AGEM: Solving Linear Inverse Problems via Deep Priors and Sampling

Bichuan Guo[1], Yuxing Han[2], Jiangtao Wen[1]

[1]Tsinghua University, [2]South China Agricultural University

Tsinghua University

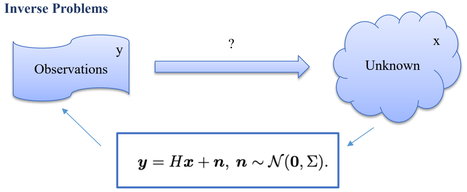
NEURAL INFORMATION PROCESSING SYSTEMS

Abstract

In this paper we propose to use a denoising autoencoder (DAE) prior to simultaneously solve a linear inverse problem and estimate its noise parameter. Existing DAE-based methods estimate the noise parameter empirically or treat it as a tunable hyper-parameter. We instead propose *autoencoder guided EM*, a probabilistically sound framework that performs Bayesian inference with intractable deep priors. We show that efficient posterior sampling from the DAE can be achieved via Metropolis-Hastings, which allows the Monte Carlo EM algorithm to be used. We demonstrate competitive results for signal denoising, image deblurring and image devignetting. Our method is an example of combining the representation power of deep learning with uncertainty quantification from Bayesian statistics.

RELATED WORK

Inverse Problems



Discriminative learning: requires task-specific training

- Task 1 – training 1 – model 1
- Task 2 – training 2 – model 2

Learned knowledge is non-trivial to transfer.

Model-based learning: require analytic models

- Behavioral model: noise, blur kernel, etc.
- Inductive bias: TV-L1, CSF, etc.

Method fits into the Bayesian paradigm:

$$\log \Pr(\boldsymbol{x} \mid \boldsymbol{y}, \Sigma) = \log \Pr(\boldsymbol{y} \mid \boldsymbol{x}, \Sigma) + \log \Pr(\boldsymbol{x}) + \textit{const.}$$

Drawback: requires handcrafting and often involves manual hyper-param tuning.

	Pros	Cons	Examples
analytic	tractability interpretability	bias prone	total variation
data-driven	scalability less handcrafting	intractable	databases, neural nets

A special data-driven prior with good tractability: **denoising autoencoder**

$$r^*(\boldsymbol{x}) = \boldsymbol{x} + \sigma_{\text{tr}}^2 \nabla_{\boldsymbol{x}} \log \Pr(\boldsymbol{x}) + o(\sigma_{\text{tr}}^2),\ \text{as } \sigma_{\text{tr}} \to 0,$$

OBJECTIVES AND METHODS

Let the data speak for itself

1. Choose model-based learning over discriminative learning
   - Free ourselves from tedious re-training
2. Use data-driven prior over analytic prior
   - Free ourselves from handcrafting priors
3. Automatic model tuning
   - Free ourselves from tedious hyper-param searching

Determining hyper-parameters

- Perform MLE on observations.
- Clean data "x" are latent variables.
- The canonical way to perform MLE with latent variables is to use the EM (expectation maximization) algorithm.

EM is intractable

$$\begin{aligned}\mathcal{Q}(\Sigma, \Sigma^{(\tau)}) &= \mathbb{E}_{\boldsymbol{x} \sim \Pr(\boldsymbol{x} \mid \boldsymbol{y}, \Sigma^{(\tau)})} \log \Pr(\boldsymbol{y}, \boldsymbol{x} \mid \Sigma) \\ &= \mathbb{E}_{\boldsymbol{x} \sim \Pr(\boldsymbol{x} \mid \boldsymbol{y}, \Sigma^{(\tau)})} \log \Pr(\boldsymbol{y} \mid \boldsymbol{x}, \Sigma) + \log \Pr(\boldsymbol{x}),\end{aligned}$$

No free lunch: we cannot expect to enjoy the advantages of data-driven priors, while demanding the tractability of analytic priors!

$$\Pr(\boldsymbol{x} \mid \boldsymbol{y}, \Sigma^{(\tau)}) = Z^{-1} \Pr(\boldsymbol{y} \mid \boldsymbol{x}, \Sigma^{(\tau)}) \Pr(\boldsymbol{x}),$$

Sampling methods

Monte Carlo EM algorithm: no tractable posterior? A group of samples will also do.

How to sample from the posterior?

