Learning with Privileged Information: New Optimization Algorithms and Applications

Dmitry Pechyony, Léon Bottou and Vladimir Vapnik

NEC Laboratories America, Princeton, USA
pechyony@nec-labs.com, leon@bottou.org and vlad@nec-labs.com

Recently a new learning paradigm, called *Learning Using Privileged Information (LUPI)*, was introduced by Vapnik et al. [5–7]. In this paradigm, in addition to the standard training data, $(\mathbf{x}, y) \in X \times \{\pm 1\}$, a teacher supplies a student with the *privileged information* $\mathbf{x}^* \in X^*$. The privileged information is only available for the training examples and is never available for the test examples. The LUPI paradigm requires, given a training set $\{(\mathbf{x}_i, \mathbf{x}_i^*, y_i)\}_{i=1}^n$, to find a function $f : X \rightarrow \{-1, 1\}$ with the small generalization error for the unknown test data $\mathbf{x} \in X$.

LUPI paradigm can be implemented based on SVM algorithm [2]. The decision function of SVM is $f(\mathbf{z}) = \mathbf{w} \cdot \mathbf{z} + b$, where $\mathbf{z}$ is a feature map of $\mathbf{x}$, and $\mathbf{w}$ and b are the solution of (1).

$$\min_{\mathbf{w},b,\xi_1,\ldots,\xi_n} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^n \xi_i \tag{1}$$

$$\text{s.t. } \forall\ 1 \leq i \leq n,\ y_i\left(\mathbf{w} \cdot \mathbf{z}_i + b\right) \geq 1 - \xi_i,$$
$$\forall\ 1 \leq i \leq n,\ \xi_i \geq 0.$$

Let $f^* = (\mathbf{w}^*, b^*)$ be the decision function found by SVM when $n = \infty$. Suppose that for each training example $\mathbf{x}_i$ an oracle gives us the value of the slack $\xi_i^* = 1 - y_i(\mathbf{w}^* \cdot \mathbf{x}_i + b^*)$. We substitute these slacks into (1), fix them and optimize (1) only over $\mathbf{w}$ and b. We denote such variant of SVM as *OracleSVM*. The generalization error of the decision function found by OracleSVM converges [6] to the one of f^* with the rate of $1/n$. This rate is much faster than the convergence rate $1/\sqrt{n}$ of SVM.

In the absence of the optimal values of slacks we use the privileged information $\{\mathbf{x}_i^*\}_{i=1}^n$ to estimate them. Let $\mathbf{z}_i^*$ be a feature map of $\mathbf{x}_i^*$. We seek a *correcting function* $\phi(\mathbf{x}_i^*) = \mathbf{w}^* \cdot \mathbf{z}_i^* + d$ that approximates ξ_i^*. We substitute $\xi_i = \phi(\mathbf{x}_i^*)$ into (1) and obtain the modification (2) of SVM, called SVM+ [5]. The objective function of SVM+ contains two hyperparameters, $C > 0$ and $\gamma > 0$. The term $\gamma\|\mathbf{w}^*\|/2$ in (2) is intended to restrict the capacity (or VC-dimension) of the function space containing ϕ.

$$\min_{\mathbf{w},b,\mathbf{w}^*,d} \frac{1}{2}\|\mathbf{w}\|_2^2 + \frac{\gamma}{2}\|\mathbf{w}^*\|_2^2 + C\sum_{i=1}^n(\mathbf{w}^* \cdot \mathbf{z}_i^* + d) \tag{2}$$

$$\text{s.t. } \forall\ 1 \leq i \leq n,\ \ y_i\left(\mathbf{w} \cdot \mathbf{z}_i + b\right) \geq 1 - (\mathbf{w}^* \cdot \mathbf{z}_i^* + d),$$
$$\forall\ 1 \leq i \leq n,\ \ \mathbf{w}^* \cdot \mathbf{z}_i^* + d \geq 0.$$

A common approach to solve (1) is to consider its dual problem. We also use this approach to solve (2). The dual optimization problems of SVM and SVM+ are (3) and (5) respectfully,

