Caltech

Derivative-Free Methods for Machine Learning Tasks

The Ensemble Kalman Filter

Nikola B. Kovachki[1] Andrew M. Stuart[1]

[1]Computational and Mathematical Sciences
California Institute of Technology

Inverse Problems and Machine Learning
February 9-11th, 2018

Table of Contents

Table of Contents

Supervised Learning

- **Data**: $\{(x_j, y_j)\}_{j=1}^{N}$ with $x_j \in \mathcal{X}$, $y_j \in \mathcal{Y}$ and $\mathcal{X}, \mathcal{Y}$ Hilbert spaces.
- **Find**: $\mathcal{G}(u|\cdot) : \mathcal{X} \to \mathcal{Y}$ for parameter $u \in \mathcal{U}$ consistent with the data.
- **Concatenate**:

$$\mathrm{y} = \mathrm{G}(u|\mathrm{x}) + \eta$$

where $\mathrm{G}(\cdot|\mathrm{x}) : \mathcal{U} \to \mathcal{Y}^N$ and η is model or data error.

- **Losses**: $\Phi(u; \mathrm{x}, \mathrm{y})$

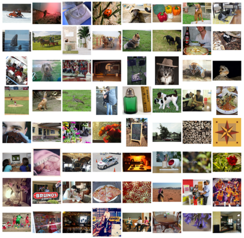
$$\frac{1}{2}\|\mathrm{y} - \mathrm{G}(u|\mathrm{x})\|^2_{\mathcal{Y}^N} + R(u) \quad \text{or} \quad -\sum_{j=1}^{N} \langle y_j, \log \mathcal{G}(u|, x_j)\rangle_{\mathcal{Y}} + R(u)$$

- **Standard Solution (SGD)**:

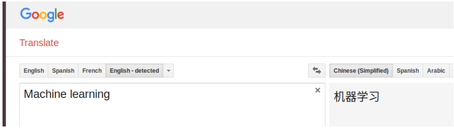
$$\dot{u} = -\nabla_u \Phi(u; \mathrm{x}, \mathrm{y}); \quad u(0) = u_0$$
$$u^* = u(T)$$

Example

- **Classification**:



- **NLP**:



Online Supervised Learning

- **Data**: As before possibly with $N = \infty$.
- **Dynamic**: For $j = 0, 1, 2, \dots$

$$\begin{aligned} u_{j+1} &= u_j \\ y_{j+1} &= \mathcal{G}(u_{j+1}|x_{j+1}) + \eta_{j+1} \end{aligned}$$

- **Find**: u_j given $Y_j = \{y_k\}_{k=1}^{j}$ and update sequentially.
- **Loss**: $\Phi(u; x, y)$

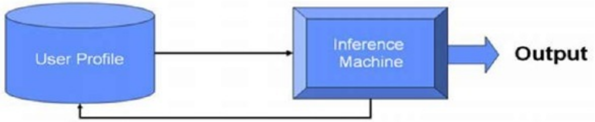
$$\frac{1}{2}\|y - \mathcal{G}(u|x)\|_{\mathcal{Y}}^2 + R(u) \quad \text{or} \quad -\langle y, \log \mathcal{G}(u|x)\rangle_{\mathcal{Y}} + R(u)$$

- **Standard Solution (OGD)**:

$$\begin{aligned} \dot{u} &= -\nabla_u \Phi(u; x_{j+1}, y_{j+1}); \quad u(0) = u_j \\ u_{j+1} &= u(T_j) \end{aligned}$$

Example

- **Model Improvement**:



- **Stream Data**:

Semi-Supervised Learning (on a graph)

Bertozzi and Flenner 2012. (MMS)

Bertozzi, Luo, Stuart, Zygalakis 2017. (preprint)

- **Data**: $\{x_j\}_{j \in Z}$ and $\{y_j\}_{j \in Z'}$ with $Z' \subset Z$ and $|Z'| \ll |Z|$.
- **Find**: $u : Z \to \mathbb{R}^m$ such that

$$y_j = S(u(j)) + \eta_j \quad \forall j \in Z'$$

$S : \mathbb{R}^m \to \mathcal{Y}$ is pre-specified.

