Online-Batch Strongly Convex Multi Kernel Learning

Francesco Orabona\textsuperscript{1} Luo Jie\textsuperscript{2,3} Barbara Caputo\textsuperscript{2}

\textsuperscript{1}DSI, Università degli Studi di Milano, Milano, Italy
\textsuperscript{2}Idiap Research Institute, Martigny, Switzerland
\textsuperscript{3}EPF Lausanne

23\textsuperscript{rd} IEEE Conference on Computer Vision and Pattern Recognition
Outline

1. Multi Kernel Learning
   - Notation
   - Previous work
   - Sparsity?
   - Dual?

2. OBSCURE
   - A different MKL formulation
   - The algorithm

3. Experimental Results
   - Caltech-101
Outline

1. Multi Kernel Learning
   - Notation
   - Previous work
   - Sparsity?
   - Dual?

2. OBSCURE
   - A different MKL formulation
   - The algorithm

3. Experimental Results
   - Caltech-101
Problem definition

- We are given $N$ training samples $\{x_i, y_i\}_{i=1}^N$, $x_i \in X$ and $y_i \in Y = \{1, \cdots, M\}$, where $M$ is the number of classes.
- We also have $F$ kernels corresponding to different features, e.g. color, shape, etc.
- We want to learn a score function $s(x, y)$ that classifies a sample $x$ as
  \[
  \arg \max_y s(x, y).
  \]
We consider score functions of the form
\[ s(x, y) = \sum_{j=1}^{F} s^j(x, y) \]

Defining joint feature maps \( \phi^j(x, y) \) on data \( X \) and labels \( Y \) [Tsochantaridis et al, 2004].

\[ s^j(x, y) = w^j \cdot \phi^j(x, y), \]

Defining with \( \vec{w} = [w^1, w^2, \cdots, w^F] \), and \( \vec{\phi}(x, y) = [\phi^1(x, y), \cdots, \phi^F(x, y)] \), we have
\[ s(x, y) = \vec{w} \cdot \vec{\phi}(x, y). \]
Multi Kernel Learning

- In MKL we minimize

\[ \lambda \| \mathbf{w} \|_{2,1}^2 + \frac{1}{N} \sum_{i=1}^{N} \ell(\mathbf{w}, x_i, y_i). \]

where \( \| \mathbf{w} \|_{2,1}^2 = \| w^1 \|_2, \| w^2 \|_2, \ldots, \| w^F \|_2 \|_1 \)

- The regularization induces sparsity in the domain of the kernels.

- All the proposed algorithms use an alternating optimization strategy, through the dual formulation.
Multi Kernel Learning

- In MKL we minimize
  \[
  \lambda \| \bar{\mathbf{w}} \|_{2,1}^2 + \frac{1}{N} \sum_{i=1}^{N} \ell(\bar{\mathbf{w}}, \mathbf{x}_i, y_i).
  \]

where \( \| \bar{\mathbf{w}} \|_{2,1}^2 = \| \| \mathbf{w}^1 \|_2, \| \mathbf{w}^2 \|_2, \cdots, \| \mathbf{w}^F \|_2 \|_1 \)

- The regularization induces sparsity in the domain of the kernels.

- All the proposed algorithms use an alternating optimization strategy, through the dual formulation.
But, are we sure this is the right regularization?

Is sparsity really needed?

Do we really want to use just a subset of the available kernels, given that each one is the result of years of research??
Why using the dual?

- Historically, dual formulation for SVM has been introduced to have an *easier* optimization problem and to use *kernels*.
Why using the dual?

- Historically, dual formulation for SVM has been introduced to have an *easier* optimization problem and to use *kernels*.
- However dual is not needed for neither of the two!
- Stochastic Sub-Gradient Descent algorithms for the primal have be proven to be better than optimizing the dual [Shalev-Shwartz and Srebro ICML08]!
Use your favorite loss!

- With stochastic sub-gradient descent methods you can use easily *any* loss.
- Computational efficient for large dataset.
- If the objective function is strongly convex functions we can prove fast convergence rate bound to the optimal solution.
  - The algorithm will converge to the optimal solution with a rate $O\left(\frac{1}{T}\right)$.
  - For alternating optimization methods this is not possible.
Use your favorite loss!

- With stochastic sub-gradient descent methods you can use easily *any* loss.
- Computational efficient for large dataset.
- If the objective function is strongly convex functions we can prove fast convergence rate bound to the optimal solution.
  - The algorithm will converge to the optimal solution with a rate $\mathcal{O}(\frac{1}{T})$.
  - For alternating optimization methods this is not possible.

However the group norm $(2, 1)$ is not strongly convex...
Outline

1. Multi Kernel Learning
   - Notation
   - Previous work
   - Sparsity?
   - Dual?

2. OBSCURE
   - A different MKL formulation
   - The algorithm

3. Experimental Results
   - Caltech-101
We propose to generalize the MKL formulation using the \((2, p)\) group norm

\[
\frac{\lambda}{2} \| \vec{w} \|_{2,p}^2 + \frac{1}{N} \sum_{i=1}^{N} \ell(\vec{w}, x_i, y_i),
\]

where \( \| \vec{w} \|_{2,p} \) is defined as \( \| w_1 \|_2, \| w_2 \|_2, \ldots, \| w_F \|_2 \|_p \).

When \( p = 1 \) we recover the sparse MKL formulation, \( p = 2 \) corresponds to using the sum of the kernels.

A similar formulation has been proposed in [Kloft et al. NIPS09].

