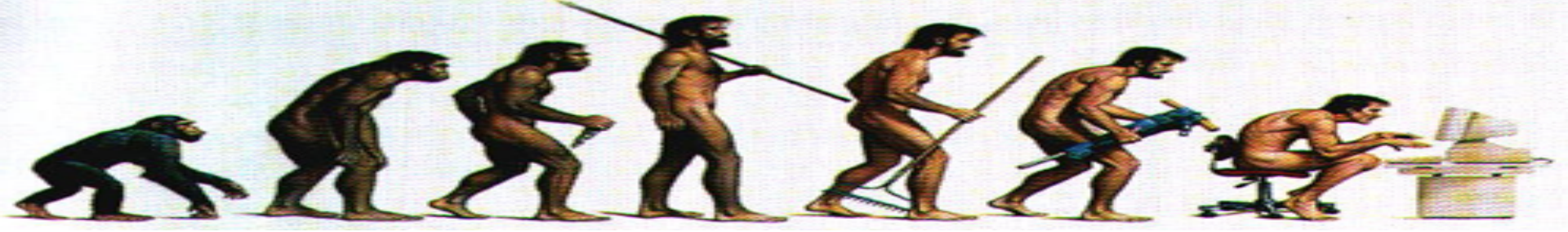


What's the Insight of Self-paced Learning

Deyu Meng

Xian Jiaotong University

dymeng@mail.xjtu.edu.cn

http://gr.xjtu.edu.cn/web/dymeng

Four key words

- Machine Learning
- Cognitive Science
- Self-paced Learning
- Big Data (Video/Multimedia)

Four key words

- Machine Learning
- Cognitive Science
- Self-paced Learning
- Big Data (Video/Multimedia)

A General Machine learning Framework

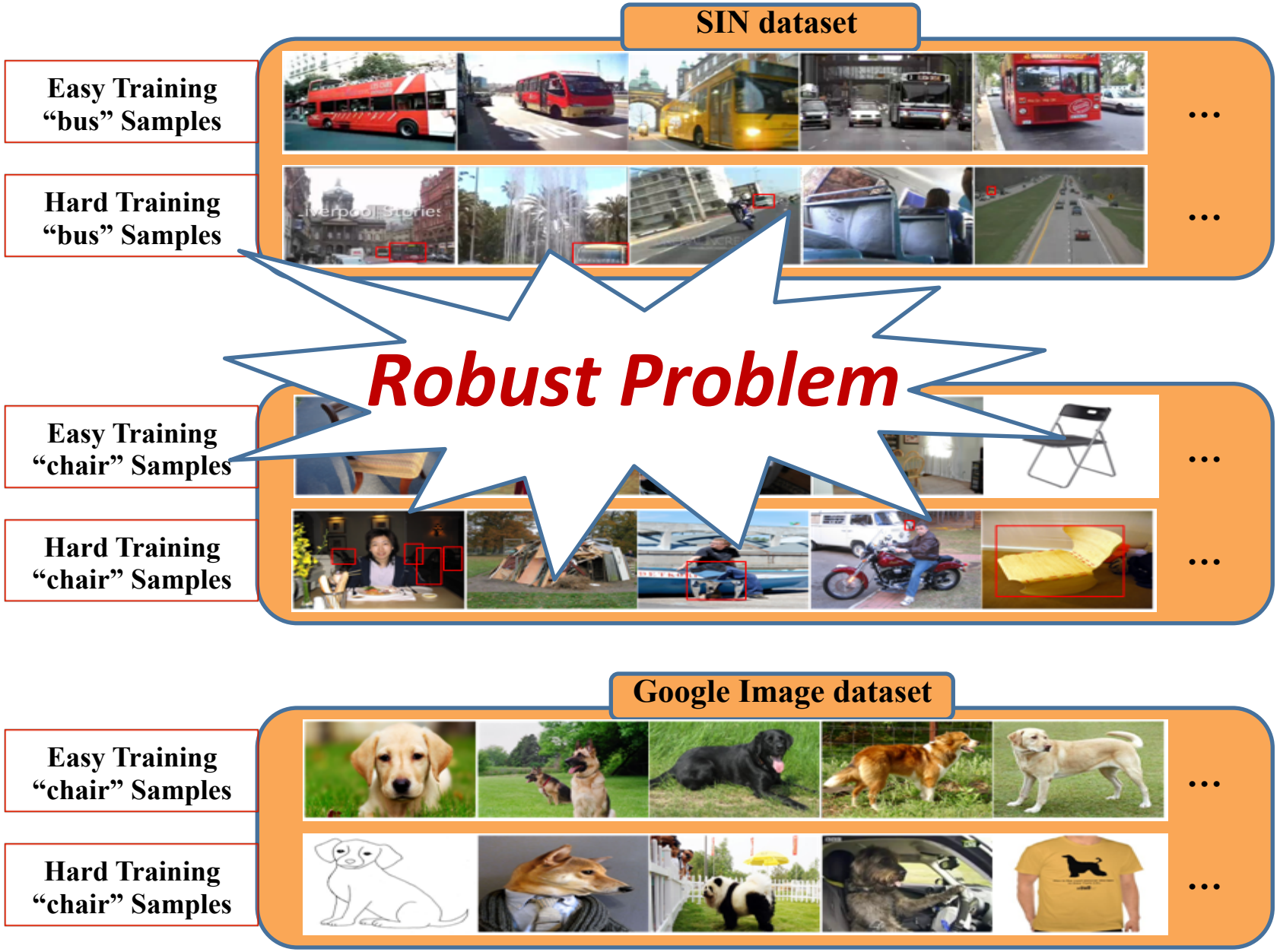
$$\min_{f \in \mathcal{F}} \; l(D, f(w)) \; + \; p(w)$$

Learning machine	Loss/likelihood term	Regularization/prior term

A General Machine learning Framework

$$\min_{f \in \mathcal{F}} \; l(D, f(w)) \; + \; p(w)$$

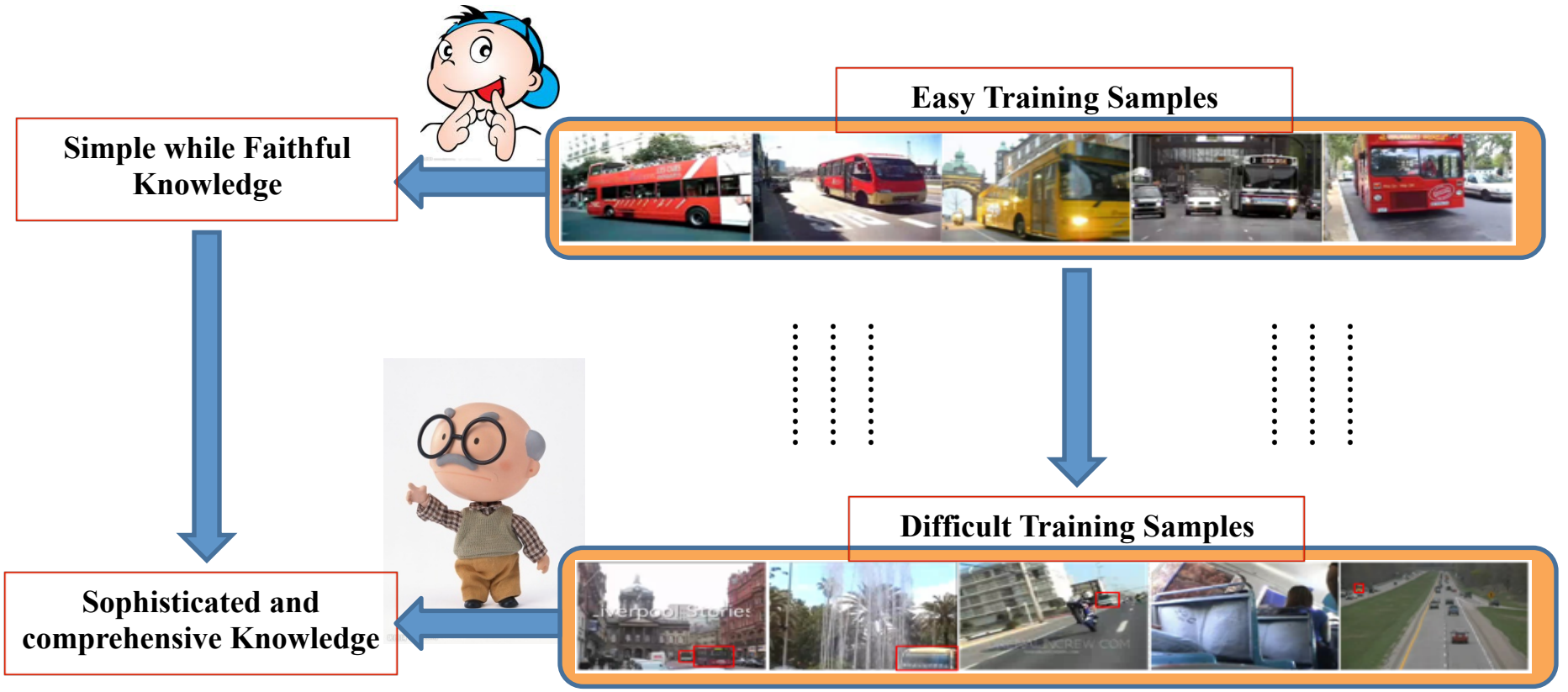
Learning machine	Loss/likelihood term	Regularization/prior term


SIN dataset
Easy Training "bus" Samples
Hard Training "bus" Samples
Robust Problem
Easy Training "chair" Samples
Hard Training "chair" Samples
Google Image dataset
Easy Training "chair" Samples
Hard Training "chair" Samples

Four key words

- Machine Learning
- Cognitive Science
- Self-paced Learning
- Big Data (Video/Multimedia)

- How human/animal learns: First input easy samples and gradually involve more into training from easy to complex



Curriculum Learning

Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *ICML, pages 41–48, 2009.*

Curriculum Learning

Yoshua Bengio[1] YOSHUA.BENGIO@UMONTREAL.CA
Jérôme Louradour[1,2] JEROMELOURADOUR@GMAIL.COM
Ronan Collobert[3] RONAN@COLLOBERT.COM
Jason Weston[3] JASONW@NEC-LABS.COM
(1) U. MONTREAL, P.O. BOX 6128, MONTREAL, CANADA (2) A2IA SA, 40BIS FABERT, PARIS, FRANCE
(3) NEC LABORATORIES AMERICA, 4 INDEPENDENCE WAY, PRINCETON, NJ, USA

Abstract

Humans and animals learn much better when the examples are not randomly presented but organized in a meaningful order which illustrates gradually more concepts, and gradually more complex ones. Here, we formalize such training strategies in the context of machine learning, and call them "curriculum learning". In the context of recent research studying the difficulty of training in the presence of non-convex training criteria (for deep deterministic and stochastic neural networks), we explore curriculum learning in various set-ups. The experiments show that significant improvements in generalization can be achieved. We hypothesize that curriculum learning has both an effect on the speed of convergence of the training process to a minimum and, in the case of non-convex criteria, on the quality of the local minima obtained: curriculum learning can be seen as a particular form of continuation method (a general strategy for global optimization of non-convex functions).

