Gradient-type methods for primal SVM model selection

Gregory Moore, Charles Bergeron and Kristin P. Bennett
Department of Mathematical Sciences
Rensselaer Polytechnic Institute
Troy, NY 12180
{mooregm,chbergeron}@gmail.com, bennek@rpi.edu

Abstract

Selection of multiple SVM model hyperparameters by cross-validation can be expressed as a bilevel optimization problem. *Explicit* methods optimize the model parameters and hyperparameters simultaneously. In *implicit* methods, the model parameters are considered to be implicit functions of the hyperparameters. Recently, gradient-based methods have emerged as an efficient way to optimize the hyperparameters. In this work, we examine both implicit and explicit model selection algorithms for linear SVM-type machine learning models expressed as nonsmooth unconstrained optimization problems. A key point is that the underlying optimization problems are nonsmooth and nonconvex, so set-valued subgradients must be used. Nonsmooth nonconvex optimization techniques can lead to scalable model selection algorithms but appropriate choice and use of subgradients is essential for good performance. A new nonconvex implicit bundle method is developed and compared computationally to recent nonsmooth implicit and explicit gradient methods and grid search. All of the gradient methods out perform grid search. The subgradients calculated using the simple implicit method may not yield good directions, leading to algorithm failures. Smaller datasets can benefit from the implicit bundle algorithms that have specialized strategies for calculating effective subgradients. The well-founded *explicit* method consistently provides robust solutions for all size problems but at greater computational expense for larger problems.

Selection of multiple SVM [18] model hyperparameters, such as the trade-off parameters C and tube-width ϵ in SVR, is important for achieving good generalization, but is an often overlooked issue. A common approach is to use cross-validation (CV) coupled with grid search, but this becomes unmanageable when there are more than 2 or 3 hyperparameters. We consider CV formulated as a bilevel minimization problem to optimize the hyperparameters efficiently.

In this paper we will consider a generalized SVM problem with model parameters $\mathbf{w}$ and hyperparameters $\boldsymbol{\gamma}$. We define Γ to be a convex set of hyperparameters of interest and for simplicity of presentation restrict the problem to a single training and validation set. The goal is to select model parameters such that the model resulting from optimizing the training problem $\mathcal{L}_{\text{trn}}(\mathbf{w}, \boldsymbol{\gamma})$ minimizes the generalized validation function $\mathcal{L}_{\text{val}}(\mathbf{w}, \boldsymbol{\gamma})$. This can be formulated as a bilevel problem [1]:

$$\begin{array}{ll} \min\limits_{\mathbf{w},\boldsymbol{\gamma}} & \mathcal{L}_{\text{val}}(\mathbf{w}, \boldsymbol{\gamma}) \\ \text{s.t.} & \boldsymbol{\gamma} \in \Gamma \\ & \mathbf{w} \in \arg\min\limits_{\mathbf{w}} \ \mathcal{L}_{\text{trn}}(\mathbf{w}, \boldsymbol{\gamma}). \end{array} \tag{1}$$

The bilevel optimization problem is nonsmooth and nonconvex in its entirety. The objectives $\mathcal{L}_{\text{val}}$ and $\mathcal{L}_{\text{trn}}$ are presumed to be convex but not necessarily differentiable. We refer to algorithms that simultaneously optimize $\mathbf{w}$ and $\boldsymbol{\gamma}$ of problem (1) as *explicit methods*.

Most current model selection algorithms [3, 5, 7, 8, 11, 15, 16, 19] do not address this bilevel optimization model directly. For instance, *implicit methods* do not optimize over the hyperparameters and model parameters simultaneously. Rather implicit methods treat $\mathbf{w}$ as an implicit function of the hyperparameters $\boldsymbol{\gamma}$, written as $\mathbf{w}(\boldsymbol{\gamma})$. This changes the bilevel program into an implicit problem:

$$\begin{array}{cl} \min\limits_{\boldsymbol{\gamma}} & \mathcal{L}_{\text{val}}(\mathbf{w}(\boldsymbol{\gamma}), \boldsymbol{\gamma}) \\ \text{s.t.} & \boldsymbol{\gamma} \in \Gamma \end{array} \qquad \text{where} \qquad \mathbf{w}(\boldsymbol{\gamma}) \in \arg\min_{\mathbf{w}} \ \mathcal{L}_{\text{trn}}(\mathbf{w}, \boldsymbol{\gamma}). \tag{2}$$

