Multiple Kernel Learning via Multi-Epochs SVRG

Mitchel Alioscha-Perez
Vrije Universiteit Brussel (VUB)
Electronics and Informatics Dept (ETRO)
Pleinlaan 2, 1050 Brussels, Belgium
maperezg@etrovub.be

Meshia Cédric Oveneke
Vrije Universiteit Brussel (VUB)
Electronics and Informatics Dept (ETRO)
Pleinlaan 2, 1050 Brussels, Belgium
mcovenek@etrovub.be

Jiang Dongmei
Northwestern Polytechnical University (NPU)
Shaanxi Key Lab on Speech and
Image Information Processing
710072 Xi'an, China
jiangdm@nwpu.edu.cn

Hichem Sahli
Vrije Universiteit Brussel (VUB)
Electronics and Informatics Dept (ETRO)
Brussels 1050, Belgium
Interuniv. MicroElectronics Centre (IMEC)
3001 Heverlee, Belgium
hsahli@etrovub.be

Abstract

This work proposes a multiple kernel learning (MKL) descent strategy based on multiple epochs of stochastic variance reduced gradients (i.e. multi-epochs SVRG). The proposed descent strategy takes place with a constant-size learning step, that is entangled to the evolution of the kernels combination coefficients, and hence corrected in between epochs. This descending regime leads to an improved MKL bound that exhibits a linear dependency on the number of samples n, and sub-linear in both the number of kernels F and tolerance error ε. In particular, for an ℓ_p-norm MKL, the proposed method is able to find an ε-accurate solution in a complexity $O(F^{1/q} \cdot n \log(\frac{1}{\varepsilon}))$, where q is the dual norm. This matches the optimal convergence rate reported for (non-accelerated) strongly-convex objectives and improves over other state-of-the-art MKL solutions.

1 Introduction

Multiple kernel learning algorithms are very well suited to address multi-cue multi-source problems, and have been certainly competitive [1, 2, 3, 4] in several problem domains. Despite this success, traditional (batch) MKL solutions such as Level set MKL [5] and SMO-MKL [6] become extremely slow in presence of large amounts of data due to their (generally quadratic) complexity. Meanwhile, other MKL approaches such as SILP [7] can handle large amounts of data, but lack of theoretical guarantees on its convergence rate. On the other hand, incremental/stochastic MKL solutions based on SGD [8, 9] have proven to be much more efficient when addressing large scale problems due to a better (linear) complexity. However, their associated convergence deteriorates when very precise solutions are required. There is therefore a growing need of MKL solutions that are able to cope with large amounts data, arbitrary small errors, and large number of kernels in a computational efficient manner.

In this work, we propose a MKL solution based stochastic variance reduced gradients [10, 11, 12] (SVRG). The proposed descent strategy performs multiple epochs of SVRG with a constant-size learning step, that is entangled to the kernel's combination coefficients evolution and hence corrected in between epochs. This strategy yielded to a sub-linear dependency on the number of kernels F, while the multi-epochs SVRG allowed to obtain a sub-linear dependency on the error ε and a linear one in the number of samples n, resulting in an overall complexity $O(F^{1/q} \cdot n \log(\frac{1}{\varepsilon}))$.

9th NIPS Workshop on Optimization for Machine Learning,
30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

2 Related Works

As a main difference with other problems, solving MKL via semi-stochasticity requires a careful consideration regarding assumptions on the strong convexity parameter due to the presence of multiple kernels in the regularization term. Authors in [8] considered (within the context of SGD) a formulation that is ensured [13] to be $\frac{\lambda}{q}$-strongly convex for their particular ℓ_p-norm MKL objective. In principle, this formulation could also be used in the semi-stochastic framework, easing the analysis since the strong convexity parameter becomes immediately clear. However, it also deteriorates the prospects of convergence by decreasing the objective curvature from λ to $\frac{\lambda}{q}$ (less strongly convex since $q > 1$ in an ℓ_p-MKL context). Instead, we followed the formulation in [14] and designed a descending strategy that yielded to an improved bound compared to other MKL solutions (see Table 1). Notably, the bigger q becomes the sparser the kernel combination vector will be, which will result in a faster convergence.