- The first term is easy (analytic)
- The second term is still intractable. Our only lead is its score function:

$$r^*(\boldsymbol{x}) = \boldsymbol{x} + \sigma_{\text{tr}}^2 \nabla_{\boldsymbol{x}} \log \Pr(\boldsymbol{x}) + o(\sigma_{\text{tr}}^2),\ \text{as } \sigma_{\text{tr}} \to 0,$$

We adopt Metropolis-Hastings algorithms to sample from the posterior:

$$\alpha = \frac{\Pr(\boldsymbol{x}^* \mid \boldsymbol{y}, \Sigma^{(\tau)}) q(\boldsymbol{x}^{(i)} \mid \boldsymbol{x}^*)}{\Pr(\boldsymbol{x}^{(i)} \mid \boldsymbol{y}, \Sigma^{(\tau)}) q(\boldsymbol{x}^* \mid \boldsymbol{x}^{(i)})},$$

$$\begin{aligned}\log \alpha &= \log \Pr(\boldsymbol{x}^* \mid \boldsymbol{y}, \Sigma^{(\tau)}) - \log \Pr(\boldsymbol{x}^{(i)} \mid \boldsymbol{y}, \Sigma^{(\tau)}) \\ &= \log \Pr(\boldsymbol{y} \mid \boldsymbol{x}^*, \Sigma^{(\tau)}) - \log \Pr(\boldsymbol{y} \mid \boldsymbol{x}^{(i)}, \Sigma^{(\tau)}) + \log \Pr(\boldsymbol{x}^*) - \log \Pr(\boldsymbol{x}^{(i)}) \\ &= \left(H \frac{\boldsymbol{x}^{(i)} + \boldsymbol{x}^*}{2} - \boldsymbol{y}\right)^\top {\Sigma^{(\tau)}}^{-1} H(\boldsymbol{x}^{(i)} - \boldsymbol{x}^*) + \boxed{\log \Pr(\boldsymbol{x}^*) - \log \Pr(\boldsymbol{x}^{(i)})},\end{aligned}$$

The score function is sufficient:

$$\begin{aligned}\boxed{\log \Pr(\boldsymbol{x}^*) - \log \Pr(\boldsymbol{x}^{(i)})} &\approx \nabla_{\boldsymbol{x}} \log \Pr(\boldsymbol{x})\big|_{\boldsymbol{x}^{(i)}} \cdot (\boldsymbol{x}^* - \boldsymbol{x}^{(i)}) \\ &\approx \sigma_{\text{tr}}^{-2} (r(\boldsymbol{x}^{(i)}) - \boldsymbol{x}^{(i)})^\top (\boldsymbol{x}^* - \boldsymbol{x}^{(i)}),\end{aligned}$$

To accelerate Markov chain mixing, we use Langevin-adjusted MH:

$$q_{\text{MALA}}(\boldsymbol{x} \mid \boldsymbol{x}^{(i)}) = \mathcal{N}(\boldsymbol{x}^{(i)} + \tfrac{1}{2}\sigma_{\text{prop}}^2 \nabla_{\boldsymbol{x}} \log \Pr(\boldsymbol{x} \mid \boldsymbol{y}, \Sigma^{(\tau)})\big|_{\boldsymbol{x}^{(i)}}, \sigma_{\text{prop}}^2 I).$$

This proposal distribution is also straightforward to evaluate:

$$\begin{aligned}\nabla_{\boldsymbol{x}} \log \Pr(\boldsymbol{x} \mid \boldsymbol{y}, \Sigma^{(\tau)}) &= \nabla_{\boldsymbol{x}} \log \Pr(\boldsymbol{y} \mid \boldsymbol{x}, \Sigma^{(\tau)}) + \nabla_{\boldsymbol{x}} \log \Pr(\boldsymbol{x}) \\ &\approx H^\top {\Sigma^{(\tau)}}^{-1} (\boldsymbol{y} - H\boldsymbol{x}) + \sigma_{\text{tr}}^{-2} (r(\boldsymbol{x}) - \boldsymbol{x}).\end{aligned}$$

METHODS AND RESULTS

Algorithm 1 Estimate latent signal $\boldsymbol{x}$ and noise level Σ with the proposed methods AGEM and AGEM-ADMM. τ is the EM iteration number, initialized as 0. $\Sigma^{(1)}$ is initialized as $\sigma_{\text{tr}}^2 I$.

