Statistical Learning and Scalability of Additive Models with Total Variation Regularization

Shin Matsushima
The University of Tokyo

東京大学
THE UNIVERSITY OF TOKYO

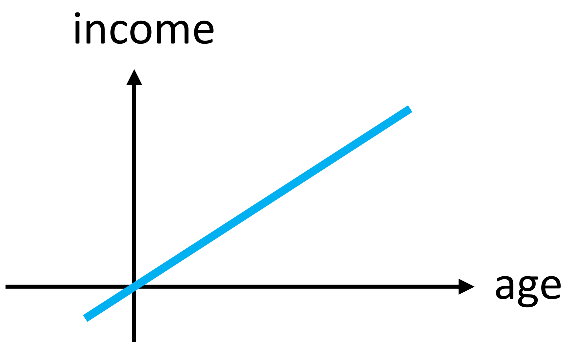
New In ML 2019

The First Group Session for Newcomers to Machine Learning

Motivation

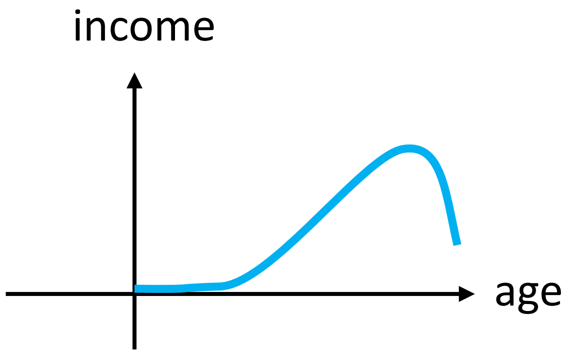
- Given dataset $(\mathbf{x}_i, y_i)_{i=1..m}, \mathbf{x}_i \in \mathbb{R}^p$
- Linear models

$$f(\mathbf{x}) = \sum_j w_j x_j$$



- **Additive models**

$$f(\mathbf{x}) = \sum_j f_j(x_j)$$



Deep knowledge discovery with more accurate predictors!

Existing methods

- Based on regularized empirical risk minimization

$$\underset{(f_j)_j}{\text{minimize}} \;\; L(f) + \lambda \sum_j \boxed{R(f_j)}$$

where L is data- and task-dependent and $R(f_j)$ is wiggleness of f_j:

$$\boxed{R(f_j)} = \int \left(\frac{d^2}{d\xi^2} f_j(\xi) \right)^2 d\xi$$

- Boils down to multiple kernel learning (MKL)