- **Loss**:

$$\Phi(u; \mathrm{x}, \mathrm{y}) = \frac{1}{2\gamma^2} \sum_{j \in Z'} \|y_j - S(u(j))\|_{\mathcal{Y}}^2 + R(u; \mathrm{x})$$

- **Standard Solution**: Probit (convex optimization) or MCMC (Bayesian).

Example

- **Clustering**:

Table of Contents

Continous-time EnKF

Kantas, Beskos, Jasra, (2014) (JUQ)

Iglesias, Law and Stuart, 2013. (IP)

- **Inverse Problem**:

$$\mathsf{y} = \mathsf{G}(u) + \eta \qquad \eta \sim \mathcal{N}(0, \Gamma)$$
$$u \sim \mu_0(u)$$

- **Sequential Monte Carlo (SMC)**:

$$\mu_n(du) \propto \exp(-nh\Phi(u; y))\mu_0(du)$$

- **Approximate SMC (EnKF)**:

$$u_{n+1}^{(j)} = u_n^{(j)} + C^{\mathrm{uw}}(u_n)(C^{\mathrm{ww}}(u_n) + \Gamma)^{-1}(\mathsf{y} - \mathsf{G}(u_n^{(j)}))$$

- **Continous-time**: $\Gamma \mapsto \frac{1}{h}\Gamma$, $h \to 0$

$$\dot{u}^{(j)} = -\frac{1}{J}\sum_{k=1}^{J}\langle \mathsf{G}(u^{(k)}) - \bar{\mathsf{G}}, \mathsf{G}(u^{(j)}) - \mathsf{y}\rangle_\Gamma \; u^{(k)}$$

Approximate Natural Gradient Decent

Amari, 1998. (NC)

- **Linear**: $\mathrm{G}(\cdot) = A\cdot$

$$\dot{u}^{(j)} = -C(u)\nabla_u \Phi(u^{(j)}, \mathrm{y})$$

where

$$C(u) = \frac{1}{J}\sum_{j=1}^{J}(u^{(j)} - \bar{u}) \otimes (u^{(j)} - \bar{u}); \quad \Phi(u, \mathrm{y}) = \frac{1}{2}\|\mathrm{y} - Au\|_\Gamma^2$$

- **Natural Gradient Decent**:

$$\dot{u} = -F^{-1}(u)\nabla_u \Phi(u, \mathrm{y})$$

- **Cramér-Rao**:

$$\mathrm{Cov}[\hat{u}] \preceq F^{-1}(u)$$

Long-time Linear Behavior

Schillings and Stuart 2017. (SINUM)

Theorem

Suppse $G(\cdot) = A\cdot$ *and that* y *is the image of a truth* $u^\dagger$ *under* A. *Define* $r^{(j)}(t) = u^{(j)}(t) - u^\dagger$ *then (under some assumptions)*

$$Ar^{(j)}(t) = Ar_{\|}^{(j)}(t) + Ar_{\perp}^{(j)}(t)$$

with $Ar_{\|}^{(j)} \in span\{A(u^{(j)}(0) - \bar{u}(0))\}$ *and* $Ar_{\perp}^{(j)} \in span\{A(u^{(j)}(0) - \bar{u}(0))\}^{\perp}$. *Furthermore*

$$Ar_{\|}^{(j)}(t) \to 0 \quad \text{as } t \to \infty$$

$$Ar_{\perp}^{(j)}(t) = Ar_{\perp}^{(1)}(0) \quad \forall t \geq 0.$$

Arbitrary Loss

- **Non-linear**:

