Belief flows for robust online learning

Pedro A. Ortega, Koby Crammer, Daniel D. Lee

University of Pennsylvania

February 2, 2015

Our goal

Design an online learning method that

1. scales to very large data
2. and complex models
3. while avoiding overfitting, . . .

. . . by combining ideas from

1. stochastic gradient descent (for its simplicity),
2. Bayesian filtering (to avoid overfitting),
3. and multi-armed bandits (to bypass costly intregration).

Big challenges in machine learning...

- life-long learning,
- computer vision,
- natural language processing,
- bioinformatics,
- robotics...

$\Rightarrow$ learning tasks with very large datasets/data streams.

. . . ask for more complex models

Common wisdom:

- With more data, our learning algorithms find better parameters. [Halevy et al., 2009]

. . . ask for more complex models

Common wisdom:

- With more data, our learning algorithms find better parameters. [Halevy et al., 2009]

However:

- Having more data asks for richer models to answer complex questions, and richer models require regularization—even when data is abundant [Welling and Teh, 2011].

Main ingredients

Stochastic gradient descent (SGD):

1. Phrases learning task as optimization problem.
2. Pros: Simple; scalable; strong theoretical guarantees (convex).
3. Cons: Overfits if not regularized.

Bayesian filtering:

1. Capture parameter uncertainty.
2. Pros: Principled approach to avoid overfitting.
3. Cons: Computationally very expensive for complex models.

Main ingredients

Stochastic gradient descent (SGD):

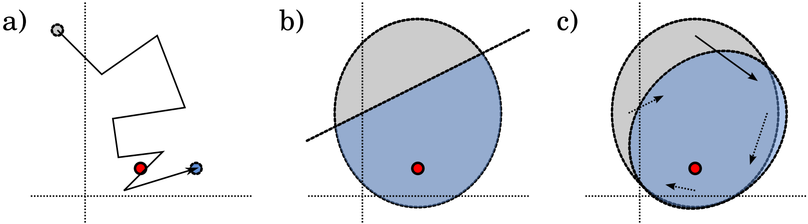
1. Phrases learning task as optimization problem.
2. Pros: Simple; scalable; strong theoretical guarantees (convex).
3. Cons: Overfits if not regularized.

Bayesian filtering:

1. Capture parameter uncertainty.
2. Pros: Principled approach to avoid overfitting.
3. Cons: Computationally very expensive for complex models.

Thompson sampling: to bypass marginalization.

Main idea



We want to combine the best of both worlds.

Belief flows: ingredients

Given:

- Model $F_w(x)$, input $x \in \mathbb{R}^p$, parameter $w \in \mathbb{R}^d$
- Family of belief distributions: $P_\theta(w)$, $\theta \in \Theta$
- Family of flows: $w' = f_\xi(w)$, $\xi \in \Xi$

Belief flows: typical algorithm

For each round $n = 1, 2, \ldots$

1. Given prior $P_n(w)$
2. Collect input x_n
3. Sample parameter $w_n \sim P_n(w)$
4. Predict output $\hat{y}_n = F_{w_n}(x_n)$
5. Observe true output y_n (and get loss $\ell(y_n, \hat{y}_n)$)
6. Observe update $w'_n = f_\xi(w_n)$
7. Infer posterior $P_{n+1}(w)$
   by minimizing $D_{KL}(P_{n+1}||P_n)$ s.t. $w'_n = f_\xi(w)$ and $\xi \in \Xi$.

Gaussian belief flows

- Family of belief distributions:

$$P(w) = \mathcal{N}(w; \mu, \Sigma), \qquad [\mu, \Sigma] \in \Theta.$$

- Family of flow fields:

$$w' = Aw + b, \qquad [A, b] \in \Xi.$$

- Update oracle:

$$w' = w - \eta \frac{\partial}{\partial w} \ell(y, \hat{y})$$

where $\ell(y, \hat{y})$ is a loss function.

Update rule

The posterior that minimizes $D_{KL}(P_{n+1} \| P_n)$ subject to the constraints is given by

$$\Sigma_{n+1} = A^* \Sigma_n A^{*T} \qquad \mu_{n+1} = \mu_n A^*(\mu_n - w_n) + w_n'$$

where

$$A^* = I_{d\times d} + U_n \sqrt{D_n} \left\{ \begin{bmatrix} \hat{\mu} & \hat{\nu} \end{bmatrix} (A_{2\times 2} - I_{2\times 2}) \begin{bmatrix} \hat{\mu}^T \\ \hat{\nu}^T \end{bmatrix} \right\} \frac{1}{\sqrt{D_n}} U_n^T$$

$$A_{2\times 2} = \frac{1}{\sqrt{v_\parallel^2 + v_\perp^2}} \frac{1}{\sqrt{v_\parallel^2 + v_\perp^2}} \begin{bmatrix} \frac{u\sqrt{v_\parallel^2+v_\perp^2}+\delta_1\sqrt{4+u^2(4+v_\parallel^2+v_\perp^2)}}{2(1+u^2)} v_\parallel & -\delta_2 v_\perp \\ \frac{u\sqrt{v_\parallel^2+v_\perp^2}+\delta_1\sqrt{4+u^2(4+v_\parallel^2+v_\perp^2)}}{2(1+u^2)} v_\perp & +\delta_2 v_\parallel \end{bmatrix}$$