$R(f_j)$ is counter-intuitive **Training costs $O(m^2)$**

Our Focus: TVAM

- Total variation-regularized additive models (TVAM)
- Learning additive models

$$\underset{(f_j)_j}{\text{minimize}}\ L(f) + \lambda \sum_j R(f_j)$$

where $R(f_j)$ is **total variation** of f_j

$$R(f_j) = \boxed{||f_j||_{\mathrm{TV}}}$$

More intuitive and more efficiently solvable

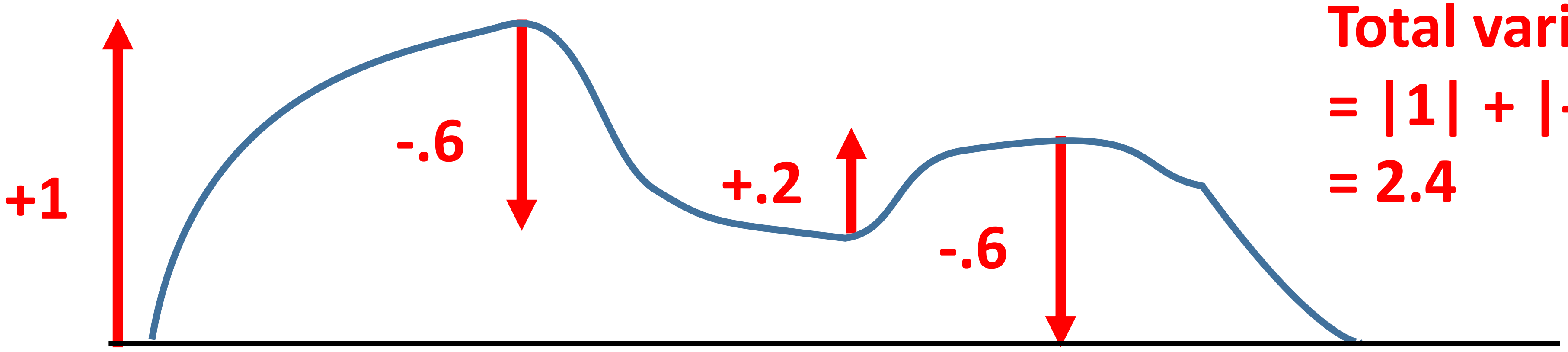
Total Variation

represents how much function value varies in total

- We assume support of each function is bounded, and function is cadlag
- Definition

$$\|f_j\|_{\mathrm{TV}} \triangleq \sup \left\{ |f_j(\xi_1)| + \sum_{i=1..n-1} |f_j(\xi_{i+1}) - f_j(\xi_i)| + |f_j(\xi_n)| \;\middle|\; \xi_1 < \xi_2 < \cdots < \xi_n, \text{ for some } n \right\}$$

- Example



Total variation
= |1| + |-.6| + |.2| +|-.6|
= 2.4

Main Questions

- We study on learning additive models based on

$$\underset{(f_j)_j}{\text{minimize}} \; L(f) + \lambda \sum_j \boxed{\|f_j\|_{\mathrm{TV}}}$$

- Algorithm
  - How to find the solution **efficiently**?
- Theory
  - **Guarantee** on the performance of the solution?

Algorithm

Parameterization

Minimization problem

$$\underset{(f_j)_j}{\text{minimize}} \quad L(f) + \lambda \sum_j \|f_j\|_{\text{TV}}$$

is boiled down to

$$\underset{(w_{jst})_{jst}}{\text{minimize}} \quad L\left(\sum_{j,s,t} w_{jst}\phi_{j,s,t}\right) + \lambda \sum_{j,s,t} |w_{jst}|$$

where

$$\phi_{j,s,t}(\mathbf{x}) = 1 \text{ if } x_{sj} \leq x_j \leq x_{tj} \text{ else } 0$$

Problem

- Minimizing

$$J(\mathbf{w}) = L\left(\sum_{j,s,t} w_{jst}\phi_{j,s,t}\right) + \lambda \sum_{j,s,t} \left|w_{jst}\right|$$

  is an L1-regularized risk minimization of linear model with a design matrix induced by $(\phi_{j,s,t})_{j,s,t}$

$$\boxed{\text{Data matrix}}_{\,j=1..p} \xrightarrow{(\phi_{j,s,t})_{j,s,t}} \boxed{\text{Design matrix}}_{\,(j,s,t)=(1,1,1)..(p,m,m)} \quad i = 1..m$$

- Design matrix is **too big** to fit in memory in a standard machine
  - We can not generate design matrix and then run standard algorithm