$$\begin{aligned}\dot{u}^{(j)} &= -C^{\mathrm{uw}}(u)\Gamma^{-1}(\mathsf{G}(u^{(j)}) - y) \\ &= -C^{\mathrm{uw}}(u)\nabla_{\mathsf{z}}\Psi(\mathsf{G}(u^{(j)}), \mathsf{y}) \\ &= -\frac{1}{J}\sum_{k=1}^{J}\langle \mathsf{G}(u^{(k)}) - \bar{\mathsf{G}}, \nabla_{\mathsf{z}}\Psi(\mathsf{G}(u^{(j)}), \mathsf{y})\rangle u^{(k)}\end{aligned}$$

$$C^{\mathrm{uw}}(u) = \frac{1}{J}\sum_{j=1}^{J}(u^{(j)} - \bar{u}) \otimes (\mathsf{G}(u^{(j)}) - \bar{\mathsf{G}}); \quad \Psi(\mathsf{z}, \mathsf{y}) = \frac{1}{2}\|\mathsf{y} - \mathsf{z}\|_{\Gamma}^{2}$$

- **Concatenate**: $u = [u^{(1)}, \dots, u^{(J)}]$

$$\dot{u} = -D(u)u$$

where

$$D^{(jk)}(u) = \frac{1}{J}\langle \mathsf{G}(u^{(k)}) - \bar{\mathsf{G}}, \nabla_{\mathsf{z}}\Psi(\mathsf{G}(u^{(j)}), \mathsf{y})\rangle$$

Nesterov Momentum

Su, Boyd, Candés 2014. (NIPS)

- **Momentum**:

$$\begin{cases} u_{n+1} = v_n - h\nabla f(v_n) \\ v_{n+1} = u_{n+1} + \frac{n}{n+3}(u_{n+1} - u_n) \\ v_0 = u_0 \end{cases} \iff \begin{cases} \ddot{u} + \frac{3}{t}\dot{u} + \nabla f(u) = 0 \\ \dot{u}(0) = 0 \\ u(0) = u_0 \end{cases}$$

- **Modified Limit**:

$$\begin{cases} \ddot{u}^{(j)} + \frac{3}{t}\dot{u}^{(j)} = -C^{\mathrm{uw}}(u)\nabla_{\mathrm{z}}\Psi(\mathrm{G}(u^{(j)}), \mathrm{y}) \\ \dot{u}^{(j)}(0) = 0 \\ u^{(j)}(0) = u_0^{(j)} \end{cases}$$

Discrete Scheme

- **Concatenate**: $u = [u^{(1)}, \ldots, u^{(J)}]$

$$\ddot{u} + \frac{3}{t}\dot{u} = -D(u)u$$

where

$$D^{(jk)}(u) = \frac{1}{J}\langle \mathrm{G}(u^{(k)}) - \bar{\mathrm{G}}, \nabla_z \Psi(\mathrm{G}(u^{(j)}), \mathrm{y})\rangle$$

- **Discretize**:

$$\begin{cases} u_{n+1} = v_n - h_n D(v_n) v_n \\ v_{n+1} = u_{n+1} + \frac{n}{n+3}(u_{n+1} - u_n) \\ v_0 = u_0 = [u_0^{(1)}, \ldots, u_0^{(J)}]^T \end{cases}$$

Initialization, Noise, and Predictions

- **Initial Ensemble**:

$$u_0^{(1)}, \ldots, u_0^{(J)} \sim \mu_0(u)$$

- **Noise (Supervised)**:

$$\tilde{v}_{n+1}^{(j)} = v_n^{(j)} + \xi_{n+1}^{(j)} \qquad \xi_{n+1}^{(j)} \sim \mu_{n+1}(u)$$
$$\text{Cov}[\mu_{n+1}] \propto \sqrt{h_n}\text{Cov}[\mu_0]$$

