Partial Linearization based Optimization for Multi-Class SVM

Pritish Mohapatra[1], Puneet K. Dokania[2], C.V. Jawahar[1], M. Pawan Kumar[3]

[1]International Institute of Information Technology, Hyderabad
[2]Centrale Supelec & INRIA Saclay, [3]University of Oxford

1. Short Summary

Contribution

A novel partial linearization [1] based approach for optimizing the multi-class SVM learning problem. Our method is an intuitive **generalization of the Frank-Wolfe** [2] **and the Exponentiated-Gradient** [3] **algorithms**.

Advantages

- **Informative descent direction**

  computed using the marginals over each output class, similar to Exponentiated-Gradient.
- **Optimal step-size in the descent direction**

  that guarantees an increase in the dual objective, similar to Frank-Wolfe.
- **Block coordinate formulation**

  similar to the one proposed for Frank-Wolfe, which allows us to solve large-scale problems.

2. Multi-class SVM

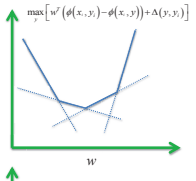
- Input: $\mathbf{x} \in X$; Output: $y \in Y$, where $Y = \{1,\dots, c\}$ & c is the no. of classes
- Feature representation for sample: $\varphi(\boldsymbol{x})$
- Joint Feature map for sample and candidate class:

$$\phi(x,y) = \left[v_1^T \cdots v_j^T \cdots v_c^T \right] \qquad where, \quad v_j = \begin{cases} \varphi(x) & if \quad j = y. \\ 0 & otherwise. \end{cases}$$

- **Prediction:** $y_{opt} = \arg\max_{y} \; w^T \phi(x,y)$
- **Learning:**

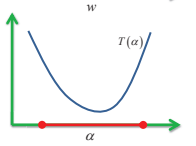
$$\min_{w,\xi} \frac{\lambda}{2}\|w\|^2 + \frac{1}{n}\sum_{i=1}^{n}\xi_i$$
$$s.t. \quad \forall y, i: \quad w^T\phi(x_i,y_i) - w^T\phi(x_i,y) \geq \Delta(y,y_i) - \xi_i$$

Primal



$$\min_{\alpha} T(\alpha) = -b^T\alpha + \frac{\lambda}{2}\alpha^T A^T A\alpha$$
$$s.t. \quad \sum_{y\in Y}\alpha_{iy} = 1; \forall i \in [n]$$

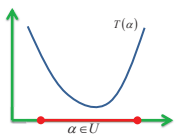
Dual



3. Partial Linearization framework

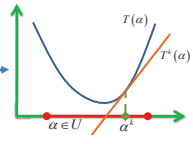
- Consider Optimization problem: $\min_{\alpha\in U} T(\alpha)$

  Where, $T(\alpha)$ is a convex and continuously differentiable function and U is a compact & convex set.



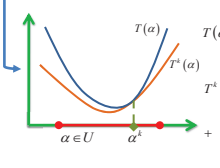
Optimization using Partial Linearization

- k=0, Initialize α^k.
- Repeat until convergence:
  - Construct **local approximation** $T^k(\alpha)$
  - Compute: $s^k = \arg\min_{\alpha\in U} T^k(\alpha)$
  - Compute optimal step size γ.
  - Update α, $\alpha^{k+1} \leftarrow (1-\gamma)\alpha^k + (\gamma)s^k$
  - k ← k+1
- Return optimal α.



Local Linear approximation (as in Frank-Wolfe)

$$T^k(\alpha) = T(\alpha^k) + \left[\nabla T(\alpha^k)\right]^T(\alpha - \alpha^k)$$



Local Partial-linear approximation

$$T(\alpha) = f(\alpha,\alpha^k) + \left(T(\alpha) - f(\alpha,\alpha^k)\right)$$

Surrogate — Error term

$$T^k(\alpha) = f(\alpha,\alpha^k) + \begin{pmatrix} T(\alpha^k) - f(\alpha^k,\alpha^k) \\ +\left[\left(\nabla T(\alpha^k) - \nabla_\alpha f(\alpha^k,\alpha^k)\right)\right]^T(\alpha - \alpha^k) \end{pmatrix}$$

Linearization of only the Error term

4. Partial Linearization based Optimization for Multi-class SVM

Appropriate design of Surrogate function:

$$f(\alpha,\alpha^k) = \frac{\tau}{n}\sum_{i\in[n]}\sum_{y\in Y}\alpha_{iy}\log(\alpha_{iy})$$

- Resulting problem is efficiently optimizable
- Captures information about violation of each primal constraint

Temperature: Non-negative hyper-parameter

Corresponding update direction:

$$s^k_{iy} = \frac{\exp\left(\log(\alpha^{k-1}_{iy}) + \frac{1}{\tau}\left(\Delta_i(y) - w^{k-1^T}(\phi(x_i,y_i) - \phi(x_i,y))\right)\right)}{z_i}$$