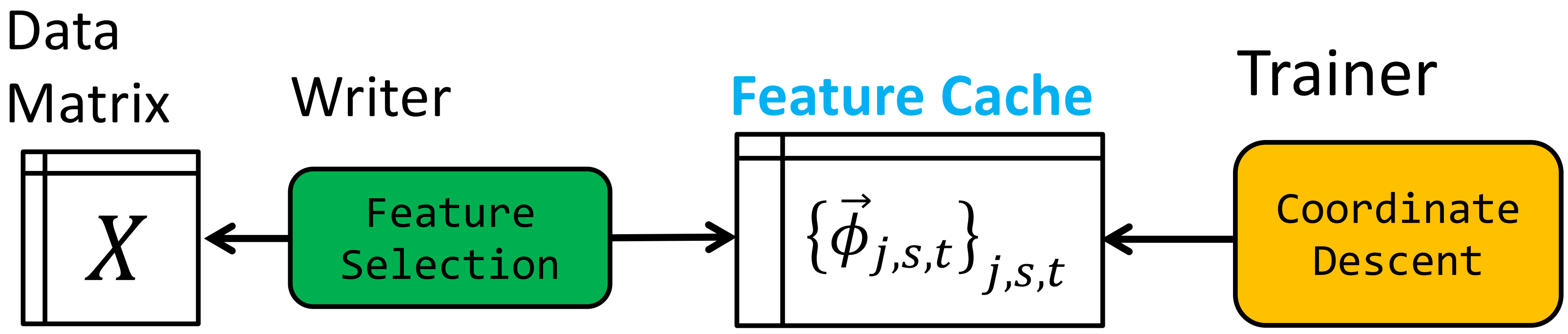
Greedy Coordinate Descent (GCD)

- Repeat until the maximum value is less than λ:
  - $\boxed{\text{Find } (j,s,t) = \underset{j,s,t=(1,1,1)..(p,m,m)}{\operatorname{argmax}} \left| \frac{\partial L}{\partial w_{jst}} \right|}$ **Done by O(mp)!**
  - $\boxed{\text{Minimize } J(\mathbf{w}) \text{ w.r.t. } w_{jst}}$ **Done by O(m)!**

Each step does not require the information of the entire design matrix

Cached Loops

Run two types of threads asynchronously



- Repeat
  - Find $(j, s, t) = \operatorname*{argmax}_{j,s,t=(1,1,1)..(p,m,m)} \left| \frac{\partial L}{\partial w_{jst}} \right|$
  - Push it in **Feature Cache**
  - If small enough then break

- Repeat
  - Randomly choose (j,s,t) in **Feature Cache**
  - Minimize $J(\mathbf{w})$ w.r.t. w_{jst}
  - If w_{jst} stayed at 0, then delete it from **Feature Cache**

Theory

Solution of TVAM

$$\underset{(f_j)_j}{\text{minimize}} \sum_i \ell\left(f(\mathbf{x}_i), y_i\right) + \lambda \sum_j \|f_j\|_{\mathrm{TV}}$$

can be expressed as

$$\hat{f} = \underset{f \in \mathrm{TVAM}(C)}{\operatorname{argmin}} \frac{1}{m} \sum_i \ell\left(f(\mathbf{x}_i), y_i\right)$$

where

$$\mathrm{TVAM}(C) = \left\{ f \,\middle|\, \sum_j \|f_j\|_{\mathrm{TV}} \le C \right\}$$

Main Results

- Gap between ideal and reality
  - Ideal

$$f^* = \operatorname*{argmin}_{f \in \mathrm{TVAM}(C)} \mathbb{E}_{x,y} \ell\big(f(x), y\big)$$

  - Reality

$$\hat{f} = \operatorname*{argmin}_{f \in \mathrm{TVAM}(C)} \frac{1}{m} \sum_i \ell\Big(f(x_i), y_i\Big)$$

- Is the gap smaller when m is large?

$$\mathbb{E}_{\mathbf{x},y} \ell\Big(\hat{f}(\mathbf{x}), y\Big) - \mathbb{E}_{\mathbf{x},y} \ell\Big(f^*(\mathbf{x}), y\Big)$$

- We show **the gap converges to 0 at rate** $O(\sqrt{\log p/m})$ **with high probability**

Why?