1: Train a DAE with quadratic loss and noise $\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma_{\text{tr}}^2 I)$
2: **repeat** $\tau \leftarrow \tau + 1$
3: Initialization: If $\tau = 1$, $\boldsymbol{x}_\tau^{(1)} \leftarrow \boldsymbol{0}$, otherwise $\boldsymbol{x}_\tau^{(1)} \leftarrow \boldsymbol{x}_{\tau-1}^{(n_{\text{MH}})}$
4: E-step: Draw n_{MH} samples $\{\boldsymbol{x}_\tau^{(i)}\}_{i=1}^{n_{\text{MH}}}$ with MALA, discard the first ½ samples as burn-in
5: M-step: Use $\{\boldsymbol{x}_\tau^{(i)}\}_{i=n_{\text{MH}}/5}^{n_{\text{MH}}}$ to compute $\Sigma^{(\tau+1)}$
6: **until** $\tau = n_{\text{EM}}$
7: [AGEM] Compute $\hat{\boldsymbol{x}} \leftarrow$ average of $\{\boldsymbol{x}_\tau^{(i)}\}_{i=n_{\text{MH}}/5}^{n_{\text{MH}}}$; **return** $(\hat{\boldsymbol{x}}, \Sigma^{(n_{\text{EM}})})$
8: [AGEM-ADMM] Use ADMM and noise level $\Sigma^{(n_{\text{EM}})}$ to compute $\hat{\boldsymbol{x}}$; **return** $(\hat{\boldsymbol{x}}, \Sigma^{(n_{\text{EM}})})$

Results

Signal denoising. Consider 50-dimensional signals lying on a latent 2D manifold, and corrupted by isotropic Gaussian noise. The DAE is a multilayer perceptron with ReLU activations and 3 hidden layers, each containing 2000 neurons.

Table 1: Signal denoising, average RMSE of the test set. Standard deviations are in parentheses, estimated noise levels are in square brackets. Best performances are in **bold**. (*All values are in* 10^{-2}).

σ_n:	1.00		2.00		3.00		4.00	
Method	mean	std.	mean	std.	mean	std.	mean	std.
DAEP+NE [3]	0.73	(0.10)	0.98	(0.13)	1.16	(0.20)	1.31	(0.27)
ADMM+NE [35]	0.37	(0.28)	0.60	(0.36)	0.93	(0.55)	1.59	(3.49)
DMSP [4]	0.50	(0.22)	0.74	(0.29)	0.99	(0.45)	1.36	(0.95)
	[1.62]	(0.14)	[2.19]	(0.22)	[3.07]	(0.35)	[4.11]	(0.75)
AGEM	0.51	(0.15)	0.70	(0.25)	**0.86**	(0.39)	**1.16**	(0.64)
	[1.19]	(0.13)	[1.93]	(0.26)	[2.96]	(0.38)	[4.03]	(0.52)
AGEM-ADMM	**0.33**	(0.23)	**0.57**	(0.34)	0.91	(0.53)	1.43	(2.05)

Our best method outperforms all baseline methods significantly statistically ($p < 0.05$), and our estimated noise strength are closer to the true values comparing to DMSP. AGEM-ADMM performs well under small noises. Indeed, since ADMM uses the trained DAE for denoising, it works well if σn is close to the training noise σtr. However, as DMSP overestimates σn especially when σn is small, it misses the "operating region" of ADMM, leading to ADMM+NE's inferior performance.

Ablation studies. We study the behavior of AGEM in detail under the settings of the previous experiment. We explore different σ_{prop}, initial noise levels Σ, strategies to construct the recovered x, and compare MALA with the symmetric Gaussian proposal. We set the test noise level $\sigma_n = 0.03$, all hyper-parameters remain unchanged except for the hyper-parameter being studied.

