Online Limited-Memory Quasi-Newton Training of Support Vector Machines

Jin Yu **Nicol N. Schraudolph** **S.V. N. Vishwanathan**

Statistical Machine Learning, National ICT Australia
Locked Bag 8001, Canberra ACT 2601, Australia

Research School of Information Sciences & Engineering
Australian National University, Canberra ACT 0200, Australia

January 19, 2007

The limited-memory Broyden-Flecher-Goldfarb-Shanno (LBFGS, Nocedal and Wright, 1999) algorithm is the standard technique for large-scale nonlinear optimization; it is scalable to very high-dimensional problems as it only requires linear time and memory. Recently we have developed oLBFGS, a stochastic variant of LBFGS for online optimization of convex functions (Schraudolph et al., 2007). oLBFGS further increases convergence speed by learning on small subsamples of data, and can outperform first-order stochastic gradient methods as it inherits the curvature-invariant property of standard LBFGS. Here we report on our work to extend oLBFGS to the online training of Support Vector Machines (SVMs).

Let $\mathcal{X}$ be the space of observations, and $\mathcal{Z}$ the space of labels. Given a set of labeled instances $\{\boldsymbol{x}_i, z_i\}_{i=1}^n \subset \mathcal{X} \times \mathcal{Z}$ where $\mathcal{X} \subseteq \mathbb{R}^n$, a non-linear SVM solves the following optimization problem:

$$\min_{f \in \mathcal{H}} J(f) := \frac{c}{2}\|f\|_{\mathcal{H}}^2 + \sum_{i=1}^{n} l(\boldsymbol{x}_i, z_i, f), \tag{1}$$

where l is a piecewise differentiable loss function $l : \mathcal{X} \times \mathcal{Z} \times \mathcal{H} \to \mathbb{R}$, $\mathcal{H}$ is a Reproducing Kernel Hilbert Space (RKHS) of functions on $\mathcal{X}$ and $\|\cdot\|_{\mathcal{H}}$ is the RKHS norm. The defining kernel [1] of $\mathcal{H}$ is k.

Since the gradient $\partial_f J(f)$ is a member of the RKHS, we reformulate the regularized risk (1) in terms of kernel function coefficients $\boldsymbol{\sigma}$:

$$J(\boldsymbol{\sigma}) := \frac{c}{2}\boldsymbol{\sigma}^\top \boldsymbol{K}\boldsymbol{\sigma} + \sum_{i=1}^{n} l(z_i, \boldsymbol{K}_i^\top \boldsymbol{\sigma}), \tag{2}$$

[1] For ease of exposition, we assume kernel function k does not depend on the value of z.

where $\boldsymbol{K}$ is the kernel matrix and $[\boldsymbol{K}]_i$ is the ith column of $\boldsymbol{K}$. Then, the gradient $\partial_{\boldsymbol{\sigma}} J(\boldsymbol{\sigma})$ becomes

$$\begin{aligned}\partial_{\boldsymbol{\sigma}} J(\boldsymbol{\sigma}) &= c\boldsymbol{K}\boldsymbol{\sigma} + \sum_{i=1}^{n} \boldsymbol{K}_i \partial l(z_i, \boldsymbol{K}_i^\top \boldsymbol{\sigma}) \\ &= c\boldsymbol{K}\boldsymbol{\sigma} + \boldsymbol{K}\boldsymbol{\xi}, \end{aligned} \tag{3}$$

where $\partial l(\cdot\,,\cdot\,)$ denotes the partial derivative of the loss function w.r.t. its second argument, and $\boldsymbol{\xi}$ is some vector in $\mathbb{R}^n$. Using (3), the optimization problem (2) can be solved via the following update

$$\boldsymbol{\sigma}_{t+1} \leftarrow \boldsymbol{\sigma}_t - \eta_t \boldsymbol{B}_t \partial_{\boldsymbol{\sigma}} J(\boldsymbol{\sigma}_t), \tag{4}$$

where $\eta_t > 0$ is a scalar step size and $\boldsymbol{B}_t$ is a positive definite matrix.

Note that if $\boldsymbol{B}_t = \boldsymbol{I}$, (4) becomes the standard gradient descent method. Chapelle (2006) derives the Hessian matrix $\boldsymbol{H}_t = \partial^2_{\boldsymbol{\sigma}} J(\boldsymbol{\sigma}_t)$ analytically and obtains the Newton direction as $\boldsymbol{B}_t = \boldsymbol{H}_t^{-1}$. Promising experimental results notwithstanding, Newton's method has intrinsic disadvantages:

1. The analytical Hessian may be singular, requiring ad-hoc modifications to keep it invertible;
2. Fitting a non-sparse Hessian matrix into memory might not be feasible;
3. Inverting the Hessian may incur $O(n^3)$ complexity in the worse case.

Replacing Newton's method by a limited-memory quasi-Newton method like LBFGS addresses the above problems. To further accelerate the training process, we are now applying our online version of LBFGS (oLBFGS) to this situation. Our early experiments suggest that we can substantially accelerate SVM training relative to the work of Chapelle (2006) on large-scale problems. We expect to have compelling experimental results to report at the workshop.

References

N. N. Schraudolph, J. Yu, and S. Günter. A Stochastic Quasi-Newton Method for Online Convex Optimization. In *Proc. 11th Intl. Conf. Artificial Intelligence and Statistics (AIstats)*, San Juan, Puerto Rico, 2007. Society for Artificial Intelligence and Statistics.

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer, 1999.

O. Chapelle. Training a support vector machine in the primal. Technical Report TR.147, Max Planck Institute for Biological Cybernetics, 2006.