Type	MKL Method	Norm	Complexity for $\varepsilon < 1$
Non-Stoch	Level MKL [5]	$p = 1$	$O\left(F \cdot \frac{n^2}{\varepsilon^2}\right)$
Stochastic	UFO-MKL [15]	$p = \frac{2 \log F}{2 \log F - 1}$	$O\left(\log F \cdot \frac{n}{\varepsilon}\right)$
Stochastic	Obscure MKL [8]	$1 < p \leq 2$	$O\left(qF^{1/q} \cdot \frac{n}{\varepsilon}\right)$
Semi-Stoch	This work	$1 < p < \infty$	$O\left(F^{1/q} \cdot n \log(\frac{1}{\varepsilon})\right)$

Table 1: Complexity of different MKL methods taking into account the ℓ_p-norm with $\frac{1}{p} + \frac{1}{q} = 1$, total samples n, number of kernels F and tolerance error ε; we recall that $\lambda \propto \frac{1}{n}$ thus $\frac{1}{\lambda}$ is $O(n)$. The comparison includes traditional (non-stochastic) methods, stochastic ones and the proposed method.

Very few works have addressed MKL solutions based on variance reduction techniques. Among the exceptions we can find the general work of [16], that could be (indirectly) used to solve MKL formulations. Our work goes precisely in this line and aims at solving MKL by making use of SVRG [10] within the semi-stochastic framework of [11].

We provide the proposed optimization Algorithm in Section 3.1 along with a sketch of the proof of its convergence in Section 3.2.

3 Proposed MKL via Multi-Epochs SVRG

3.1 Problem Formulation and Solution

We assume that the MKL objective function is L-smooth, and each individual single-kernel problem is λ-strongly convex with respect to the Euclidean norm. Consider thus an ℓ_p-norm ($1 < p < \infty$) combination of F kernels in the following constrained optimization problem:

$$\min_{\{d^k \geq 0\}, w} \quad g(w) = \frac{1}{2} \sum_{k=1}^{F} \frac{\lambda}{d^k} \|w^k\|_2^2 + \frac{1}{n} \sum_{t=1}^{n} \ell\left(w, x_t, y_t\right) \quad \text{s.t.} \sum_k (d^k)^p \leq 1 \tag{1}$$

with $w = (w^1, \cdots, w^F)$, given n tuples $(x, y) \in \mathcal{X} \times \mathcal{Y}$ and a regularization parameter $\lambda > 0$, who's value is commonly [14, 8, 15] set to $\lambda = \frac{1}{Cn}$ (C is the well-known SVM hyper-parameter). We also adopted the framework of Kloft. *et al.* [14], where $\frac{x}{0} = 0$ if $x = 0$, otherwise $\frac{x}{0} = \infty$ [14, 17]; hence $w^k = 0$ whenever $d^k = 0$ to reach a finite objective. Among other benefits, this formulation allows involving an ℓ_p-norm such that $1 < p < \infty$ (consequently $1 < q < \infty$) and is slightly different from [8, 15]. The solution of the combination coefficients $d^1, \ldots, d^F$ uniquely depends on the values $w^1, \ldots, w^F$ respectively, and can be obtained through a Lagrange derivation in a closed form as follows:

$$d^k \quad = \quad \left(\|w^k\|_2^2\right)^{\frac{1}{p+1}} \cdot \left[\sum_{s=1}^{F} \left(\|w^s\|_2^2\right)^{\frac{p}{p+1}}\right]^{-\frac{1}{p}} \tag{2}$$

which is in accordance with [14] and [18]. In the proposed solution, we solve (1) interleavingly by epochs in a semi-stochastic approach. At the beginning of each i-th epoch we consider a fixed set of kernel coefficients $d_i^k \, \forall k$ and descend at each k- th particular block w_i^k at a time. Under this view, we have k individual single kernel problems that share the same loss ℓ, each one with a potentially different strong convexity degree $\tilde{\lambda}^k = \frac{\lambda}{d^k}$ according to their associated coefficient d^k. The semi-stochastic descending regime [11] is fully determined by the achieved convergence c from the settings of basically three parameters, that we will refer to as descent parameters: the target decrease in the objective $\triangle$ (defined indirectly by ε), and a proper combination of the learning step h and number of iterations m. The setting of both h, m depends on the strong convexity parameter λ and Lipschitz constant L. The Lipschitz constant L bounds the gradient $\nabla g = \sum \frac{1}{n} \nabla g_t$ of (1), where[1] g_t considers only one tuple $(x_t, y_t) \in \mathcal{X} \times \mathcal{Y}$ from the training set [11].