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i,j=1}^n \alpha_i\alpha_j y_i y_j K_{ij} \tag{3}$$

$$\text{s.t. } \sum_{i=1}^n y_i\alpha_i = 0, \tag{4}$$
$$\forall\ 1 \leq i \leq n,\ \ 0 \leq \alpha_i \leq C.$$

$$\max_{\boldsymbol{\alpha},\boldsymbol{\beta}} \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i,j=1}^n \alpha_i\alpha_j y_i y_j K_{ij} - \frac{1}{2\gamma}\sum_{i,j=1}^n \tau_i\tau_j K_{ij}^* \tag{5}$$

$$\text{s.t. } \sum_{i=1}^n \tau_i = 0, \quad \sum_{i=1}^n y_i\alpha_i = 0,$$
$$\forall\ 1 \leq i \leq n,\ \ \tau_i = \alpha_i + \beta_i - C,\ \ \alpha_i \geq 0,\ \ \beta_i \geq 0.$$

where $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$ and $K_{ij}^* = K^*(\mathbf{x}_i^*, \mathbf{x}_j^*)$ are kernels in the decision and the correcting space respectfully. The decision and the correcting functions, expressed in the terms of the dual variables, are $f(\mathbf{x}) = \sum_{j=1}^{n} \alpha_j y_j K(\mathbf{x}_j, \mathbf{x}) + b$ and $\phi(\mathbf{x}_i^*) = \frac{1}{\gamma} \sum_{j=1}^{n} (\alpha_j + \beta_j - C) K_{ij}^* + d$ respectfully. SVM and SVM+ have syntactically the same decision function. However semantically they are different, since the values of $\boldsymbol{\alpha}$'s found by SVM and SVM+ can differ significantly.

One of the widely used algorithms for solving (3) is SMO [4]. At each iteration SMO optimizes the *working set* of two variables, α_s and α_t, while keeping all other variables fixed. We cannot optimize the proper subset of such working set, say α_s: due to (4), if we fix $n-1$ variables then the last variable is also fixed. Hence the working sets selected by SMO are *irreducible*.

Following [1], we present SMO as an instantiation of the framework of sparse line search algorithms to the optimization problem of SVM. At each iteration the algorithms in this framework perform line search in some chosen sparse direction which is close to the gradient direction. We instantiate the above framework to SVM+ optimization problem and obtain alternating SMO (aSMO) algorithm. aSMO works with irreducible working sets of two or three variables.

Sparse line search algorithms in turn belong to the family of approximate gradient descent algorithms. The latter algorithms optimize by going roughly in the direction of gradient. Unfortunately gradient descent algorithms can convergence very slowly. One of the possible remedies is to use conjugate direction optimization, where each new search direction is conjugate to all previous ones. When applied to SVM-like problems, the conjugate direction optimization is very expensive, since it requires to store in memory the entire kernel matrix.

We combine the ideas of sparse line search and conjugate direction and present a framework of optimization algorithms, that at each iteration perform line search in a chosen sparse direction which is close to the gradient direction and is conjugate to k previously chosen ones. We instantiate this framework to SVM and SVM+ problems and obtain Conjugate SMO (cSMO) and Conjugate Alternating SMO (caSMO) algorithms. cSMO and caSMO work with irreducible working sets of size up to $k+2$ and $k+3$ respectfully. We show empirically that for large values of hyperparameter C, cSMO is significantly faster than SMO. Also our experiments indicate an order-of-magnitude running time improvement of caSMO over cSMO.

Vapnik et al. [6, 7] showed that LUPI paradigm emerges in several domains, for example, time series prediction and protein classification. To motivate further the usage of LUPI paradigm and SVM+, we show how it can be used to learn from the data generated by human computation games. Our experiments indicate that the tags generated by the players in ESP [8] and Tag a Tune [3] games result in accuracy improvement in image and music classification.

References

1. L. Bottou and C.-J. Lin. Support Vector Machine solvers. In *Large Scale Kernel Machines*, pages 1–27. 2007.
2. C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
3. E. Law and L. von Ahn. Input-agreement: A new mechanism for data collection using human computation games. In *CHI*, pages 1197–1206, 2009.
4. J. Platt. Fast training of Support Vector Machines using Sequential Minimal Optimization. In *Advances in Kernel Methods - Support Vector Learning*, pages 185–208. MIT Press, 1999.
5. V. Vapnik. *Estimation of dependencies based on empirical data*. Springer–Verlag, 2nd edition, 2006.
6. V. Vapnik and A. Vashist. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22(5-6):544–557, 2009.
7. V. Vapnik, A. Vashist, and N. Pavlovich. Learning using hidden information: Master class learning. In *Proceedings of NATO workshop on Mining Massive Data Sets for Security*, pages 3–14. 2008.
8. L. von Ahn and L. Dabbish. Labeling images with a computer game. In *CHI*, pages 319–326, 2004.

Topic: learning algorithms
Preference: oral/poster