1. Introduction

Humans need about two decades to be trained as fully functional adults of our society. That training is highly organized, based on an education system and a curriculum which introduces different concepts at different times, exploiting previously learned concepts to ease the learning of new abstractions. By choosing which examples to present and in which order to present them to the learning system, one can *guide* training and remarkably increase the speed at which learning can occur. This idea is routinely exploited in *animal training* where it is called **shaping** (Skinner, 1958; Peterson, 2004; Krueger & Dayan, 2009).

Previous research (Elman, 1993; Rohde & Plaut, 1999; Krueger & Dayan, 2009) at the intersection of cognitive science and machine learning has raised the following question: can machine learning algorithms benefit from a similar training strategy? The idea of training a learning machine with a curriculum can be traced back at least to Elman (1993). The basic idea is to *start small*, learn easier aspects of the task or easier subtasks, and then gradually increase the difficulty level. The experimental results, based on learning a simple grammar with a recurrent network (Elman, 1993), suggested that successful learning of grammatical structure depends, not on innate knowledge of grammar, but on starting with a limited architecture that is at first quite restricted in complexity, but then expands its resources gradually as it learns. Such conclusions are important for developmental psychology, because they illustrate the adaptive value of starting, as human infants do, with a simpler initial state, and then building on that to develop more and more sophisticated representations of structure. Elman (1993) makes the statement that this strategy could make it possible for humans to learn what might otherwise prove to be unlearnable. However, these conclusions have been seriously questioned in Rohde and Plaut (1999). The question of guiding learning of a recurrent neural network for learning a simple language and increasing its capacity along the way was recently revisited from the cognitive perspective (Krueger & Dayan, 2009), providing evidence for faster convergence using a shaping procedure. Similar ideas were also explored in robotics (Sanger, 1994), by gradually making the learning task more difficult.

We want to clarify when and why a curriculum or

Appearing in *Proceedings of the 26th International Conference on Machine Learning*, Montreal, Canada, 2009.

Curriculum Learning

- Insight from cognitive science
- Machine learning algorithms can benefit from a similar training strategy
- Learning from easier aspects of the task, and gradually increase the difficulty level
- Expected two advantages:
  - Help find a better local minima (as a regularizer)
  - Speed the convergence of training towards the global minimum (for convex problem)
- Basic steps:
  - Sort samples according to certain “easiness” measure
  - Gradually add samples into training from easy to complex

Self-paced Learning

M. P. Kumar, B. Packer, and D. Koller. Self-paced learning for latent variable models. In *NIPS, pages 1189–1197, 2010.*

Self-Paced Learning for Latent Variable Models

M. Pawan Kumar Ben Packer Daphne Koller
Computer Science Department
Stanford University
{pawan,bpacker,koller}@cs.stanford.edu

Abstract

Latent variable models are a powerful tool for addressing several tasks in machine learning. However, the algorithms for learning the parameters of latent variable models are prone to getting stuck in a bad local optimum. To alleviate this problem, we build on the intuition that, rather than considering all samples simultaneously, the algorithm should be presented with the training data in a *meaningful* order that facilitates learning. The order of the samples is determined by how *easy* they are. The main challenge is that typically we are not provided with a readily computable measure of the easiness of samples. We address this issue by proposing a novel, iterative *self-paced learning* algorithm where each iteration simultaneously selects easy samples and learns a new parameter vector. The number of samples selected is governed by a weight that is annealed until the entire training data has been considered. We empirically demonstrate that the self-paced learning algorithm outperforms the state of the art method for learning a latent structural SVM on four applications: object localization, noun phrase coreference, motif finding and handwritten digit recognition.

1 Introduction

Latent variable models provide an elegant formulation for several applications of machine learning. For example, in computer vision, we may have many 'car' images from which we wish to learn a 'car' model. However, the exact location of the cars may be unknown and can be modeled as latent variables. In medical diagnosis, learning to diagnose a disease based on symptoms can be improved by treating unknown or unobserved diseases as latent variables (to deal with confounding factors). Learning the parameters of a latent variable model often requires solving a non-convex optimization problem. Some common approaches for obtaining an approximate solution include the well-known EM [8] and CCCP algorithms [9, 23, 24]. However, these approaches are prone to getting stuck in a bad local optimum with high training and generalization error.

Machine learning literature is filled with scenarios in which one is required to solve a non-convex optimization task, for example learning latent-variable conditional random fields or deep belief nets. A common approach for avoiding a bad local minimum in these cases is to use multiple runs with random initializations and pick the best solution amongst them (as determined, for example, by testing on a validation set). However, this approach is ad hoc and computationally expensive as one may be required to use several runs to obtain an accurate solution. Bengio *et al.* [3] recently proposed an alternative method for training with non-convex objectives, called curriculum learning. The idea is inspired by the way children are taught: start with easier concepts (for example, recognizing objects in simple scenes where an object is clearly visible) and build up to more complex ones (for example, cluttered images with occlusions). Curriculum learning suggests using the easy samples first and gradually introducing the learning algorithm to more complex ones. The main challenge in using the curriculum learning strategy is that it requires the identification of easy and hard samples in a given training dataset. However, in many real-world applications, such a ranking of training samples may be onerous or conceptually difficult for a human to provide; andxo even if this additional human supervision can be provided, what is intuitively "easy" for a human may not match what is easy for the algorithm in the feature and hypothesis space employed for the given application.

To alleviate these deficiencies, we introduce *self-paced learning*. In the context of human education, self-paced learning refers to a system where the curriculum is determined by the pupil's abilities rather than being fixed by a teacher. We build on this intuition for learning latent variable models by

Self-paced Learning

- Model:

$$\min_{\mathbf{w},\mathbf{v}\in[0,1]^n} \sum_{i=1}^{n} v_i \, L(f(\mathbf{x}_i;\mathbf{w}), y_i) + \gamma g(\mathbf{w}) - \lambda \|\mathbf{v}\|_1$$

- Algorithm: Alternative search
  - Fix w:

$$v_i = \begin{cases} 1, & L(f(\mathbf{x}_i;\mathbf{w}), y_i) \le \lambda, \\ 0, & \text{otherwise} \end{cases}.$$

  - Fix v: A standard classification problem.

Four key words

- Machine Learning
- Cognitive Science
- Self-paced Learning
- Big Data (Video/Multimedia)

SPL Regularizer

- Koller's SPL model:

$$\min_{\mathbf{w},\mathbf{v}\in[0,1]^n} \sum_{i=1}^{n} v_i \, L(f(\mathbf{x}_i;\mathbf{w}), y_i) + \gamma g(\mathbf{w}) - \lambda \|\mathbf{v}\|_1$$

- v's value is determined by a SPL regularizer:

$$\arg \min_{\mathbf{v}\in[0,1]^n} \sum_{i=1}^{n} v_i l_i + f(\mathbf{v}, \lambda)$$

SPL Regularizer

$$\arg\min_{\mathbf{v}\in[0,1]^n} \sum_{i=1}^{n} v_i l_i + f(\mathbf{v}, \lambda)$$

- Axiom for self-paced regularizer:

DEFINITION (Self-paced Regularizer): Suppose that v denotes a weight variable, l is the loss, and λ is the learning pace parameter, $f(v, \lambda)$ is called a self-paced regularizer, if:

- $f(v, \lambda)$ is convex with respect to $v \in [0,1]$;
- $v^*(\lambda, l)$ is monotonically decreasing with respect to l, and it holds that $\log_{l\to 0} v^*(\lambda, l) = 1$, $\log_{l\to\infty} v^*(\lambda, l) = 0$;
- $v^*(\lambda, l)$ is monotonically increasing with respect to λ, and it holds that $\log_{\lambda\to 0} v^*(\lambda, l) = 0$, $\log_{\lambda\to\infty} v^*(\lambda, l) = 1$, where $v^*(\lambda, l) = \arg\min_{v\in[0,1]} vl + f(v, \lambda)$.

(Lu Jiang, Deyu Meng et al. ACM MM, 2014; Qian Zhao, Deyu Meng, et al. AAAI, 2015)

SPL Regularizer

$$\arg\min_{\mathbf{v}\in[0,1]^n} \sum_{i=1}^{n} v_i l_i + f(\mathbf{v},\lambda)$$

- Axiom for self-paced regularizer :

DEFINITION (Self-paced Regularizer): Suppose that v denotes a weight variable, l is the loss, and λ is the learning pace parameter, $f(v,\lambda)$ is called a self-paced regularizer, if:

- $f(v,\lambda)$ is convex with respect to $v \in [0,1]$;
- $v^*(\lambda,l)$ is monotonically decreasing with respect to l, and it holds that $\log_{l\to 0} v^*(\lambda,l) = 1$, $\log_{l\to\infty} v^*(\lambda,l) = 0$;
- $v^*(\lambda,l)$ is monotonically increasing with respect to λ, and it holds that $\log_{\lambda\to 0} v^*(\lambda,l) = 0$, $\log_{\lambda\to\infty} v^*(\lambda,l) = 1$, where $v^*(\lambda,l) = \arg\min_{v\in[0,1]} vl + f(v,\lambda)$.