As stated above, we consider k problems each one with a (potentially) different strong convexity parameter $\tilde{\lambda}^k$. As consequence, each single-kernel problem will have associated its own set of descent parameters, defined in Proposition 1 in Section (3.2), and hence each kernel follow its own tailored descending regime. The number of iterations at the i-th epoch for the k kernels is therefore $m(\triangle, \frac{L}{\tilde{\lambda}^k})$, with $\tilde{\lambda}^k = \frac{\lambda}{d_i^k}$, or equivalently:

$$m\left(\triangle, \frac{L}{\tilde{\lambda}^k}\right) = m^k\left(\triangle, d_i^k \frac{L}{\lambda}\right) \overset{(5)}{\leq} d_i^k m\left(\triangle, \frac{L}{\lambda}\right) = d_i^k m\left(\triangle, \kappa\right) \tag{3}$$

where $\kappa = \frac{L}{\lambda}$ is a fixed/unique condition number associated to (1). A key observation in the above equivalence is that the number of iterations at each k-th kernel, for a fixed d_i^k at some i- th epoch, is simply a partitioning (according to d_i) of certain amount of iterations $m(\triangle, \kappa)$ between all the kernels. This observation, in combination with the constraint $\sum (d^k)^p \leq 1$, is at the core of our optimization strategy (detailed in Algorithm 1) and yields the improvements related to the dependency on F. Theorem 1 in Section (3.2) shows that due to this partitioning, the total amount of iterations performed by all the kernels at each epoch is at most $F^{1/q} m(\kappa)$.

Algorithm 1 Semi-Stochastic MKL

Require: Parameters: $p, L, \lambda, \varepsilon$

Initialize $\tilde{c}_0^k = 1$, $d^k = \frac{1}{F}$ and $w_0^k \leftarrow$ `Randomly` $\quad \forall k$
for $j = 1, \cdots, [\log(1/\varepsilon)]$ **do**
  Set parameters $\triangle_j^k(\tilde{c}_j^k), h_j^k(\triangle_j^k, L_k, \tilde{\lambda}_k), m_j^k(\triangle_j^k, \frac{L_k}{\tilde{\lambda}_k}) \quad \forall k$
  Let $T_j^k \leftarrow t$ with probability $(1 - \tilde{\lambda}_k h_j^k)^{m_j^k - t} \quad$ for $\quad t = 1, .., m_j^k \quad \forall k$
  Initialize $\omega_0 \leftarrow (w_{j-1}^1, \cdots, w_{j-1}^F)$
  Prepare full gradient snapshot $\mathcal{G}_j = \sum_{t=1}^n \frac{1}{n} \nabla g_t(\omega_0; d)$
  for $t = 0, \cdots, \max(m_j^1, \ldots, m_j^F)$ **do**
    $(x_t, y_t) \leftarrow$ `Random sample (uniformly selected from training set)`
    for $k = 1, \cdots, F$ **do**
      if $t < T_j^k$ **then**
        Update solution $\omega_{t+1}^k = \omega_t^k - h_j^k(\mathcal{G}_j^k + \frac{\partial g_t}{\partial \omega_t^k}(\omega_t; d) - \frac{\partial g_t}{\partial \omega_t^k}(\omega_0; d))$
      end if
    end for
  end for
  $w_j^k \leftarrow \omega_{T_j^k}^k \quad \forall k$
  Update d^k and set $\tilde{\lambda}^k \leftarrow \min(\frac{\lambda}{d^k}, \frac{L}{2}) \quad \forall k$
  Re-assess work done $\tilde{c}_i^k \leftarrow c\left(\tilde{\lambda}^k, h_i^k, m_i^k\right) \, \forall_{i=1}^j \, \forall k$
end for
return d, w_J

[1]For sample (x_t, y_t), ∇g_t denotes gradient with respect to ω, and $\frac{\partial g_t}{\partial \omega^k}$ partial derivative with respect to ω^k

Since the descent parameters are set according to $\tilde{\lambda}$ which changes in between epochs, then the learning step h and the number of iterations m have to be adjusted in order to guarantee convergence to the desired precision ε. This is done by reassessing the progress achieved up to the current epoch via $\tilde{c}$. High values of $\tilde{c}$ (lower convergence rate) achieved at early epochs will require a compensation in further epochs by descending at higher convergence rate, which in turn will require more iterations (smaller h and bigger m).