Table 2: Ablation study, average RMSE of the test set. Noise level is $\Sigma = \sigma_n^2 I$, where $\sigma_n = 3.00 \times 10^{-2}$. Estimated noise levels are in square brackets. (*All values are in* 10^{-2}).

	mean	std.	mean	std.	mean	std.	mean	std.	mean	std.
σ_{prop}:	0.01		0.10		1.00		10.0		100	
	34.1	(12.7)	1.20	(0.40)	0.86	(0.39)	does not converge		does not converge	
	[34.2]	(12.7)	[2.87]	(0.40)	[2.96]	(0.38)				
$\Sigma^{(1)}$:	$0.5I$		$1.0I$		$2.0I$		$4.0I$		$8.0I$	
	does not converge		0.86	(0.39)	0.87	(0.40)	0.86	(0.39)	does not converge	
			[2.96]	(0.38)	[2.96]	(0.37)	[2.96]	(0.37)		
misc.	mean		median		last		first		Gaussian	
	0.86	(0.39)	0.87	(0.39)	1.57	(0.38)	1.58	(0.40)	7.61	(4.43)
	[2.96]	(0.38)	[2.96]	(0.37)	[2.96]	(0.38)	[2.96]	(0.38)	[9.51]	(3.82)

- If σ_{prop} is too small, the results are incorrect, as it takes impractically many samples to achieve good mixing. If σ_{prop} is too large, new samples deviate from high density regions, and the algorithm fails to converge as no new samples are accepted.
- As long as the initialization is within a good range, the results are stable. In practice one can try a wide range of initializations to seek convergence.
- The Gaussian proposal gives incorrect results, as it fails to exploit gradient information and is stuck at local maxima.

RESULTS

Image deblurring. We perform image deblurring with the STL-10 unlabeled dataset, which contains 105 colored 96×96 images. The DAE uses the full convolutional, residual architecture, where the input is added to the final layer's output. For testing, images are blurred using a 5 × 5 Gaussian filter with $\sigma = 0.6$.

Table 3: Average PSNR for image deblurring. Estimated noise levels are in square brackets.

σ_n:	0.01		0.02		0.03		0.04	
Method	mean	std.	mean	std.	mean	std.	mean	std.
DAEP+NE [3]	33.13	(1.39)	27.77	(0.89)	25.48	(0.70)	24.30	(0.61)
ADMM+NE [35]	32.43	(3.08)	29.48	(3.16)	27.87	(2.97)	25.78	(3.16)
DMSP [4]	33.60	(2.46)	30.89	(2.14)	28.93	(2.18)	27.40	(2.33)
	[0.017]	(1e-3)	[0.023]	(2e-3)	[0.031]	(3e-3)	[0.041]	(4e-3)
AGEM	**34.79**	(2.00)	**31.42**	(1.81)	**29.47**	(1.92)	28.00	(2.10)
	[0.014]	(1e-3)	[0.021]	(2e-3)	[0.030]	(3e-3)	[0.040]	(3e-3)
AGEM-ADMM	33.75	(2.77)	30.00	(3.20)	28.00	(2.88)	26.05	(3.51)
Hyper-Laplacian [21]	33.28	(0.65)	30.26	(0.40)	29.28	(0.35)	**28.82**	(0.35)
CSF [30]	32.97	(0.68)	29.94	(0.41)	29.02	(0.37)	28.61	(0.36)

AGEM consistently outperforms all baseline methods significantly statistically ($p < 0.01$), and its estimated σn are closer to true values than DMSP. We also compare with some analytic priors. Although these priors are specifically designed for image deconvolution, our generic approach outperforms them except for σn = 0.04, indicating that our trained DAE learns the distribution of natural images well, and DAE-based methods are indeed relevant in practice.

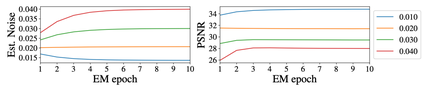


Figure 2: Convergence visualization for image deblurring. Left: average estimated noise level; right: mean PSNR. The legend shows the true noise level σ_n. Stable convergence is quickly reached. Each EM epoch draws 300 MCMC samples.

Image devignetting. Vignetting is a prevalent artifact in photography that brightness attenuates away from the center. We perform image devignetting with the CelebA dataset, The transformation is based on the Kang-Weiss vignetting model.

$$p(r) = \frac{1 - \alpha r}{[1 + (r/f)^2]^2}.$$

Table 4: Average PSNR for image devignetting. Estimated noise levels are in square brackets.