Favors Easy Samples

(Lu Jiang, Deyu Meng et al. ACM MM, 2014; Qian Zhao, Deyu Meng, et al. AAAI, 2015)

SPL Regularizer

$$\arg\min_{\mathbf{v}\in[0,1]^n} \sum_{i=1}^{n} v_i l_i + f(\mathbf{v},\lambda)$$

- Axiom for self-paced regularizer :

DEFINITION (Self-paced Regularizer): Suppose
denotes a weight variable, l is the loss, and λ is the
pace parameter, $f(v,\lambda)$ is called a self-paced regularize

- $f(v,\lambda)$ is convex with respect to $v \in [0,1]$;
- $v^*(\lambda,l)$ is monotonically decreasing with respect t
  it holds that $\log_{l\to 0} v^*(\lambda,l) = 1$, $\log_{l\to\infty} v^*(\lambda,l) =$
- $v^*(\lambda,l)$ is monotonically increasing with respect to λ, and it holds that $\log_{\lambda\to 0} v^*(\lambda,l) = 0$, $\log_{\lambda\to\infty} v^*(\lambda,l) = 1$,
  where $v^*(\lambda,l) = \arg\min_{v\in[0,1]} vl + f(v,\lambda)$.

When the model is young, use less samples; when the model is mature, use more.

(Lu Jiang, Deyu Meng et al. ACM MM, 2014; Qian Zhao, Deyu Meng, et al. AAAI, 2015)

SPL Regularizer

$$\arg\min_{\mathbf{v}\in[0,1]^n} \sum_{i=1}^{n} v_i l_i + f(\mathbf{v}, \lambda)$$

- Axiom for self-paced regularizer:

DEFINITION (Self-paced Regularizer): Suppose that v denotes a weight variable, l is the loss, and λ is the learning pace parameter, $f(v, \lambda)$ is called a self-paced regularizer, if:

- $f(v, \lambda)$ is convex with respect to $v \in [0,1]$; → Convex
- $v^*(\lambda, l)$ is monotonically decreasing with respect to l, and it holds that $\log_{l\to 0} v^*(\lambda, l) = 1$, $\log_{l\to\infty} v^*(\lambda, l) = 0$;
- $v^*(\lambda, l)$ is monotonically increasing with respect to λ, and it holds that $\log_{\lambda\to 0} v^*(\lambda, l) = 0$, $\log_{\lambda\to\infty} v^*(\lambda, l) = 1$,

where $v^*(\lambda, l) = \arg\min_{v\in[0,1]} vl + f(v, \lambda)$.

(Lu Jiang, Deyu Meng et al. ACM MM, 2014; Qian Zhao, Deyu Meng, et al. AAAI, 2015)

SPL Regularizer

- Some soft extensions for self-paced regularizer:

Linear Soft Weighting:

$$f(\mathbf{v},\lambda)=\lambda(\frac{1}{2}\|\mathbf{v}\|^2-\sum_{i=1}^{n}v_i) \quad \Rightarrow \quad v_i^*(\lambda,l)=\begin{cases}-\frac{l}{\lambda}+1, l<\lambda\\ 0, \quad l\geq\lambda\end{cases}$$

Logarithmic Soft Weighting:

$$f(\mathbf{v},\lambda)=\sum_{i=1}^{n}(1-\lambda)v_i-\frac{(1-\lambda)^{v_i}}{\log(1-\lambda)} \quad \Rightarrow \quad v_i^*(\lambda,l)=\begin{cases}\frac{\log(l+1-\lambda)}{\log(1-\lambda)}, l<\lambda\\ 0, \quad l\geq\lambda\end{cases}$$

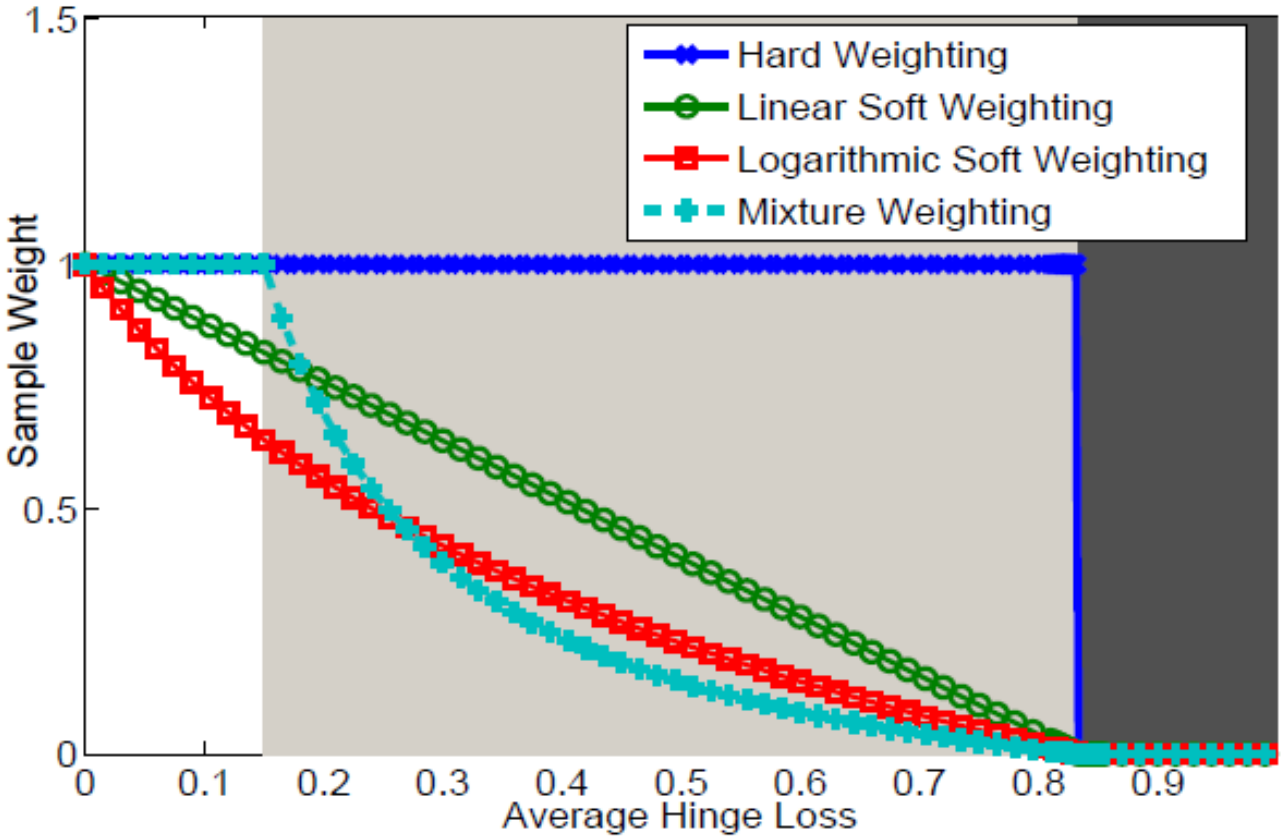
Mixture Weighting:

$$f(\mathbf{v},\lambda,\gamma)=\sum_{i=1}^{n}\frac{\gamma^2}{v_i+\gamma/\lambda} \quad \Rightarrow \quad v_i^*(\lambda,\gamma,l)=\begin{cases}1, l<(\frac{\lambda\gamma}{\lambda+\gamma})^2\\ 0, \quad l\geq\lambda^2\\ \gamma(1/\sqrt{l}-1/\lambda), \text{otherwise}\end{cases}$$

(Lu Jiang, Deyu Meng et al. ACM MM, 2014; Qian Zhao, Deyu Meng, et al. AAAI, 2015)

SPL Regularizer

- Some soft extensions for self-paced regularizer



(Lu Jiang, Deyu Meng et al. ACM MM, 2014; Qian Zhao, Deyu Meng, et al. AAAI, 2015)

SPL Model:

$$\min_{\mathbf{w},\mathbf{v}\in[0,1]^n} \mathbb{E}(\mathbf{w},\mathbf{v},\lambda) = \sum_{i=1} v_i L(y_i, g(\mathbf{x}_i,\mathbf{w})) + f(\mathbf{v};\lambda)$$

SPL Algorithm:

Algorithm 1: SPL Algorithm.

input : Input dataset $\mathcal{D}$, pace parameter $\mu > 1$, and self-paced function f.
output: Model parameter $\mathbf{w}$

Initialize $\mathbf{w}^*, \lambda$; // assign the starting value
while *not converged* **do**
  while *not converged* **do**
    Update $\mathbf{v}^* = \arg\min_{\mathbf{v}} \mathbb{E}(\mathbf{w}^*, \mathbf{v}; \lambda)$;
    Update $\mathbf{w}^* = \arg\min_{\mathbf{w}} \mathbb{E}(\mathbf{w}, \mathbf{v}^*; \lambda)$;
  end
  $\lambda \leftarrow \mu\lambda$; // update the learning pace
end
return $\mathbf{w} = \mathbf{w}^*$;

