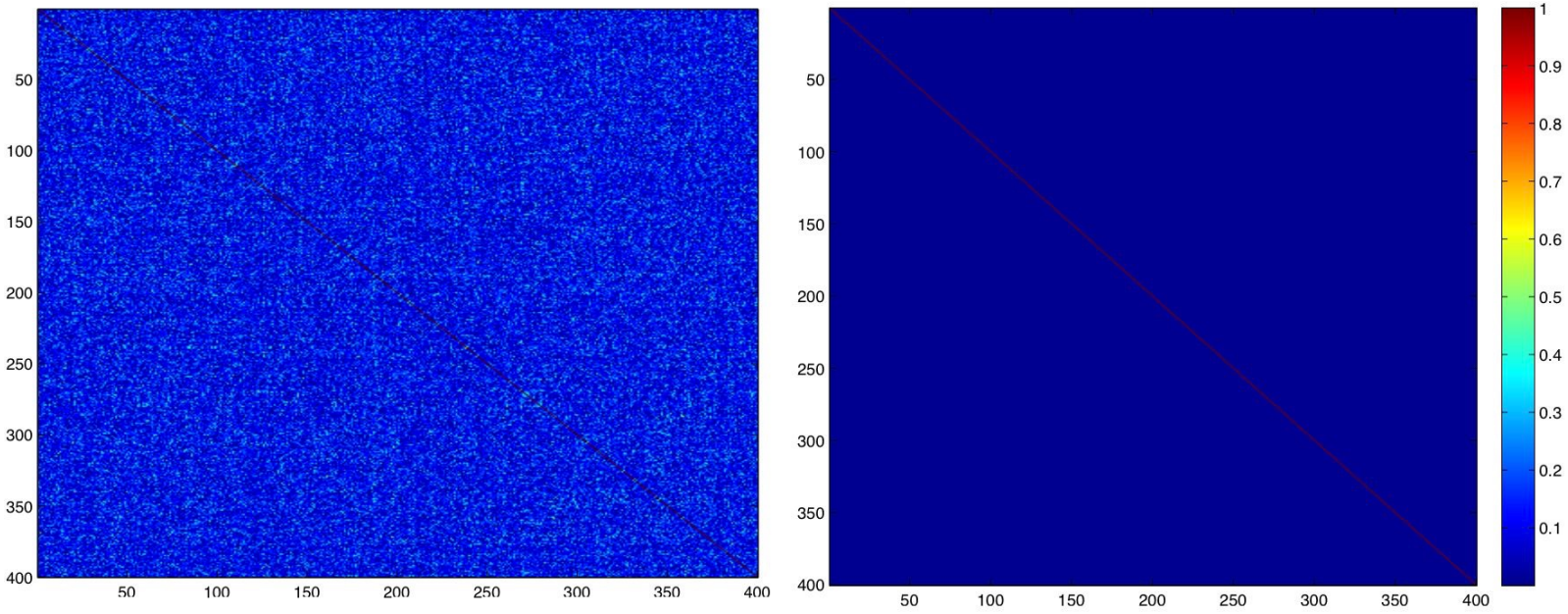
- Introduction
- Hybrid Orthogonal Projection and Estimation (HOPE)
- Link HOPE to Neural Networks (NN) and Deep Learning
  - Unsupervised Learning of NN under HOPE
  - Supervised Learning of NN under HOPE
  - HOPE for Deep Learning
- **Experiments on MNIST and TIMIT**
- Conclusions

Experiments: MNIST

- The standard MNIST data set *(LeCun et al., 1998)*
- 60000 training images; 10000 test images
- Each image is 28 x 28 pixel greyscale images of handwriting digits 0-9
- **Supervised** learning: use whole images as NN input
- **Unsupervised** or **semi-supervised** learning: use 6x6 patches to learn feature representations

Supervised Learning of NNs (I): MNIST

- Supervised learning of NNs with or without HOPE
- With HOPE: imposing orthogonal constraints in BP



Supervised Learning of NNs (II): MNIST

- Use whole MNIST images as NN inputs
- With or without imposing orthogonal constraints for the linear projection layer

K=	1k	2k	5k
Baseline: 784-K-10	**1.49**	**1.35**	**1.28**
HOPE1: 784-[200-K]-10	**1.21**	**1.20**	**1.17**
HOPE2: 784-[400-K]-10	**1.19**	**1.23**	**1.25**
Linear1: 784-(200-K)-10	**1.52**	**1.50**	**1.54**
Linear2: 784-(400-K)-10	**1.53**	**1.52**	**1.49**