σ_n:	0.015		0.02		0.025		0.03	
Method	mean	std.	mean	std.	mean	std.	mean	std.
DAEP+NE [3]	33.76	(0.71)	31.19	(0.69)	29.16	(0.64)	27.51	(0.58)
ADMM+NE [35]	34.10	(1.62)	32.95	(1.56)	31.60	(1.56)	29.96	(1.68)
DMSP [4]	35.78	(0.99)	34.43	(0.94)	33.26	(0.94)	32.18	(1.03)
	[0.022]	(1e-3)	[0.024]	(1e-3)	[0.027]	(1e-3)	[0.032]	(1e-3)
AGEM	**36.34**	(0.65)	**34.76**	(0.68)	**33.58**	(0.77)	**32.55**	(0.88)
	[0.017]	(1e-3)	[0.020]	(1e-3)	[0.024]	(1e-3)	[0.029]	(1e-3)
AGEM-ADMM	36.16	(1.54)	34.56	(1.53)	32.87	(1.54)	31.07	(1.60)
LIE [23]	29.61	(1.72)	29.43	(1.43)	29.23	(1.16)	29.05	(0.95)
SIVC [47]	29.55	(0.87)	29.44	(0.78)	29.33	(0.71)	29.22	(0.64)

Figure 3: Visual comparison for image devignetting with $\sigma_n = 0.015$. Numbers above the images are: PSNR of the image / average PSNR of the test set (in dB). Zoom in for more details.

CONCLUSIONS

In this paper, we propose a probabilistic framework that uses DAE prior to simultaneously solve linear inverse problems and estimate noise levels, based on the Monte Carlo EM algorithm. We show that during the Monte Carlo E-step, efficient posterior sampling can be performed, as the reconstruction error of DAE captures the gradient of log prior. Our framework allows us to use deep priors trained by unsupervised learning for a wide range of tasks, including signal denoising, image deblurring and image devignetting. Experimental results show that our method outperforms the previous state-of-the-art DAE-based methods. However, this study is not without limitations. Since our method is based on sampling, it usually takes several times longer than non-sampling-based methods to achieve stable convergence. A possible direction for future research is to extend our framework to nonlinear inverse problems. We are also considering using other forms of deep priors.

REFERENCES

[1] G. C. G. Wei and M. A. Tanner, "A Monte Carlo Implementation of the EM Algorithm and the Poor Man's Data Augmentation Algorithms," p. 7.

[2] S. A. Bigdeli, M. Zwicker, P. Favaro, and M. Jin, "Deep Mean-Shift Priors for Image Restoration," p. 10.

[3] W. Dong, P. Wang, W. Yin, G. Shi, F. Wu, and X. Lu, "Denoising Prior Driven Deep Neural Network for Image Restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 10, pp. 2305–2318, Oct. 2019.

[4] A. Durmus, G. O. Roberts, G. Vilmart, and K. C. Zygalakis, "Fast Langevin based algorithm for MCMC in high dimensions," *Ann. Appl. Probab.*, vol. 27, no. 4, pp. 2195–2237, Aug. 2017.

[5] S. A. Bigdeli and M. Zwicker, "Image Restoration using Autoencoding Priors," *arXiv:1703.09964 [cs]*, Mar. 2017.

[6] D. Vats, "MALA versus Random Walk Metropolis," p. 9.

[7] X. Wang and S. H. Chan, "Parameter-free Plug-and-Play ADMM for image restoration," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, 2017, pp. 1323–1327.

[8] S. H. Chan, X. Wang, and O. A. Elgendy, "Plug-and-Play ADMM for Image Restoration: Fixed-Point Convergence and Applications," *IEEE Trans. Comput. Imaging*, vol. 3, no. 1, pp. 84–98, Mar. 2017.

[9] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-Play priors for model based reconstruction," in *2013 IEEE Global Conference on Signal and Information Processing*, Austin, TX, USA, 2013, pp. 945–948.

[10] G. Alain and Y. Bengio, "What Regularized Auto-Encoders Learn from the Data-Generating Distribution," p. 31.

ACKNOWLEDGMENTS

This work is supported by the NSF of China (Project Number 61521002). We would like to thank Xinyue Liang for discussions on MCMC methods. We also thank the reviewers and the area chair for their valuable comments.

CONTACTS

This work is supported by the NSF of China (Project Number 61521002). We would like to thank Xinyue Liang for discussions on MCMC methods. We also thank the reviewers and the area chair for their valuable comments.

wechat

linkedin

github