More extensions

- SPaR: Lu Jiang, Deyu Meng, Qian Zhao et al. *ACM MM*, 2014.
  - **Soft extension on MED Ex0 problem**
- SPMF: Qian Zhao, Deyu Meng, Lu Jiang et al. *AAAI*, 2015.
  - **Mixture extension on matrix factorization**
- SPLD: Lu Jiang, Deyu Meng, Shoou-I Yu et al. *NIPS*, 2014.
  - **Diversity extension on action recognition**
- SPCL: Lu Jiang, Deyu Meng, Teruko Mitamura et al. *AAAI*. 2015.
  - **Curriculum extension on MED and matrix factorization**
- SP-MIL: Dingwen Zhang, Deyu Meng, Junwei Han. *ICCV*. 2015.
  - **Weakly supervised extension on co-saliency detection**
- MOSPL: Submitted to AAAI 2015 (Cooperated with Maoguo Gong)
  - **Multi-objective extension on action recognition**
- ASPL: In process (Cooperated with Liang Lin, Wangmeng Zuo)
  - **Active curriculum extension on face identification**

Some successful applications

- State-of-the-art performance on
  - Web Query dataset
  - Hollywood2 dataset
  - Olympic Sports dataset
  - iCoseg dataset
  - MSRC dataset
  - Trecvid MED Ex0test 2013
  - Trecvid MED Ex0test 2014

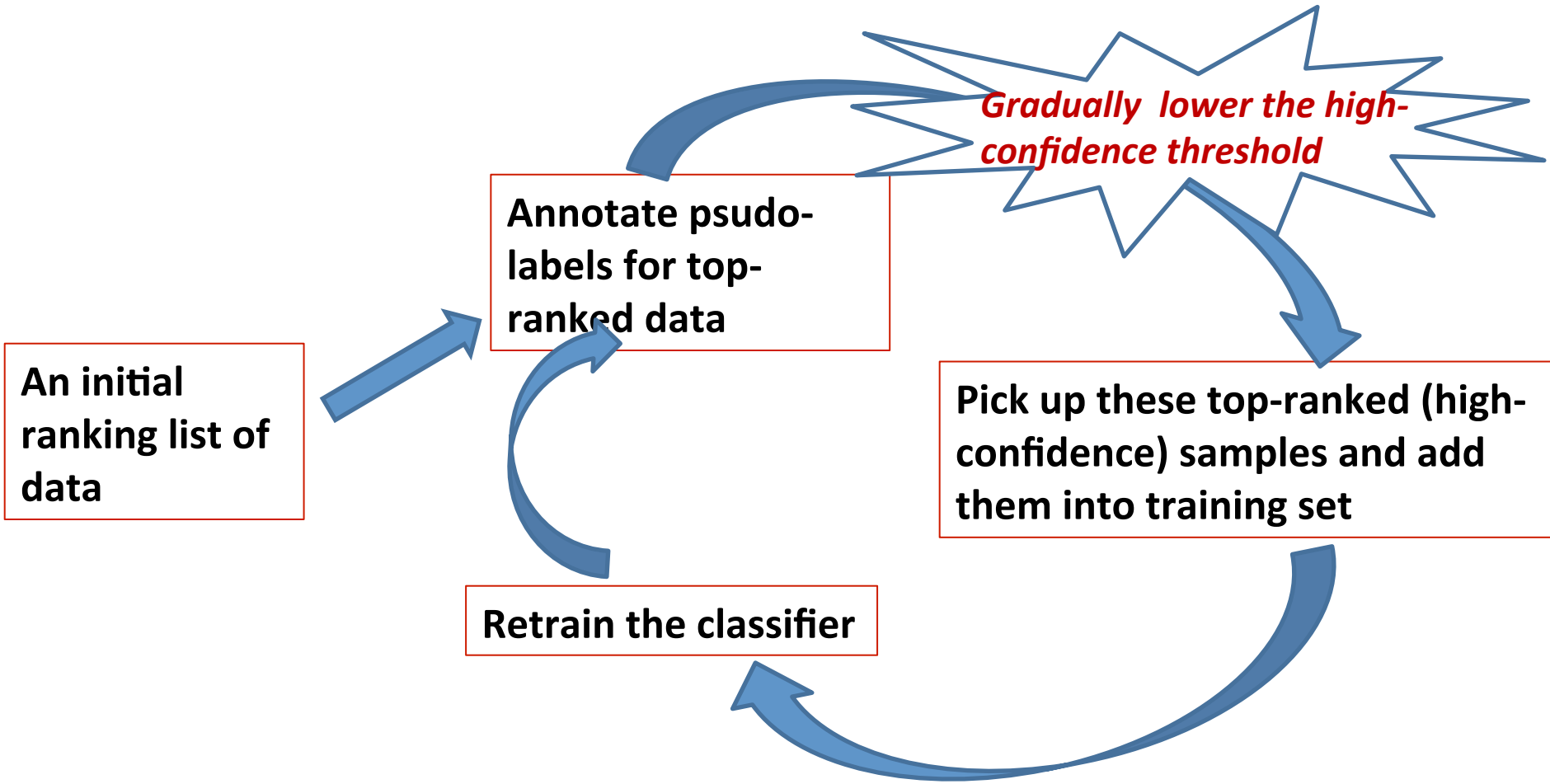
Four key words

- Machine Learning
- Cognitive Science
- Self-paced Learning
- Big Data (Video/Multimedia)

Zero-Example Search

- Zero-Example Search (also known as Ex0) represents a multimedia search condition where zero relevant examples are provided
  - Content-based search
- An example: TRECVID Multimedia Event Detection (MED) competition. The task is very challenging
  - Detect every-day event in Internet
    - Birthday party
    - Wedding ceremony
    - Changing a vehicle tire

Informedia@CMU 2013 Pipeline for Ex0



SPaR

- Model

$$\min_{\Theta_1,\ldots,\Theta_m,\mathbf{y},\mathbf{v}} \mathbb{E}(\Theta_1,\ldots,\Theta_m,\mathbf{v},\mathbf{y};C,\lambda) =$$

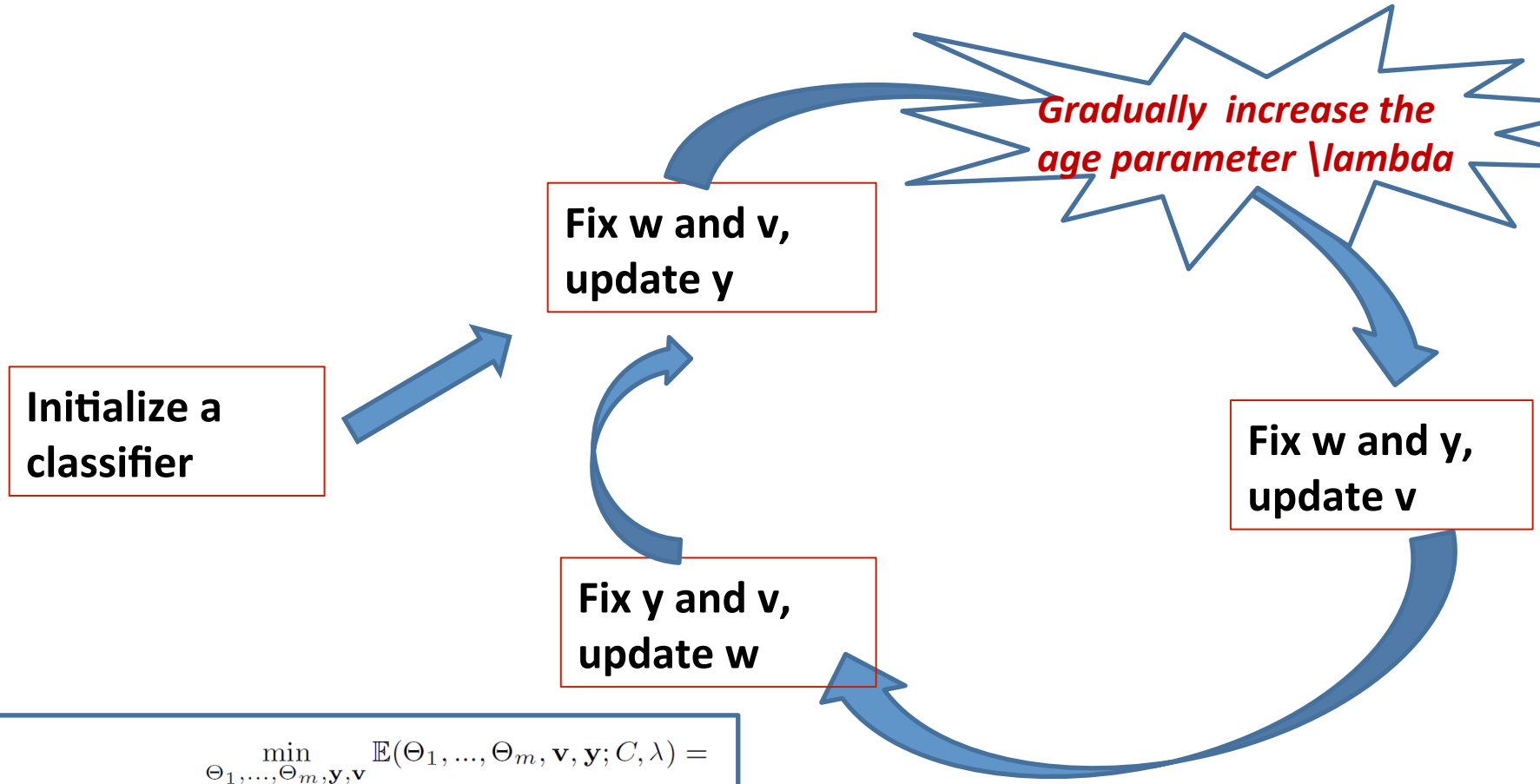
$$\min_{\substack{\mathbf{y},\mathbf{v},\mathbf{w}_1,\ldots,\mathbf{w}_m,\\ b_1,\ldots,b_m,\{\ell_{ij}\}}} \sum_{j=1}^{m}\frac{1}{2}\|\mathbf{w}_j\|_2^2 + C\sum_{i=1}^{n}\sum_{j=1}^{m} v_i\ell_{ij} + mf(\mathbf{v};\lambda)$$

$$\text{s.t. } \forall i,\forall j, y_i(\mathbf{w}_j^T\phi(\mathbf{x}_{ij}) + b_j) \geq 1-\ell_{ij}, \ell_{ij}\geq 0$$

$$\mathbf{y}\in\{-1,+1\}^n, \mathbf{v}\in[0,1]^n$$

(Lu Jiang, Deyu Meng et al. ACM MM, 2014)