3.2 Convergence Analysis

Proposition 1: Assume an objective that is L-smooth and λ-strongly convex. Consider $\varepsilon < 1$ and a total of $J = \log(1/\varepsilon)$ epochs. Given $\epsilon < \tilde{c}_1\tilde{c}_2\cdots\tilde{c}_{j-1}$, at each j-th epoch, define:

$$\triangle_j(\tilde{c}_1,\ldots,\tilde{c}_{j-1}) = \left(\frac{\min(\epsilon,\varepsilon)}{\prod_{z=0}^{j-1}\tilde{c}_z}\right)^{\frac{1}{J-j}} < 1 \tag{4}$$

and fix the learning step $0 < h < \frac{1}{2L}$, and the number of iterations m as (with $\kappa = \frac{L}{\lambda}$):

$$h(\triangle_j, L, \lambda) = \frac{1}{\frac{4}{\triangle_j}(L-\lambda)+2L} \quad , \quad m(\triangle_j,\kappa) \geq \left(\frac{4(\kappa-1)}{\triangle_j}+2k\right)\log\left(\frac{2}{\triangle_j}+\frac{2\kappa-1}{\kappa-1}\right) \tag{5}$$

resulting in a convergence at j-th epoch as:

$$c(\lambda,h,m) = \frac{(1-\lambda h)^m}{(1-(1-\lambda h)^m)(1-2Lh)} + \frac{2(L-\lambda)h}{1-2Lh} \leq \triangle_j \tag{6}$$

Then, running J epochs of Algorithm 1 allows converging to an ε-accurate solution at a rate $\tilde{c}_1\tilde{c}_2\cdots\tilde{c}_J \leq \varepsilon$. In particular, since $J = \log(\frac{1}{\varepsilon})$, then $\frac{1}{\triangle} \leq \exp(1)$ and hence $m(\kappa) = O(\kappa)$.

Proof. By choosing h and m as in (5), (Theorem 6 in [11]) establishes that $c \leq \triangle$. Then, after J epochs, by definition (4) we have that $\tilde{c}_1\tilde{c}_2\cdots\tilde{c}_J \leq \varepsilon$, with $\tilde{c}_j = \varepsilon^{1/J}\,\forall j$ as a particular case. Finally, for the given $\prod\tilde{c}_j \leq \varepsilon$, (Theorem 4 in [11]) guarantees convergence to an ε-accurate solution in J epochs. □

Theorem 1: Consider a MKL problem in a setup of an ℓ_p-norm combination of F kernels, with $1 < p < \infty$. Fix the number of epochs $J = \log(\frac{1}{\varepsilon})$ for some $\varepsilon < 1$. Set descent parameters ($\triangle$, h and m) according to Proposition 1. Then, starting from a solution w_0 and running J epochs of Algorithm 1 allows finding an ε-accurate solution w_J such that in the expectation:

$$E\left(g(w_J) - g(w_*)\right) \leq \varepsilon E\left(g(w_0) - g(w_*)\right) \tag{7}$$

in a complexity $\mathcal{W}^*(J,h,m) \leq O\left(\left(n + F^{1/q}\kappa\right)\log\left(\frac{1}{\varepsilon}\right)\right)$.

Proof. By choosing descent parameters according to Proposition 1, then convergence in view of (7) is guaranteed with $\tilde{c}_1\tilde{c}_2\cdots\tilde{c}_J \leq \varepsilon$. For the second part of the claim, denote $j = \arg\max_z \varphi(z)$ as the epoch where the total number of iterations is maximum, thus:

$$\varphi(j) \overset{(3)}{=} \sum_{k=1}^{F} o_j^k m\left(\kappa\right) \leq \max_{\{o_j^k\}\geq 0} \sum_{k=1}^{F} o_j^k m\left(\kappa\right) \leq \left(\sum_{k=1}^{F} m(\kappa)^q\right)^{1/q} \leq F^{1/q} m(k) \qquad \forall j \tag{8}$$

for $\mathbf{o}_j = (o_j^1,\cdots,o_j^F)$, $\|\mathbf{o}_j\|_p^p \leq 1$; the solution (8) can also be obtained in closed form via Lagrange derivation. The total cost of performing J-epochs of Algorithm 1 is at most $O(nJ + \varphi(j)J)$ (see lines $2, 7, 9, 10$), accounting for both full and stochastic gradients. Then, since $m(\kappa) = O(\kappa)$ (Proposition 1), from (8) we have $\varphi(j) \leq F^{1/q}m(\kappa) \leq O(F^{1/q}\kappa)$ which completes the proof. □