Supervised Learning of NNs (III): MNIST

- Use whole MNIST images as NN inputs
- Supervised learning of shallow NNs with HOPE
- Without data augmentation; without CNN

Using dropout	1k	2k	5k
Baseline: 784-K-10	**1.05**	**1.01**	**1.01**
HOPE1: 784-[200-K]-10	**0.99**	**0.85**	**0.89**
HOPE2: 784-[400-K]-10	**0.86**	**0.86**	**0.85**

Supervised Learning of NN (IV): MNIST

- Use whole MNIST images as NN inputs
- Supervised learning of deep NN with HOPE
- Without data augmentation; without using CNN

Model	no dropout	dropout
DNN: 784-1200-1200-10	1.25	0.92
HOPE+NN: 784 [400-1200]-1200-10	0.99	0.82
HOPE*2: 784-[400-1200]-[400-1200]-10	0.97	0.81

Unsupervised Feature Learning (I): MNIST

- Use 400,000 MNIST image patches (6x6) to unsupervised learn feature extractors
- Average features in four quadrants *(Coates et al. 2011)*
- Feed to a post-stage classifier (linear SVM) supervised learned from data and labels

1. **Kmeans** *(Coates et al. 2011)* **: k-means to learn K centroids,**

$$f_k(x) = \max(0, \ |x - \mu_k| - \varepsilon)$$

2. **Spkmeans** *(Coates et al. 2011)***: spk-means to learn K centroids,**

$$f_k(x) = \max(0, \ x^T \mu_k - \varepsilon)$$

3. **movMF: use EM to learn movMF,** $f_k(x) = \max(0, \ \ln \pi_k + \ln(C_k) + z^T \mu_k - \varepsilon)$
4. **PCA-movMF: PCA (M=20) + EM learned movMF**
5. **HOPE-movMF: HOPE model with movMF (M=20)**

Unsupervised Feature Learning (II): MNIST

K=	400	800	1200	1600
K-means	1.41	1.31	1.16	1.13
spkmeans	1.09	0.90	0.86	0.81
movMF	0.89	0.82	0.81	0.84
PCA-movMF	0.87	0.75	0.73	0.74
HOPE-movMF	0.76	0.71	0.64	0.67

Unsupervised learned features with HOPE work well with a separately trained simple classifier in the post-stage.

Semi-supervised Learning: MNIST

- Use 400,000 MNIST image patches (6x6) from all training data to unsupervised learn feature extractors

- Learn a post-stage classifier using only a part of training data

Method	1k	2k	5k	60k (all)
CDBN (Lee et al 2009)	2.62	2.13	1.59	0.82
Pixel + DNN	8.32	4.71	3.20	0.92
Pixel + HOPE-DNN	7.21	4.02	2.60	0.82
USL + SVM	2.91	2.38	1.47	0.71
USL + DNN	2.83	1.99	1.03	0.43
USL + HOPE-DNN	2.46	1.70	0.90	0.40

TIMIT: speech recognition

- TIMIT: a small phoneme recognition task
- Supervised learning of NNs for acoustic modelling in speech recognition

model	net architecture	FACC (%)	PER(%)
NN	1845-10240-183	61.45	23.85
HOPE-NN	1845-[256-10240]-183	62.11	**23.04**
DNN	1845-3*2048-183	63.13	22.37
HOPE-DNN	1845-[512-2048]-2*2046-183	63.55	**21.59**

- HOPE may help to learn a strong shallow NN.
- HOPE is beneficial to deep NNs as well.

Conclusion

- **HOPE** is a generative model combining linear orthogonal projection (feature selection) and finite mixture models (data modelling).
- Links between HOPE and NN: each NN hidden layer = a HOPE
- NN can be learned under the HOPE framework, either supervised or **unsupervised** ways.
- NN features may be interpreted as a ***trilateration*** problem in the latent feature space.
- ***Strong performance on MNIST*** for all supervised, unsupervised and semi-supervised tasks.

More on HOPE

- MNIST matlab codes downloading from:

http://wiki.eecs.yorku.ca/lab/MLL/projects:hope:start

- A full technical report:

http://arxiv.org/abs/1502.00702