This implicit problem is in general a nonconvex and nonsmooth problem. In this form it is easy to see that grid search is an implicit method. The validation function $\mathcal{L}_{\text{val}}(\mathbf{w}(\boldsymbol{\gamma}), \boldsymbol{\gamma})$ is minimized by "optimizing" over the possible hyperparameters. In grid search, optimization is done by discretizing the hyperparameter variables, and then computing the function value at each point. Each function value requires training of the model $\mathcal{L}_{\text{trn}}(\mathbf{w}, \boldsymbol{\gamma})$. "Optimization" is performed by choosing the best, or lowest, validation error.

Several papers have shown that both implicit and explicit gradient-based methods can lead to much more computationally efficient model selection algorithms than grid search [3, 4, 9, 10, 12]. Since the function is not differentiable everywhere, we use the term gradient loosely to refer to subderivatives or elements of the Clarke-subdifferential [14].

This paper examines model selection algorithms that treat the linear SVM training problem directly as nonsmooth unconstrained minimization which contrasts with prior methods that uses the less efficient smooth SVM formulation [2, 3, 4, 9, 10, 12]. First we examine the simple nonsmooth implicit gradient descent algorithm (`ImpGrad`) and a novel and more robust nonsmooth and nonconvex bundle algorithm (`ImpBundle`). A primal nonsmooth explicit method (`PBP`) that solves for the hyperparameters and model hyperparameters simultaneously is then introduced. Computational results show that appropriate choice and use of subgradients are key for the performance of the algorithms for small problems. However, for massive problems, simplified approaches can create models that generalize just as well while benefiting from faster computational performance.

1 Existing Dual Implicit Gradient Descent Algorithms

Dual implicit gradient descent methods [3, 9] optimize the hyperparameters in problem (2) using gradient information calculated using the dual SVM version of the training problem in (2). Assuming the Representer Theorem applies, the standard SVM approach of smoothing the training problem in (2) by adding nonnegative slack variables and then taking the dual produces a convex linearly constrained problem in the dual variables $\boldsymbol{\alpha}$ with at least one variable per training point. For simplicity of presentation, we denote the constraints of the dual problem as $A\boldsymbol{\alpha} \leq b$ where A and b are an appropriately defined matrix and vector, respectively, which may themselves be functions of $\boldsymbol{\gamma}$. The dual implicit problem becomes:

$$\begin{array}{cl} \min\limits_{\boldsymbol{\gamma}} & \mathcal{L}_{\text{val}}(\mathbf{w}(\boldsymbol{\gamma}), \boldsymbol{\gamma}) \\ \text{s.t.} & \mathbf{w}(\boldsymbol{\gamma}) = X'\boldsymbol{\alpha}(\boldsymbol{\gamma}) \\ & \boldsymbol{\gamma} \in \Gamma \end{array} \tag{3}$$

where

$$\boldsymbol{\alpha}(\boldsymbol{\gamma}) \in \begin{array}{cl} \arg\min\limits_{\boldsymbol{\alpha}} & \mathcal{L}^{\text{dual}}_{\text{trn}}(\boldsymbol{\alpha}, \boldsymbol{\gamma}) \\ \text{s.t.} & A\boldsymbol{\alpha} \leq b. \end{array} \tag{4}$$

Problem (4) is solved directly to find $\mathbf{w}$ using any appropriate SVM algorithm. Then a subgradient is calculated using the Karush-Kuhn-Tucker (KKT) optimality conditions:

$$\mathbf{0} = \frac{\partial \mathcal{L}^{\text{dual}}_{\text{trn}}(\alpha, \gamma)}{\partial \alpha} + A'\sigma \tag{5}$$

$$A\alpha \leq b \tag{6}$$

$$\sigma \geq 0 \tag{7}$$

$$\sigma'(A\alpha - b) = 0, \tag{8}$$

where σ is the dual variable to the constraints $A\boldsymbol{\alpha} \leq b$.