- **Ensemble Refresh (Online)**:

$$u_{n+1}^{(j)} = \bar{u}_n + \xi_{n+1}^{(j)} \qquad \xi_{n+1}^{(j)} \sim \mu_0(u)$$

- **Predict**:

$$\bar{u}_{n+1} = \frac{1}{J}\sum_{j=1}^{J} u_{n+1}^{(j)}$$

Complete Algorithm

- **Mini-batch data** (at each step):

$$\mathsf{x}_n = \{x_{i_l^{(n)}}\}_{l=1}^m \quad \mathsf{y}_n = \{y_{i_l^{(n)}}\}_{l=1}^m$$

  where $\{i_1^{(n)}, \dots, i_m^{(n)}\} \subseteq \{1, \dots, N\}$.

- **Compute**: use discrete scheme as shown with

$$\mathsf{x} \mapsto \mathsf{x}_n \quad \mathsf{y} \mapsto \mathsf{y}_n$$

- **Step-size**: adaptive

$$h_n = \frac{h}{\epsilon + \|D_n\|}$$

Table of Contents

Convolutional Models

Ba, Kiros and Hinton 2016. (NIPS)

Net 1 $\sim$ 14k	**Net 2** $\sim$ 30k
Conv12x3x3 MaxPool2x2	Conv12x3x3 Conv12x3x3 MaxPool2x2
Conv24x3x3 MaxPool2x2	Conv24x3x3 Conv24x3x3 MaxPool2x2
Conv32x3x3 MaxPool2x2	Conv32x3x3 Conv32x3x3
FC-100	FC-100
FC-10	FC-10

- ReLU applied after each block.
- Layer Normalization applied after each convolutional layer.

MNIST Dataset

LeCun and Cortes 1999.

Caltech

MNIST Supervised

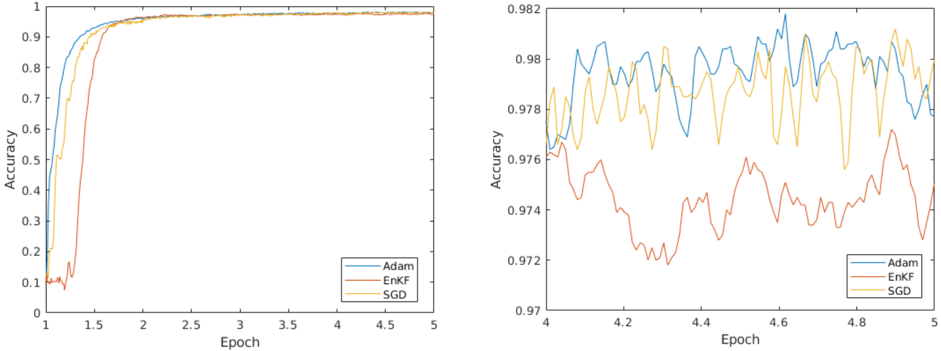


Figure: Test Accuracy of **Net 1** on MNIST (batched).

J	Loss	Momentum	Noise	Refresh
5000	Cross Entropy	✓	✓	χ

MNIST Online

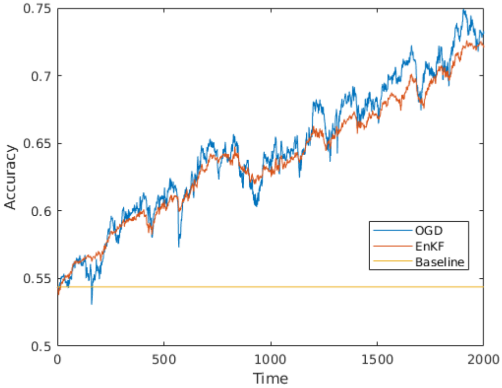
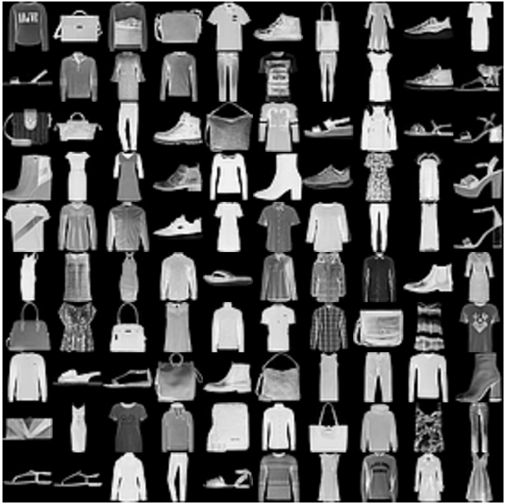