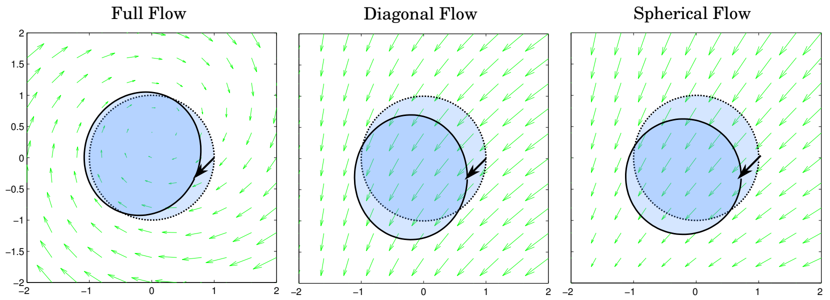
Restricting Gaussian belief flows

We obtain simpler update rules by restricting the flows and the shapes of the belief distributions.

Examples:

1. Diagonal $\Sigma \Rightarrow$ diagonal flows
2. Isotropic $\Sigma \Rightarrow$ spherical flows
3. < 1 singular values of $A \Rightarrow$ non-expansive flows

Example Gaussian belief flows



Empirical evaluation

Goal: compare effects of regularization schemes.

- online classification error (%80 of data)
- test classification error (%20 of data)

Tasks:

1. Logistic regression
   - Data: MUSHROOM, COVTYPE, IJCNN, EEG, A9A
   - Algorithms: AROW, SGD, Bayesian Langevin
2. Feed forward neural network (784-200-10)
   - MNIST: plain, random & image background
   - Algorithms: SGD, DROPOUT

Experimental results: logistic regression

Binary Classification Results

Online Classification Error in %

	MUSHR.	COVTYPE	IJCNN	EEG	A9A	Rank
AROW	**5.32**	**22.58**	**8.44**	43.59	**17.79**	**1.2**
SGD	11.86	28.03	9.01	**43.39**	18.62	1.8
BLANG	14.44	29.30	12.86	43.71	20.51	3.2
BFLO	14.30	28.14	10.34	44.07	19.03	3.8

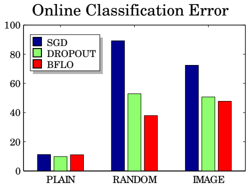
Test Classification Error in %

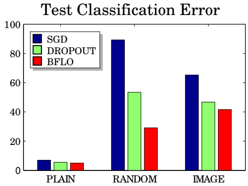
	MUSHR.	COVTYPE	IJCNN	EEG	A9A	Rank
$\max\{\sigma_{\text{err}}\}$	0.23	0.12	0.26	0.69	0.08	
AROW	9.59	37.18	20.10	65.38	**15.85**	3.0
SGD	5.35	37.45	19.10	**60.57**	17.45	2.6
BLANG	**1.16**	38.39	**15.97**	64.85	17.68	2.6
BFLO	1.79	**37.03**	16.92	62.76	17.00	**1.8**

Experimental results: neural networks

MNIST Classification Results

	PLAIN	RANDOM	IMAGES	Rank
Online Classification Error in %				
$\max\{\sigma_{\text{err}}\}$	0.07	0.96	1.16	
SGD	11.25	89.14	72.41	3.0
DROPOUT	9.84	52.87	50.68	1.6
BFLO	11.01	37.94	47.71	**1.3**
Test Classification Error in %				
$\max\{\sigma_{\text{err}}\}$	0.44	3.33	6.05	
SGD	7.01	89.17	65.17	3.0
DROPOUT	5.52	53.42	46.67	2.0
BFLO	5.00	29.11	41.55	**1.0**





Conclusions

- Regularization scheme.
- Works best for complex models.
- Fairly robust to noise.
- Related to ensemble learning methods under quadratic cost functions.
- Related to multi-armed bandits (exploration-exploitation dilemma).
- Can be extended to other belief shapes and flows.

Thank you!

References

A. Halevy, P. Norvig, and F. Pereira. The unreasonable effectiveness of data. *Intelligent Systems, IEEE*, 24(2):8–12, 2009.

M. Welling and Y.-W. Teh. Bayesian Learning via Stochastic Gradient Langevin Dynamics. In *ICML*, 2011.