- Complexity of TVAM is as much as complexity of linear models!
  - Rademacher complexity for $(\mathbf{x}_i, y_i)_{i=1..m}, \mathbf{x}_i \in \mathbb{R}^p$

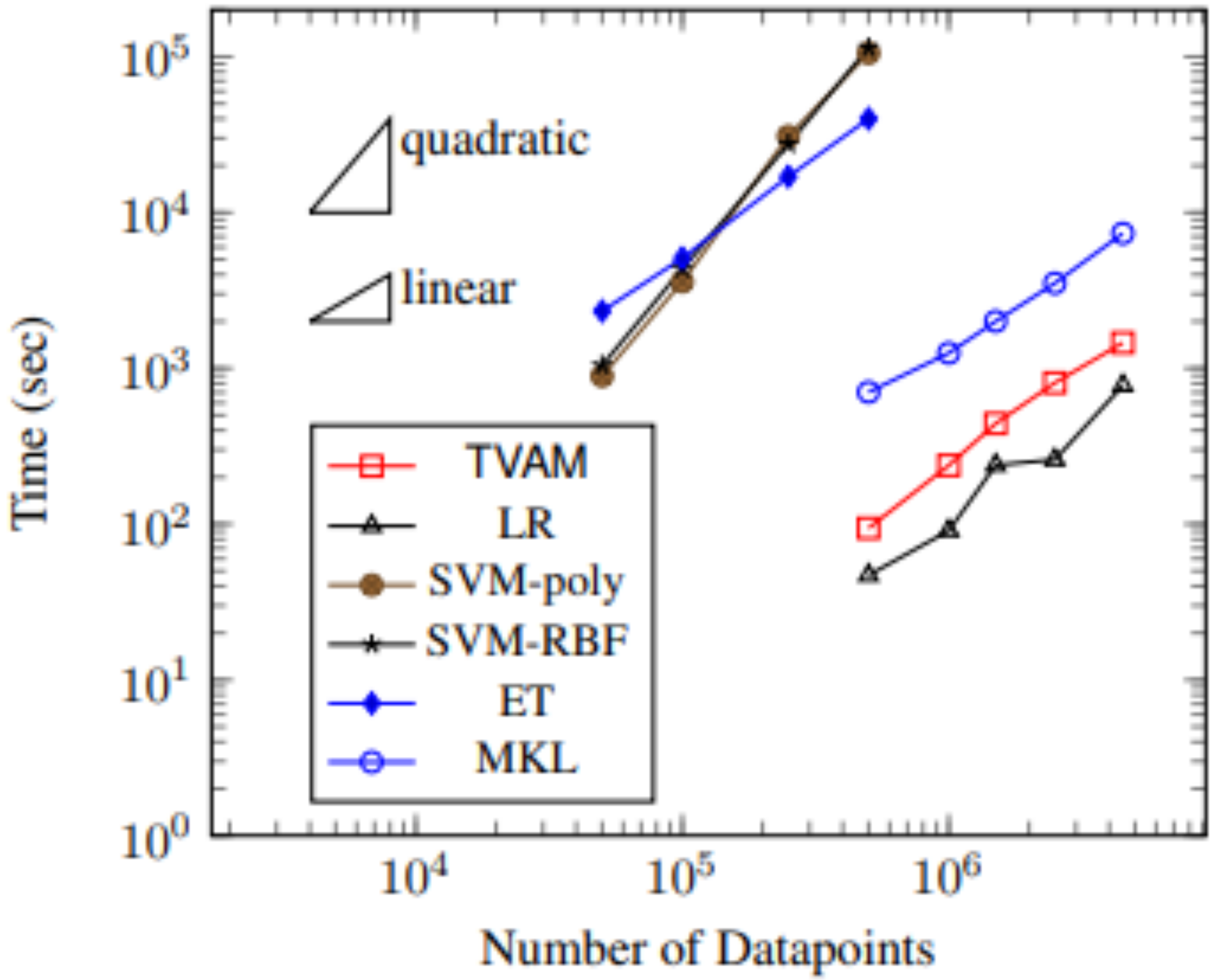
$$\mathrm{Rad}(\mathrm{TVAM}(C)) \sim \sqrt{\frac{\log p}{m}}$$

$$\mathrm{Rad}(\text{Linear model}) \sim \sqrt{\frac{\log p}{m}}$$

Experimental Results

Scalability

- Log loss for binary classification
- Exhibits **linear scalability** in experiments
- TVAM is only a few times slower than learning linear models



Predictive Performance (AUC)

- TVAM achieves **best AUC** among all additive predictors!