ASS for solving SPaR model



$$\min_{\Theta_1,\ldots,\Theta_m,\mathbf{y},\mathbf{v}} \mathbb{E}(\Theta_1,\ldots,\Theta_m,\mathbf{v},\mathbf{y};C,\lambda) =$$

$$\min_{\substack{\mathbf{y},\mathbf{v},\mathbf{w}_1,\ldots,\mathbf{w}_m,\\ b_1,\ldots,b_m,\{\ell_{ij}\}}} \sum_{j=1}^{m}\frac{1}{2}\|\mathbf{w}_j\|_2^2 + C\sum_{i=1}^{n}\sum_{j=1}^{m} v_i\ell_{ij} + mf(\mathbf{v};\lambda)$$

$$\text{s.t. } \forall i,\forall j, y_i(\mathbf{w}_j^T\phi(\mathbf{x}_{ij}) + b_j) \geq 1-\ell_{ij}, \ell_{ij}\geq 0$$

$$\mathbf{y}\in\{-1,+1\}^n, \mathbf{v}\in[0,1]^n$$

Reranking is a self-paced learning process!

- On TRECVID MED 2013 Ex0 dataset

Method	NIST's split	10 splits
Without Reranking	3.9	4.9 ± 1.6
Rocchio	5.7	7.4 ± 2.2
Relevance Model	2.6	3.4 ± 1.0
CPRF	6.4	8.3 ± 1.8
Learning to Rank	3.4	4.2 ± 1.4
MMPRF	10.1	13.6 ± 2.4
SPaR	**12.9**	$\mathbf{15.3 \pm 2.6}$

- On Web Query dataset

Method	MAP	MAP@100
Without Reranking [17]	0.569	0.431
CPRF [38]	0.658	-
Random Walk [10]	0.616	-
Bayesian Reranking [33, 32]	0.658	0.529
Preference Learning Model [32]	-	0.534
BVLS [26]	0.670	-
Query-Relative(visual) [17]	0.649	-
Supervised Reranking [39]	0.665	-
SPaR	**0.672**	**0.557**

(Lu Jiang, Deyu Meng et al. ACM MM, 2014)

$$\min_{\mathbf{w},\mathbf{v}\in[0,1]^n} \mathbb{E}(\mathbf{w},\mathbf{v},\lambda) = \sum_{i=1} v_i L(y_i, g(\mathbf{x}_i,\mathbf{w})) + f(\mathbf{v};\lambda)$$

Theoretical insight of SPL is still entirely unknown

- Why it's effective in outlier/heavy noise cases
- Where it converges to
- What's the theoretical insight of SPL working mechanism

Four key words

- Machine Learning
- Cognitive Science
- Self-paced Learning
- Big Data (Video/Multimedia)

Majorization Minimization Algorithm

$$\min_{\mathbf{w}} \mathbf{F}(\mathbf{w})$$

Majorization Step: Substitute $\mathbf{F}(\mathbf{w})$ by a surrogate function $Q(\mathbf{w}|\mathbf{w}^k)$ such that

$$F(\mathbf{w}) \leq Q(\mathbf{w}|\mathbf{w}^k)$$

with equality holding at $\mathbf{w} = \mathbf{w}^k$.

Minimization Step: Obtain the next parameter estimate $\mathbf{w}^{k+1}$ by solving the following minimization problem:

$$\mathbf{w}^{k+1} = \arg\min_{\mathbf{w}} Q(\mathbf{w}|\mathbf{w}^k).$$

- ***An effective technique utilized in optimization and machine learning!***

Latent loss function under SPL:

$$F_\lambda(\ell) = \int_0^\ell v^*(\lambda; l)dl$$

$$v^*(\lambda, l) = \arg\min_{v \in [0,1]} vl + f(v, \lambda).$$

Latent loss function under SPL: $$F_\lambda(\ell) = \int_0^\ell v^*(\lambda; l)dl$$

Theorem 1 *For $v^*(\lambda;\ell)$ conducted by an SP-regularizer and $F_\lambda(\ell)$ calculated by (5), given a fixed $\mathbf{w}^*$, it holds that:*

$$F_\lambda(\ell(\mathbf{w})) \leq Q_\lambda(\mathbf{w}|\mathbf{w}^*) = F_\lambda(\ell(\mathbf{w}^*)) + v^*(\lambda;\ell(\mathbf{w}^*))(\ell(\mathbf{w})-\ell(\mathbf{w}^*)).$$

Latent SPL objective: $$\sum_{i=1}^{n} F_\lambda(l_i(\mathbf{w})) \leq \sum_{i=1}^{n} Q_\lambda^{(i)}(\mathbf{w}|\mathbf{w}^*)$$

$$Q_\lambda^{(i)}(\mathbf{w}|\mathbf{w}^*) = F_\lambda(l_i(\mathbf{w}^*)) + v^*(\lambda; l_i(\mathbf{w}^*))(l_i(\mathbf{w})-l_i(\mathbf{w}^*)$$

ASS algorithm for SPL ***Exactly is***
MM Algorithm the latent SPL objective

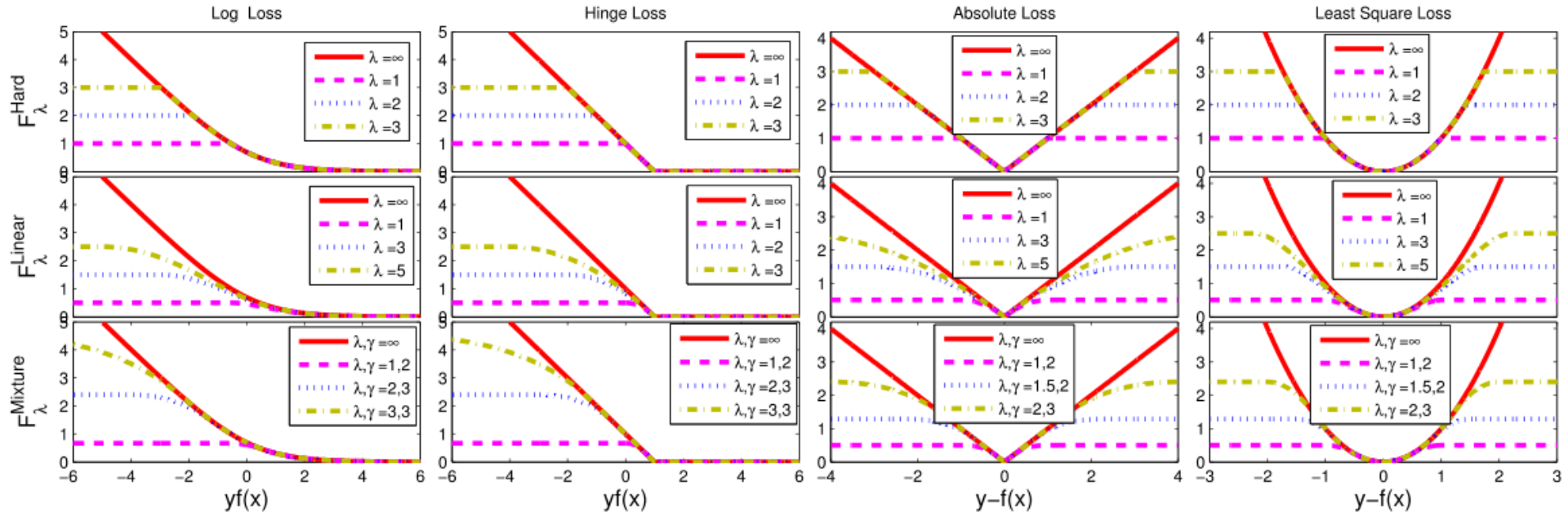
Let's see what hides behind this latent SPL objective:

$$F_\lambda(l) = \int_0^l v^*(\lambda;\ell)dl$$

$$F_\lambda^H(l) = \begin{cases} l, & l < \lambda, \\ \lambda, & l \geq \lambda; \end{cases}$$

$$F_\lambda^L(l) = \begin{cases} l - l^2/2\lambda, & l < \lambda, \\ \lambda/2, & l \geq \lambda; \end{cases}$$

$$F_{\lambda,\gamma}^M(l) = \begin{cases} l, & l < \frac{1}{(1/\lambda+1/\gamma)^2}, \\ \gamma(2\sqrt{l} - l/\lambda) - \frac{\gamma}{(1/\lambda+1/\gamma)}, & \frac{1}{(1/\lambda+1/\gamma)^2} \leq l < \lambda^2 \\ \gamma(\lambda - \frac{1}{1/\lambda+1/\gamma}), & l \geq \lambda^2. \end{cases}$$



Let's see some known non-convex penalties:

Capped-norm penalty: $p_{\gamma,\lambda}^{CAP}(t) = \gamma \min(|t|, \lambda),\ \lambda > 0;$

$$\text{MCP: } p_{\gamma,\lambda}^{MCP}(t) = \begin{cases} \gamma(|t| - \frac{t^2}{2\gamma\lambda}), \text{ if } |t| < \gamma\lambda \\ \frac{\gamma^2\lambda}{2}, \text{ if } |t| \geq \gamma\lambda \end{cases};$$

$$\text{SCAD: } p_{\gamma,\lambda}^{SCAD}(t) = \begin{cases} \lambda|t|, \text{ if } |t| \leq \lambda \\ -\frac{t^2 - 2\gamma\lambda|t| + \lambda^2}{2(\gamma-1)}, \text{ if } \lambda < |t| \leq \gamma\lambda \\ \frac{(\gamma+1)\lambda^2}{2}, \text{ if } |t| \geq \gamma\lambda \end{cases}$$

The research on non-convex penalty/loss attracts increasing attention in statistics and machine learning!