If \( p \in (1, 2] \) this new formulation is \((1 - 1/p)\)-strongly convex.
A small ball is better than a big one...

- We want to minimize a convex function.
- If someone tells us that the solution is living in a small ball the problem is easier.
  - We can use this information with proximal regularization methods.
A small ball is better than a big one...

- We want to minimize a convex function.
- If someone tells us that the solution is living in a small ball the problem is easier.
  - We can use this information with proximal regularization methods.
A small ball is better than a big one...

- We want to minimize a convex function.
- If someone tells us that the solution is living in a small ball the problem is easier.
  - We can use this information with proximal regularization methods.
- But how to estimate this ball?
A small ball is better than a big one...

- We want to minimize a convex function.
- If someone tells us that the solution is living in a small ball the problem is easier.
  - We can use this information with proximal regularization methods.
- But how to estimate this ball?
- **Solution:** use a fast online algorithm!
Online-Batch Strongly Convex Multi Kernel Learning

- Start a quick online $(2, p)$ MKL algorithm.
- Stop it at any time to obtain an estimate of the radius of the ball, $R$, where the optimal solution lives.
- Start a stochastic gradient descent algorithm for the $(2, p)$ MKL problem, using the previous solution as starting point and the information on the radius of the ball.
Convergence rate for OBSCURE

Theorem

Let $1 < p \leq 2$, and $q = \frac{p}{p-1}$, $R$ the value returned by the online stage. Then in expectation after $T$ iterations of the 2nd stage of the OBSCURE algorithm, the gap from the optimal solution is

$$\mathcal{O} \left( F^{1/q} \min \left( \frac{q}{\lambda T}, \frac{R \sqrt{q}}{\sqrt{T}} \right) \right)$$
Convergence rate for OBSCURE

**Theorem**

Let $1 < p \leq 2$, and $q = \frac{p}{p-1}$, $R$ the value returned by the online stage. Then in expectation after $T$ iterations of the 2nd stage of the OBSCURE algorithm, the gap from the optimal solution is

$$O\left( F^{1/q} \min \left( \frac{q}{\lambda T}, \frac{R\sqrt{q}}{\sqrt{T}} \right) \right)$$

Moreover, if the problem is linearly separable by a hyperplane $\bar{u}$, the first stage will stop after $4qF^{2/q}\|\bar{u}\|_{2,p}^2$ updates, $R$ will overestimate the radius of the ball at most by a factor of 4.
A draft of the general algorithm

\textbf{Input:} $q, \bar{\theta}_1, \bar{w}_1, R, \lambda$

\textbf{for} $t = 1, 2, \ldots, T$ \textbf{do}

Sample at random $(x_t, y_t)$

Theory tells us how to set $\eta_t$ and $\alpha_t$

$\bar{\theta}_{t+\frac{1}{2}} = \alpha_t \bar{\theta}_t + \eta_t \partial \ell(\bar{w}_t, x_t, y_t)$

$w^j_{t+1} = \frac{1}{q} \left( \frac{\|\theta^j_{t+1}\|_2}{\|\bar{\theta}_t\|_2, q} \right)^{q-2} \theta^j_{t+1}, \forall j = 1, \ldots, F$

\textbf{end for}
\( \alpha_t \) and \( \eta_t \) are the core of the algorithm

- The choice of \( \alpha_t \) and \( \eta_t \) are critical to guarantee fast convergence to the optimal solution.
- Our particular choice is given by the theory: the details are in the paper.
- We just want to try it? Fine! Grab the source code at: http://dogma.sourceforge.net
  - Discriminative Online (Good?) Matlab Algorithms
  - The library is explicitly designed to have easy to modify algorithms.
Outline

1. Multi Kernel Learning
   - Notation
   - Previous work
   - Sparsity?
   - Dual?

2. OBSCURE
   - A different MKL formulation
   - The algorithm

3. Experimental Results
   - Caltech-101
We compared OBSCURE to SILP [Sonnenburg et al. JMLR06], LP-$\beta$ [Gehler and Nowozin ICCV09] and to SVM using average of all the kernels.

We used the Caltech-101 with 39 kernels, as in [Gehler and Nowozin ICCV09].
Caltech-101 Experiments: Performance

- OBSCURE is better than average kernel.
- Performance on par of LP-$\beta$.  

![Graph showing performance comparison of different methods across varying numbers of training examples. The graph plots classification rate against the number of training examples. The methods compared include OBSCURE, LP-$\beta$, MKL (SILP), and an average kernel. The graph indicates that OBSCURE performs better than the average kernel and is on par with LP-$\beta$.](image-url)
Caltech-101 Experiments: Time

- With 15 samples, time similar to LP-\(\beta\) and SILP.
- With 30 samples, OBSCURE is 7-10 times faster than LP-\(\beta\) and SILP.
Different settings of $p$

When there are few good kernel, the sparse solution is worst.

The optimal one corresponds to $p = 1.1$. 
More kernels = faster convergence

- We reach a given classification rate faster if we use more kernels.
Summary

- We have introduced a new formulation for MKL problems and an algorithm to solve it.
- The online stage of OBSCURE quickly estimates the region where the solution lives.
- The second stage reaches the solution with a guaranteed convergence rate.

Future work

- Extending OBSCURE to work with hierarchical losses.
Thanks for your attention

**Code:** [http://dogma.sourceforge.net](http://dogma.sourceforge.net)
**My website:** [http://francesco.orabona.com](http://francesco.orabona.com)