Dataset	n	d	TVAM	ET	MKL	FB	NB	LR
				additive predictors				
				nonlinear			linear	
SUSY	4,500,000	17	**87.4**	/	87.2	82.9	78.6	85.4
covtype.binary	500,000	54	**73.8**	55.7	67.6	70.3	70.3	68.5
skin-nonskin	200,000	3	**98.9**	63.6	94.4	67.4	91.2	93.3
cod-rna	59,535	8	**99.3**	99.1	93.6	79.3	81.4	98.8
ijcnn1	49,990	22	96.2	**96.6**	91.5	79.7	90.5	91.3
wilt	4,339	5	**92.3**	85.0	83.6	45.2	50.0	78.8
madelon	2,000	500	**71.3**	68.2	60.9	64.7	62.1	62.2
german.numer	1,000	24	80.8	81.3	**83.0**	80.6	82.6	82.9

Summary of Our Paper

- We learn interpretable additive models based on

$$\underset{(f_j)_j}{\text{minimize}} \; L(f) + \lambda \sum_j \boxed{\|f_j\|_{\mathrm{TV}}}$$

- Algorithm
  - Establish an efficient algorithm
- Theory
  - Statistical learning rate of $O(\sqrt{\log p/m})$
- Experiments
  - Scalability and predictive performance

END

Predictive Performance (AUC)

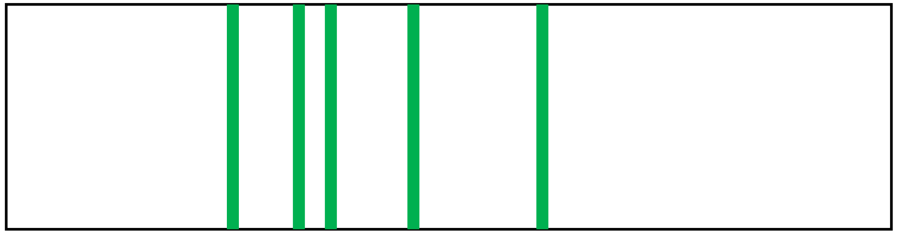
- TVAM achieves **best AUC** among all additive predictors!

Dataset	n	d	additive predictors						non-additive predictors		
				nonlinear			linear		(Kernel SVM)		
			TVAM	ET	MKL	FB	NB	LR	RBF	poly	XGB
SUSY	4,500,000	17	**87.4**	/	87.2	82.9	78.6	85.4	/	/	87.5
covtype.binary	500,000	54	**73.8**	55.7	67.6	70.3	70.3	68.5	/	/	78.4
skin-nonskin	200,000	3	**98.9**	63.6	94.4	67.4	91.2	93.3	/	/	100.0
cod-rna	59,535	8	**99.3**	99.1	93.6	79.3	81.4	98.8	99.6	98.1	99.3
ijcnn1	49,990	22	96.2	**96.6**	91.5	79.7	90.5	91.3	98.2	97.5	99.6
wilt	4,339	5	**92.3**	85.0	83.6	45.2	50.0	78.8	70.4	89.2	94.4
madelon	2,000	500	**71.3**	68.2	60.9	64.7	62.1	62.2	50.0	69.0	86.5
german.numer	1,000	24	80.8	81.3	**83.0**	80.6	82.6	82.9	81.7	80.0	79.3

END

- https://ttsreader.com/

Design matrix



$i = 1..m$

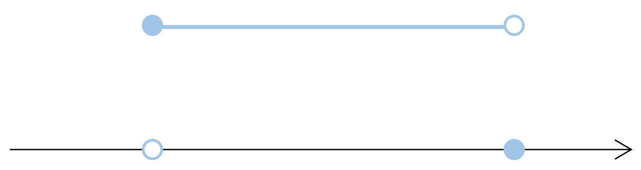
$j, s, t = (1, 1, 1)..(p, m, m)$

With probability more than 1-δ:

$$\mathbb{E}_{x,y}\ell(\hat{f}(x), y) - \ell^* \leq \rho C\sqrt{\frac{5\lceil \log d \rceil}{m}} + \frac{5C}{2}\sqrt{\frac{2\log(2/\delta)}{m}}$$