Let's see some known non-convex penalties:

Capped-norm penalty: $p_{\gamma,\lambda}^{CAP}(t) = \gamma \min(|t|, \lambda),\ \lambda > 0;$

MCP: $p_{\gamma,\lambda}^{MCP}(t) = \begin{cases} \gamma(|t| - \frac{t^2}{2\gamma\lambda}), \text{ if } |t| < \gamma\lambda \\ \frac{\gamma^2\lambda}{2}, \text{ if } |t| \geq \gamma\lambda \end{cases};$

SCAD: $p_{\gamma,\lambda}^{SCAD}(t) = \begin{cases} \lambda|t|, \text{ if } |t| \leq \lambda \\ -\frac{t^2 - 2\gamma\lambda|t| + \lambda^2}{2(\gamma-1)}, \text{ if } \lambda < |t| \leq \gamma\lambda \\ \frac{(\gamma+1)\lambda^2}{2}, \text{ if } |t| \geq \gamma\lambda \end{cases}$

$$F_\lambda^H(l) = \begin{cases} l, & l < \lambda, \\ \lambda, & l \geq \lambda; \end{cases}$$

$$F_\lambda^L(l) = \begin{cases} l - l^2/2\lambda, & l < \lambda, \\ \lambda/2, & l \geq \lambda; \end{cases}$$

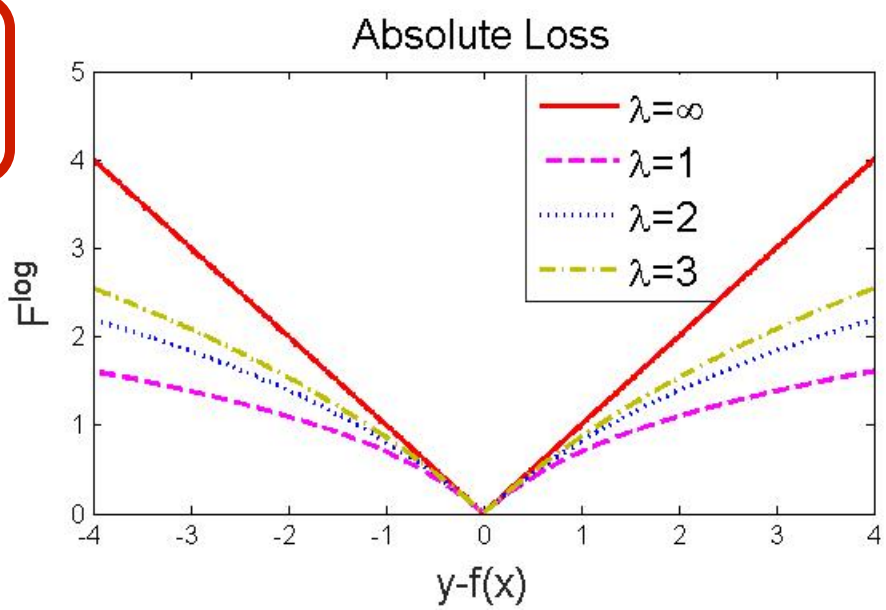
$$F_{\lambda,\gamma}^M(l) = \begin{cases} l, & l < \frac{1}{(1/\lambda+1/\gamma)^2}, \\ \gamma(2\sqrt{l} - l/\lambda) - \frac{\gamma}{(1/\lambda+1/\gamma)}, & \frac{1}{(1/\lambda+1/\gamma)^2} \leq l < \lambda^2 \\ \gamma(\lambda - \frac{1}{1/\lambda+1/\gamma}), & l \geq \lambda^2. \end{cases}$$

- Hard SPL ***exactly*** complies with the Capped-norm penalty
- Linear SPL ***exactly*** complies with the MCP penalty
- Mixture SPL is ***very similar*** to the SCAD penalty

- Such theoretical understanding constructs a natural connection between non-convex penalties(losses) and SPL regimes
  - SPL provides more rational choices for non-convex penalty/loss
  - More SPL formats for multiple known non-convex penalties can be found

LOG

$$F_\lambda(l) = LOG_\lambda(l) = \lambda \log(l/\lambda + 1)$$



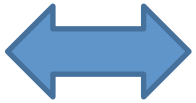
LOG SP-regularizer

$$f(v, \lambda) = KL(v, \lambda) = -\lambda \ln v$$

- Such theoretical understanding constructs a natural connection between non-convex penalties(losses) and SPL regimes
  - SPL provides more rational choices for non-convex penalty/loss
  - More SPL formats for multiple known non-convex penalties can be found

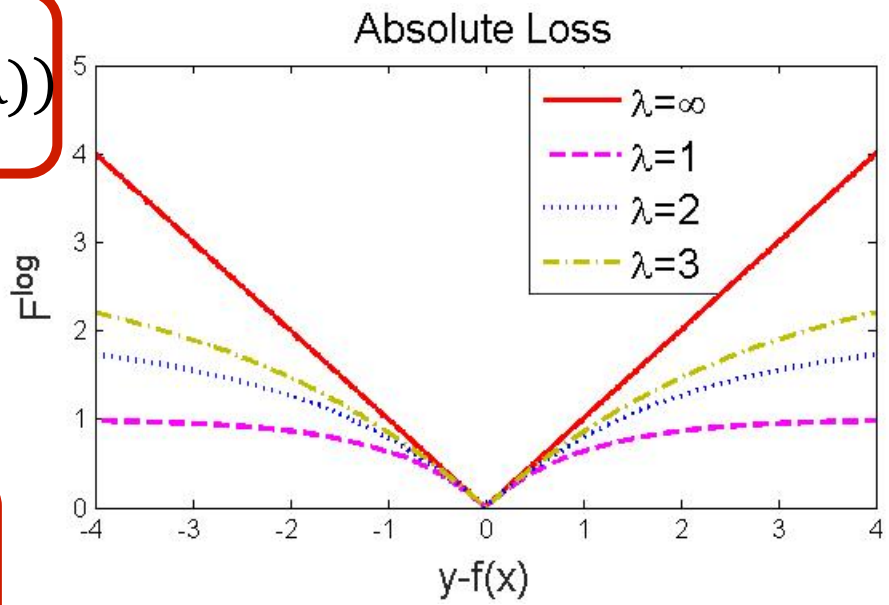
EXP

$$F_\lambda(l) = EXP_\lambda(l) = \lambda(1 - \exp(-l/\lambda))$$



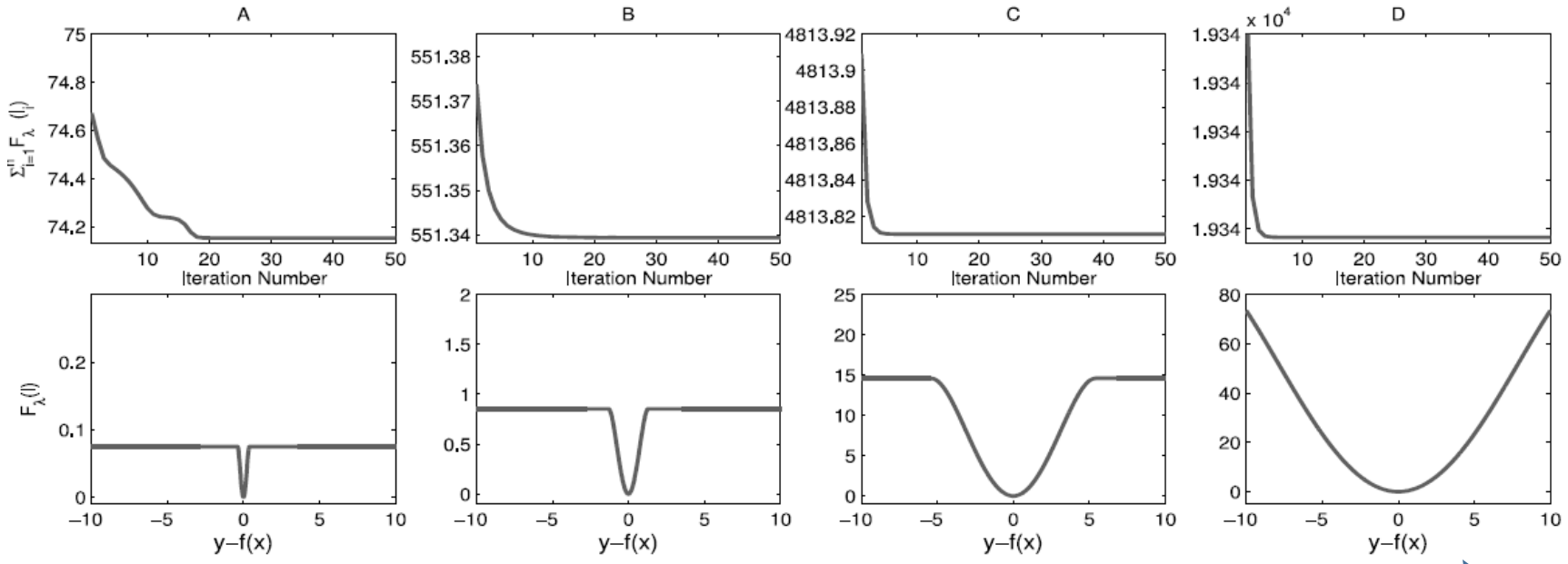
EXP SP-regularizer

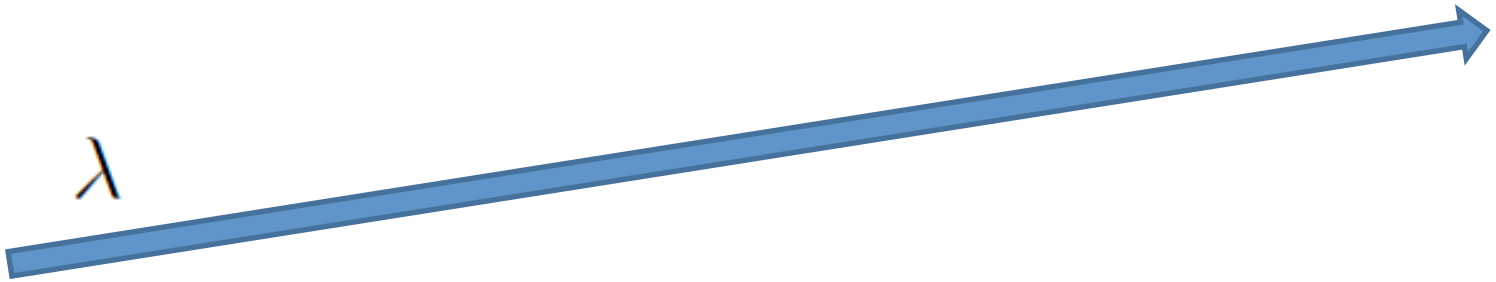
$$f(v,\lambda) = KL(v,\lambda) = \lambda v \ln v$$