Figure: Test Accuracy of **Net 1** on MNIST (online).

J	Loss	Momentum	Noise	Refresh
5000	Cross Entropy	χ	χ	$\checkmark$

Fashion MNIST Dataset

Xiao, Rasul and Vollgraf 2017.



Fashion MNIST Supervised

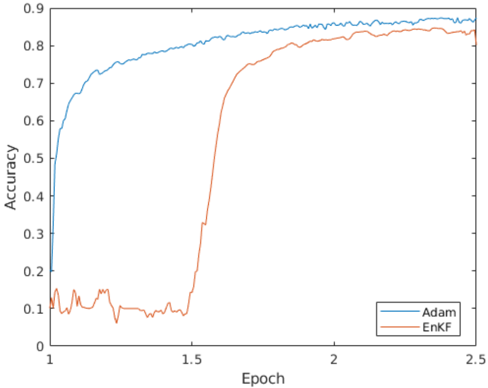
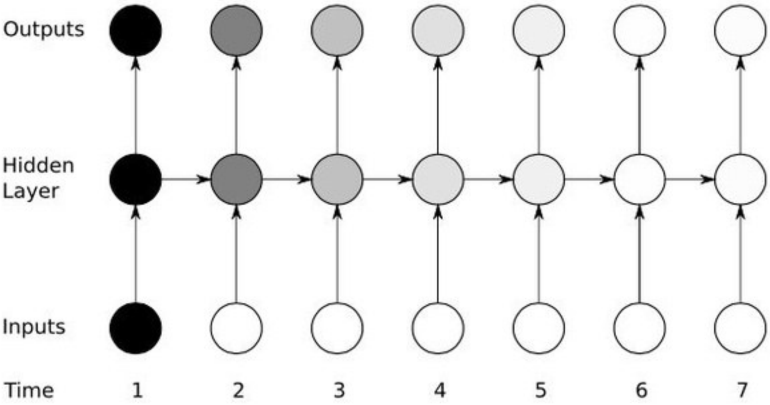


Figure: Test Accuracy of **Net 2** on Fashion MNIST (batched).

J	Loss	Momentum	Noise	Refresh
5000	Cross Entropy	✓	✓	χ

RNN



Time Series Online

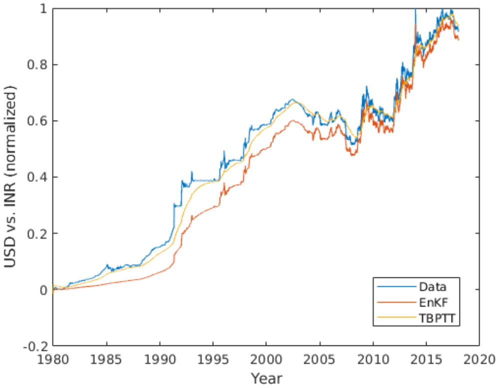


Figure: Time Series prediction with a RNN.