Partial Linearization for Optimizing Multi-Class SVM

- k=0, Initialize α^k.
- Repeat until convergence:
  - for all $i \in [n]$ do:

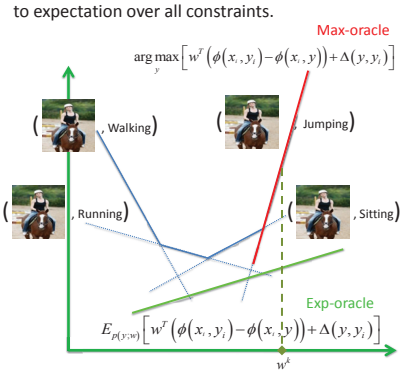
$$\forall y \in Y \quad s^k_{iy} = \frac{\exp\left(\log(\alpha^{k-1}_{iy}) + \frac{1}{\tau}\left(\Delta_i(y) - w^{k-1^T}(\phi(x_i,y_i) - \phi(x_i,y))\right)\right)}{z_i}$$

  - Compute optimal step size,

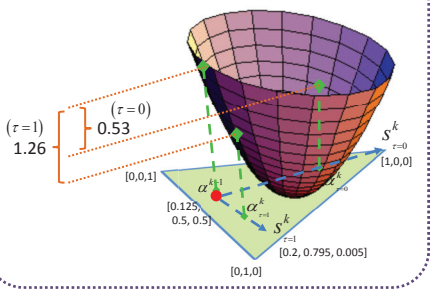
$$\gamma \leftarrow \frac{\langle \alpha^{k-1} - s^k, -b + A^T A\alpha^{k-1}\rangle}{\lambda\|A(\alpha^{k-1} - s^k)\|^2}$$

  - Update α, $\alpha^{k+1} \leftarrow (1-\gamma)\alpha^k + (\gamma)s^k$
  - Update w, $w^k \leftarrow A\alpha^k$
  - k ← k+1
- Return optimal **w**.

In primal space, new update direction corresponds to expectation over all constraints.



Illustrative example: Update steps for τ=0 and τ>0



- For Temperature: **τ=0**,
  Partial-Linearization ↕ **Frank-Wolfe**
  For **τ>0**, can have potentially better update direction
- For Step-size: **γ=1**,
  Partial-Linearization ↕ **Exponentiated-Gradient**
  For **γ<1**, can have potentially better update step

Block Coordinate Partial Linearization

- Useful/Necessary for very large datasets.
- A coordinate descent like approach.
- Updates model after every single sample and is more efficient than the batch version.

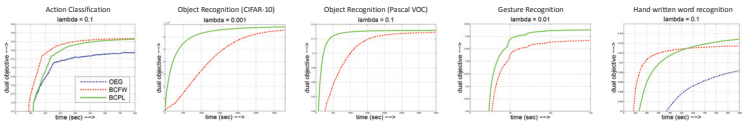
Partial Linearization based optimization for Structured-SVM

- Can be used for tree-structured output spaces.
- Message passing for computing exact marginals for the update step.

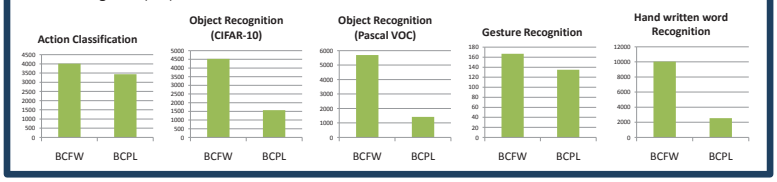
5. Experiments and Results

- **Problems and Datasets:** Training Multi-Class SVM or Structured SVM models for Vision tasks like Classification (Pascal VOC), Object Recognition (CIFAR-10, Pascal VOC), Gesture Recognition (MSRC-12) and Hand written word recognition (OCR).
- **Methods:** We compare our Block-Coordinate Partial Linearization algorithm (BCPL) with Block-coordinate Frank-Wolfe (BCFW) algorithm [2] and Online exponentiated gradient (OEG) algorithm. We use a fixed temperature of τ=0.01 for our BCPL algorithm in all the experiments.
- **Results:**

Example training curves (Dual objective Vs. time (sec)) :



Mean training time (sec) of Multi-Class SVM or Structured-SVM models for different tasks:



6. References

[1] M. Patriksson, ``Partial linearization methods in nonlinear programming. Journal of Optimization Theory and Applications", Springer, 1993.

[2] S. Lacoste-Julien, M. Jaggi, M. Schmidt, P. Pletscher, ``Block-coordinate frank-wolfe for structural SVMs", ICML , 2012.

[3] M. Collins, A. Globerson, T. Koo, X. Carreras, P. L. Bartlett, ``Exponentiated gradient algorithms for conditional random fields and max-margin markov networks", The Journal of Machine Learning Research, 2008.