Acknowledgments

The Belgian Federal Science Policy Office (Belspo), the European Space Agency (ESA) PRODEX program (project ILSRA-2009-1156 FluoCells) and the Research Council of the Vrije Universiteit Brussel supported this work.

References

[1] J. Yang, Y. Li, Y. Tian, L. Duan, and W. Gao, "Group-sensitive multiple kernel learning for object categorization," in *IEEE 12th International Conference on Computer Vision*. Ieee, sep 2009, pp. 436–443.

[2] L. Duan, I. W. Tsang, and D. Xu, "Domain transfer multiple kernel learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 465–479, 2012.

[3] B. Ni, T. Li, and P. Moulin, "Beta Process Multiple Kernel Learning," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 963–970, 2014.

[4] S. Bucak, R. Jin, and A. Jain, "Multiple Kernel Learning for Visual Object Recognition: A Review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1354–1369, 2014.

[5] Z. Xu, R. Jin, I. King, and M. Lyu, "An extended level method for efficient multiple kernel learning," *Advances in Neural Information Processing Systems*, pp. 1825–1832, 2009.

[6] Z. Sun, N. Ampornpunt, and S. Vishwanathan, "Multiple kernel learning and the SMO algorithm," *Advances in Neural Information Processing Systems*, pp. 2361–2369, 2010.

[7] S. Sonnenburg, R. Gunnar, C. Schafer, and S. Bernhard, "Large scale multiple kernel learning," *Journal of Machine Learning Research*, vol. 7, pp. 1531–1565, 2006.

[8] F. Orabona, L. Jie, and B. Caputo, "Online-batch strongly convex multi kernel learning," in *Computer Vision and Pattern Recognition*, 2010, pp. 787–794.

[9] H. Xia, S. C. H. Hoi, R. Jin, and P. Zhao, "Online Multiple Kernel Similarity Learning for Visual Search," *Pattern Analysis and Machine Intelligence*, vol. 36, no. 3, pp. 536–549, 2014.

[10] R. Johnson and T. Zhang, "Accelerating Stochastic Gradient Descent using Predictive Variance Reduction," *Advances in Neural Information Processing Systems*, pp. 315–323, 2013.

[11] J. Konečný and P. Richtárik, "Semi-Stochastic Gradient Descent Methods," vol. 1, p. 19, 2013. [Online]. Available: http://arxiv.org/abs/1312.1666

[12] J. Mairal, "Incremental Majorization-Minimization Optimization with Application to Large-Scale Machine Learning," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 829–855, 2015.

[13] S. M. Kakade, S. Shalev-Shwartz, and A. Tewari, "On the duality of strong convexity and strong smoothness: Learning applications and matrix regularization," *Unpublished Manuscript*, 2009.

[14] M. Kloft, U. Brefeld, P. Laskov, K.-R. Müller, A. Zien, and S. Sonnenburg, "Efficient and accurate lp-norm multiple kernel learning," *Advances in Neural Information Processing Systems*, pp. 997–1005, 2009.

[15] F. Orabona and L. Jie, "Ultra-Fast Optimization Algorithm for Sparse Multi Kernel Learning," *Proceedings of the 28th International Conference on Machine Learning*, pp. 249–256, 2011.

[16] S. Shalev-shwartz, "Stochastic Dual Coordinate Ascent Methods for Regularized Loss," *Journal of Machine Learning Research*, vol. 14, pp. 567–599, 2013.

[17] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet, "Simplemkl," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2491–2521, 2008.

[18] Z. Xu, R. Jin, H. Yang, I. King, and M. R. Lyu, "Simple and efficient multiple kernel learning by group lasso," *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 1175—-1182, 2010.