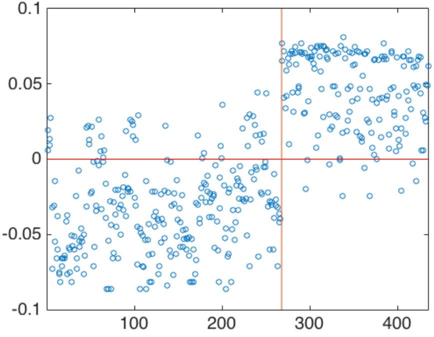
J	Loss	Momentum	Noise	Refresh
1000	MSE	χ	χ	χ

Voting Records Dataset

U.S. House of Representatives 1984, 16 key votes. For each congress representative we have an associated feature vector $x_j \in \mathbb{R}^{16}$ such as

$$x_j = (1, -1, 0, \cdots, 1)^T;$$

1 is "yes", -1 is "no" and 0 abstain/no-show. Here $|Z| = 435$ and $|Z'| = 5$.



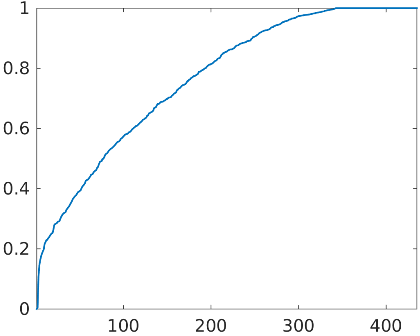


Figure: Strong Prior Information: Fiedler Vector and Spectrum (Normalized)

Voting Records Semi-Supervised

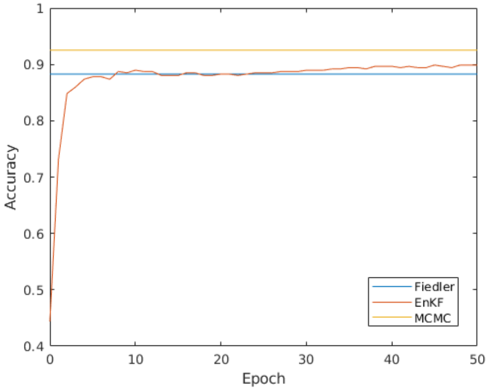


Figure: Accuracy on Voting Records.

J	Loss	Momentum	Noise	Refresh
2000	MSE	χ	✓	χ

Summary

- Machine learning as inverse/filtering problem.
- EnKF as a minimization scheme.
- Modifications to original method.
- Numerics show promise as alternative for:
  - SGD
  - OGD
  - BPTT
  - MC

References I

- Bertozzi L. A., Flenner A. Diffuse Interface Models on Graphs for Classification of High Dimensional Data. *Multiscale Model. Simul.* 10(3). pp. 1090–1118. 2012.
- Bertozzi L. A., Luo X., Stuart M. A., Zygalakis C. K. Uncertainty Qunatification in the Classification of High Dimensional Data. *Preprint*. arXiv:1703.08816. 2017.
- Kantas N., Luo X., Beskos A., Jasra A. Sequential Monte Carlo methods for high-dimensional inverse problems. *SIAM/ASA Journal on Uncertainty Quantification*, 2. pp. 464-489. 2014.
- Iglesias M., Law K., Stuart M. A. Ensemble Kalman methods for inverse problems. *Inverse Problems*, 29. p. 045001. 2013.
- Schillings C., Stuart M. A. Analysis of the ensemble Kalman filter for inverse problems. *SIAM J Numerical Analysis*, 55(3). pp. 1264-1290. 2017.

References II

- Amari S. Natural Gradient Works Efficiently in Learning. *Neural Computation*, 10. pp. 251–276. 1998.
- Su W., Boyd S., Candés J. E. A Differential Equation for Modeling Nesterov's Accelerated Gradient Method: Theory and Insights. *Advances in Neural Information Processing Systems*, 28. 2014.
- Ba J., Kiros J., Hinton J. Layer Normalization. *Advances in Neural Information Processing Systems*, 30. 2016.
- LeCun Y., Cortes C. The MNIST database of handwritten digits. *http://yann.lecun.com/exdb/mnist/*. 1998.
- Xiao H., Rasul K., Vollgraf R. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv:1708.07747*. 2017.