Working mechanism under SPL

- Linear SPL performance demonstration in a synthetic regression problem containing outliers and noises





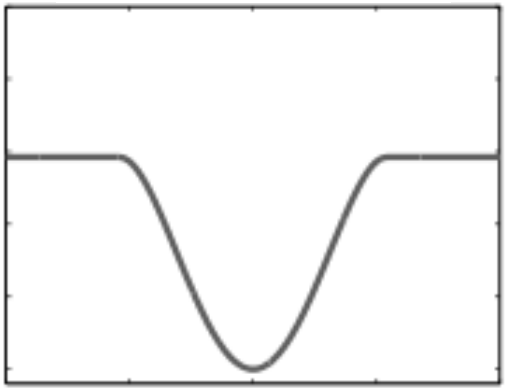
(Deyu Meng et al. submitted to AAAI, 2016)

A natural problem is:

Why not directly optimize the latent SPL objective, while we prefer to use SPL regime instead?

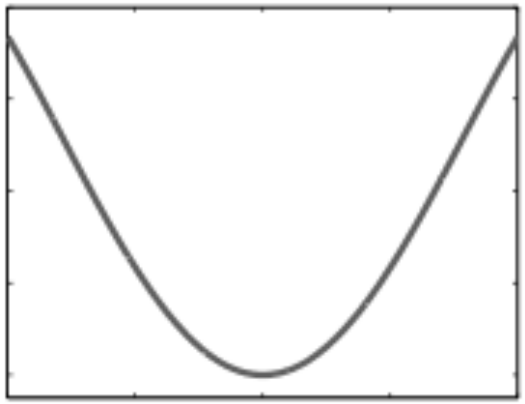
Superiority of SPL: Non-convex Optimization

$$\min_{\mathbf{w}} \sum_{i=1}^{n} F_{\lambda}(l_i(\mathbf{w}))$$

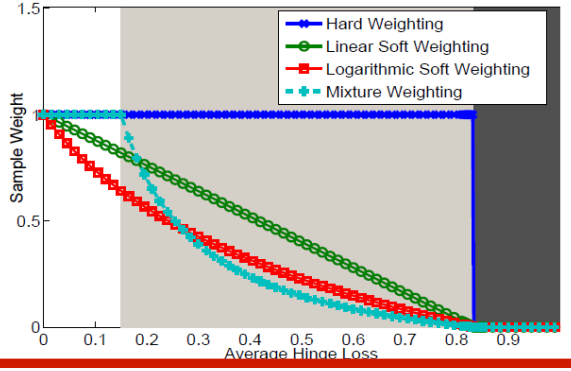


Decompose

$$\min_{\mathbf{w}} \sum_{i=1} v_i L(y_i, g(\mathbf{x}_i, \mathbf{w}))$$



Weighted Easy Loss Minimization (generally convex)



SP weights Updating Problem (Convex)

Useful sample loss/ importance prior knowledge can be easily embedded

$$\min_{\mathbf{v}\in[0,1]^n} \sum_{i=1} v_i L(y_i, g(\mathbf{x}_i, \mathbf{w})) + f(\mathbf{v}; \lambda)$$

Some useful sample loss/importance priors:

- Spatial/temporal smoothness prior:

- Partial order prior:

- Diversity prior:

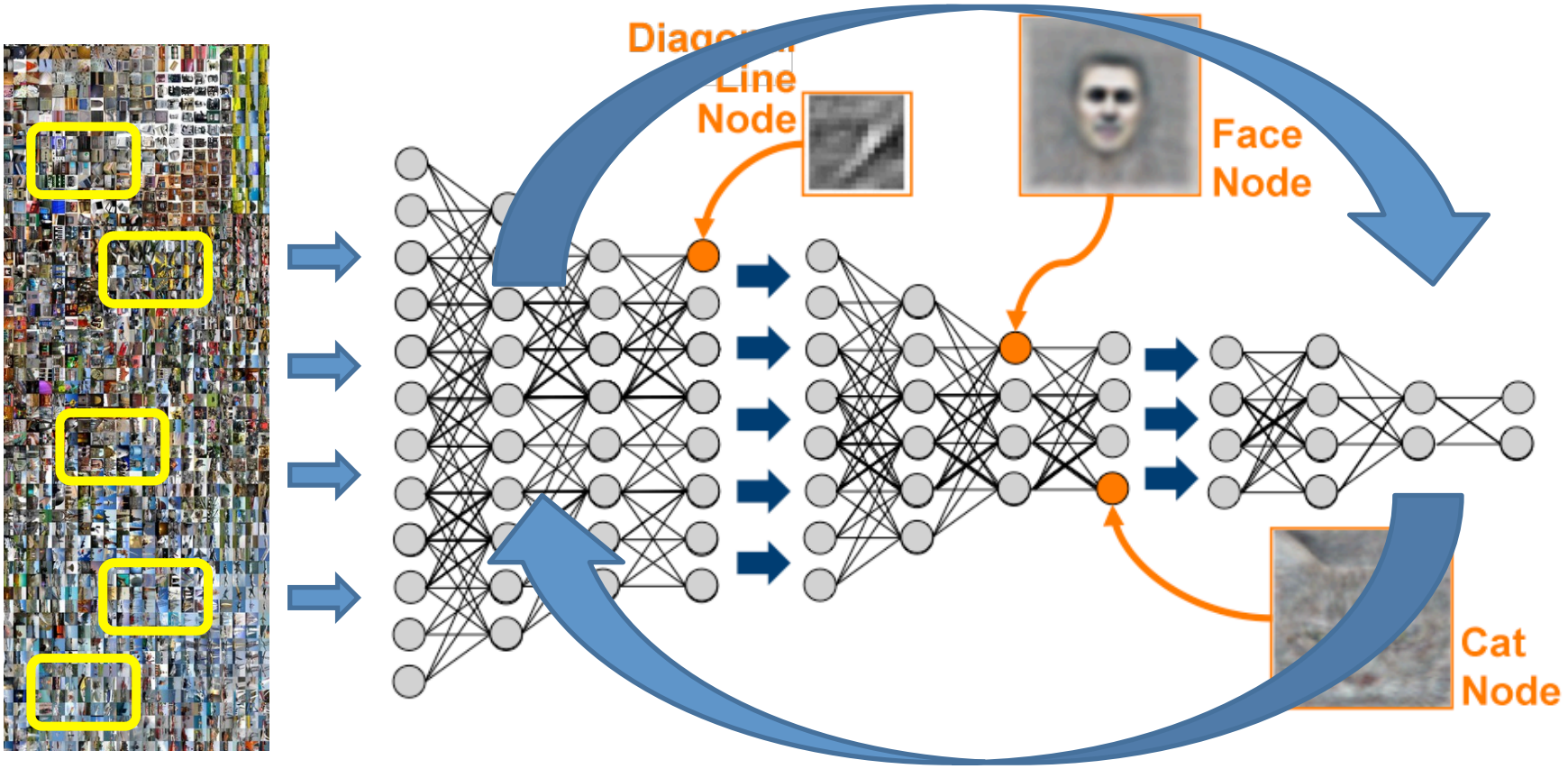
Easy encoding:

- Spatial/temporal smoothness prior: $\mathbf{v}^T\mathbf{L}\mathbf{v}$
- Partial order prior: $v_i > v_j$
- Diversity prior: $-\|\mathbf{v}\|_{2,1}$, $-\|\mathbf{v}\|_{0.5,1}$

- SPCL on MED Ex0: Partial order
  - SPCL, AAAI 2015
- SPLD on action recognition: Diversity
  - SPLD, NIPS 2014
- SP-MIL on co-saliency detection: Diversity + Spatial smoothness
  - SP-MIL, ICCV 2015