$\ell : \rho$-Lipschitz

Form of $\phi_{j,s,t}(\cdot)$

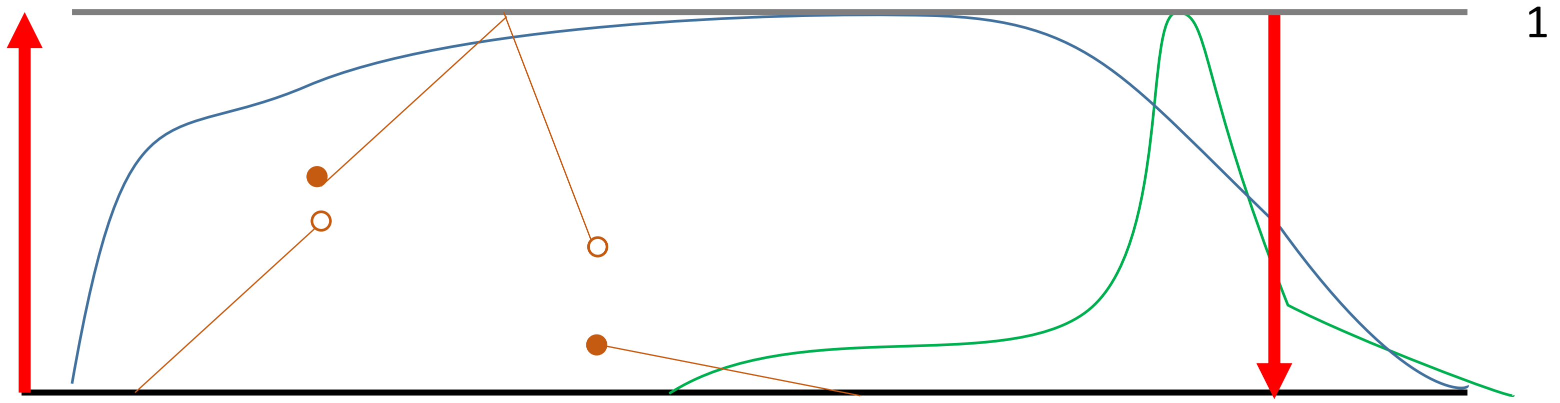


Total variation

We assume support of each function is bounded, and function is cadlag

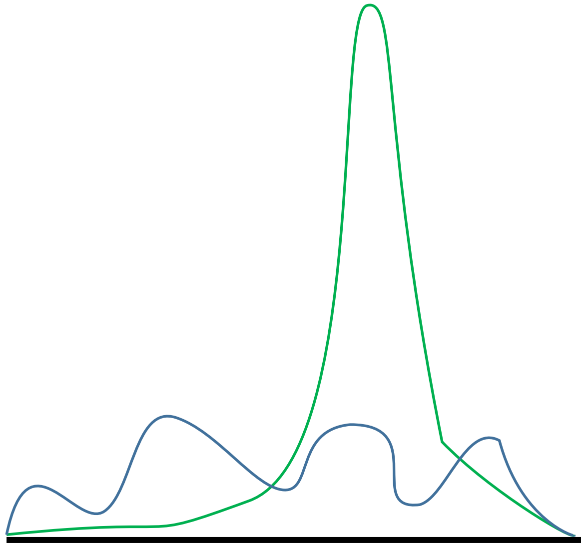
- Total variation represents how much function value varies in total
- Definition

- They all have total variation of 2

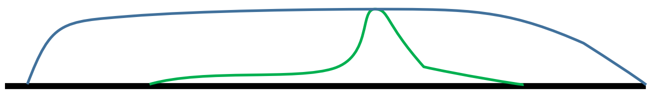


Total variation

- Large total variation



- Small total variation



- Definition