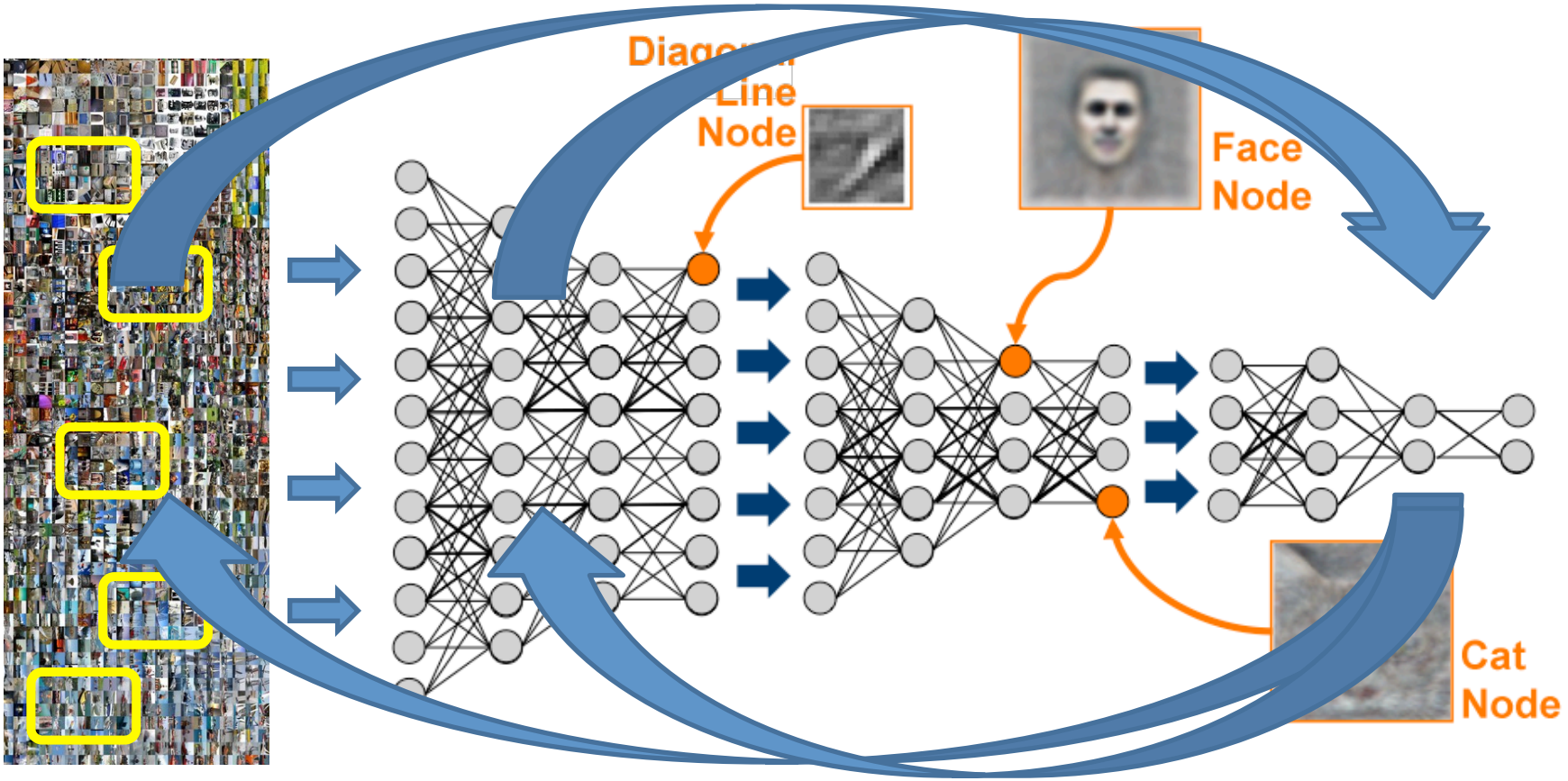
➢ ***Such utilization of loss priors greatly help alleviate the local minimum issue in non-convex penalty/loss optimization problems!***

Superiority of SPL: Data Screening



Superiority of SPL: Data Screening

- Integrating data screening process into automatic network training!
- SPL provides a sound guidance for this aim, both empirically and theoretically
- Then all ML elements can be integrated into E2E DNN consideration



Four key words

- Machine Learning
- Cognitive Science
- Self-paced Learning
- Big Data (Video/Multimedia)

- CL:
  - **Pros:** Flexible to incorporate prior knowledge from various sources
  - **Cons:** The curriculum design is determined independently of the subsequent learning; there is no guarantee that the predetermined curriculum can even lead to a converged solution

- SPL:
  - **Pros:** Hard to incorporating prior knowledge into learning, rendering it prone to overfitting
  - **Cons:** Concise formulations; automatically learning process

- CL: Instructor-driven
  - **Pros:** Flexible to incorporate prior knowledge from various sources
  - **Cons:** The curriculum design is determined independently of the subsequent learning; there is no guarantee that the predetermined curriculum can even lead to a converged solution

- SPL: Student-driven
  - **Pros:** Hard to incorporating prior knowledge into learning, rendering it prone to overfitting
  - **Cons:** Concise formulations; automatically learning process

SPCL

- SPCL: Instructor-student-collaborative

$$\min_{\mathbf{w},\mathbf{v}\in[0,1]^n} \mathbb{E}(\mathbf{w},\mathbf{v},\lambda,\Psi) = \sum_{i=1} v_i L(y_i, g(\mathbf{x}_i,\mathbf{w})) + f(\mathbf{v};\lambda)$$

$$\text{s.t. } \mathbf{v} \in \Psi$$

	CL	SPL	Proposed SPCL
Comparable to human learning	Instructor-driven	Student-driven	Instructor-student collaborative
Curriculum design	Prior knowledge	Learning objective	Learning objective + prior knowledge
Learning schemes	Multiple	Single	Multiple
Iterative training	Heuristic approach	Gradient-based	Gradient-based

(Lu Jiang, Deyu Meng et al. AAAI, 2015)

SPCL

- SPCL: Instructor-student-collaborative

$$\min_{\mathbf{w},\mathbf{v}\in[0,1]^n} \mathbb{E}(\mathbf{w},\mathbf{v},\lambda,\Psi)=\sum_{i=1} v_i L(y_i,g(\mathbf{x}_i,\mathbf{w}))+f(\mathbf{v};\lambda)$$

$$\text{s.t. } \mathbf{v}\in\Psi$$

- An interesting guess is:
  - –Nonconvex optimization corresponds to student learning, which easily stuck to local minimum
  - –Loss prior corresponds to teacher's prior knowledge, which might be significantly useful to help alleviate such local-minimum issue

(Lu Jiang, Deyu Meng et al. AAAI, 2015)

Four key words

- Machine Learning
- Cognitive Science
- Self-paced Learning
- Big Data (Video/Multimedia)

A General Machine learning Framework

$$\min_{f \in \mathcal{F}} \quad l(D, f(w)) \quad + \quad p(w)$$

Loss/likelihood term

Self-paced Learning

- **2014.1** SPaR, *ACM MM*
  - With CMU Group
- **2014.3** SPMF, *AAAI*
  - With Qian Zhao
- **2014.4** SPLD, *NIPS*
  - With CMU Group
- **2014.8** SPCL, *AAAI*
  - With CMU Group
- **2014.9** *TRECVID* competition
  - With CMU Group
- **2014.11** SP-MIL, *ICCV*
  - With Junwei Han & Dingwen Zhang
- **2015.1** ASPL, in process
  - With Liang Lin's Group
- **2015.3** Ex0 system, *ICMR (best paper runner up)*
  - With CMU Group
- **2015.8** SPL Insight, Submitted to *AAAI*
  - With XJTU Group
- **2015.9** SP-McMIL, Submitted to *CVPR*
  - With Junwei Han & Dingwen Zhang

Noise modeling

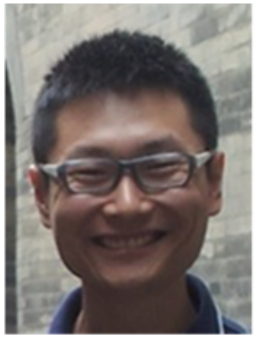
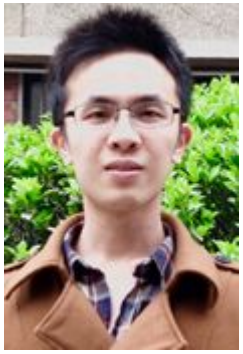
- **2012.10** L1 loss in SPCA, *PR*
  - With XJTU Group
- **2012.11** Laplacian noise in MF, *TNNLS*
  - With XJTU Group
- **2013.1** L1 loss in MF, *AAAI*
  - With XJTU Group
- **2013.3** L1 loss RPCA in Infra image, *TIP*
  - With Chenqiang Gao
- **2013.4** MOG noise in MF, *ICCV*
  - With Fernando De la Torre
- **2013.9** MOG noise in RPCA, *ICML*
  - With XJTU&PolyU Group
- **2014.11** GMD noise in MF, *ICCV (oral)*
  - With XJTU Group
- **2015.5** Non-i.i.d. MoG noise, submitted to CVPR
  - With XJTU Group

Adapt Loss Function to Data

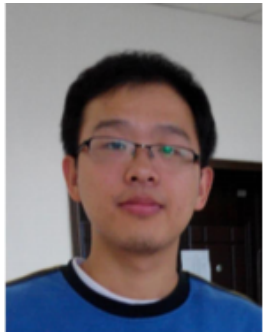
Future work

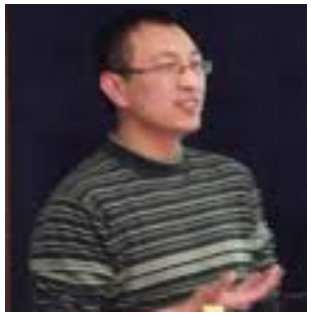
- Theoretical issues
  - Parameter setting, convergence analysis, statistical properties
- Modeling issues
  - More useful SP regularizer formats, integration with more machine learning models
- Application issues
  - Attempts on more computer vision, multimedia, data mining applications

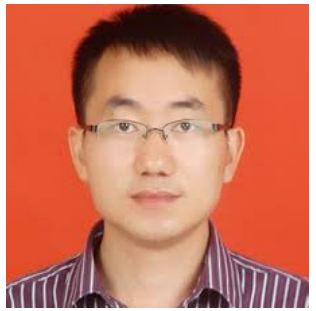

世界上本没有路
但只要努力自己(Self)去走一走(Pace)
看上去似乎好像就有路了
----鲁小迅











Special thanks to my graduate students: *Jiangjun Peng*, *Jie Lu*, *Zongsheng Yue*. An impressive summer holiday in XJTU, right ☺