For typical linear or quadratic SVM training problems, the first three constraints are linear. The last constraint is a complementary constraint, it enforces that either $A_i\alpha - b_i = 0$ or $\sigma_i = 0$ for each constraint i. In prior methods, this constraint is ignored under that assumption that the active support vector set does not change at the current point. This assumption is typically false. Further, the inactive inequalities are also ignored, as for a small enough step, they will remain inactive. This reduces the KKT system to a linear system from which a gradient estimate can be readily computed [3, 9]. However this may not be a good choice of subgradient since it represents one of many possible choices and may not yield a descent direction or even a directional derivative.

Previously in [9] a standard Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton algorithm with the above subgradient is used to optimize the hyperparameters. While this method is *not* guaranteed to converge to a locally optimal solution for subdifferentiable problems, it can work quite well in practice for large problems. For the case when the loss function is not twice differentiable, the subgradient selected may not be valid, and thus may yield a poor search direction. Consequently the algorithm may fail to make significant progress and exit at a non-locally optimal point. Further the BFGS algorithm is not applicable without modification to nonconvex functions, and thus this can create additional computational errors. The method requires an optimal solution of the more costly dual SVM problem. Linear scalability of primal SVM has been achieved by using subgradient methods on the unconstrained nonsmooth primal SVM problem with no introduction of dual variables [17], suggesting that primal nonsmooth methods for model selection may also be more computationally effective.

2 Primal Implicit Methods `ImpGrad` and `ImpBundle`

Increased efficiency for primal SVM model selection can be achieved by working directly on the unconstrained nonsmooth SVM training problem. We devised two approaches: `ImpGrad` that directly generalizes the prior dual implicit approach to the primal case and `ImpBundle` that uses a more robust nonsmooth nonconvex bundle method.

These algorithms solve formulation (2) with the training problem as an unconstrained problem, thus the training problem's optimality condition is

$$\mathbf{0} \in \frac{\partial \mathcal{L}_{\text{trn}}(\mathbf{w}, \boldsymbol{\gamma})}{\partial \mathbf{w}}. \tag{9}$$

If $\mathcal{L}_{\text{trn}}$ is nonsmooth, then it is a set-valued constraint. If we assume that $\mathcal{L}_{\text{trn}}(\mathbf{w}, \boldsymbol{\gamma})$ is once differentiable, like for ridge regression, least squares SVM and quadratically penalized ϵ-insensitive regression, then the constraint becomes an equality constraint. This equation is analogous to the KKT system used in the dual implicit methods. Using this constraint, the subgradient of $\mathcal{L}_{\text{val}}$ with respect to $\boldsymbol{\gamma}$ can be computed, but it may not be unique. The `ImpGrad` algorithm selects an arbitrary subgradient to act as a gradient, and then proceeds similarly to the previous implicit method with a BFGS algorithm. `ImpGrad` similarly assumes that the problem is smooth and convex, of which it is neither.

The more advanced `ImpBundle` algorithm does *not* assume that the problem is smooth or convex. It does not use a BFGS algorithm, rather it uses a nonconvex, nonsmooth bundle algorithm to optimize the hyperparameters subject to linear constraints. Much like convex bundle algorithms, `ImpBundle` builds up a piecewise linear approximation of the implicit function using subgradients that are locally valid about the stability center. The approximation is improved until a better solution is found and then the stability center is updated. To deal with nonconvexity, the subgradients can be used as lower or upper approximations of the implicit function as appropriate such as in [6].

In a novel strategy designed for fast performance, `ImpBundle` selects subgradients that correspond to directional derivatives. `ImpBundle` uses the nonconvex nonsmooth bundle method in [6] except that the bundle is *retroactively* revised. Specifically, if a sufficient decrease is not obtained (i.e. a null step is found), the subgradient of the stability center is revised to correspond to the directional derivative corresponding to the last search direction. This subgradient is computationally efficient, but weaker than the directional derivative found in [13]. Revising the subgradient to be a directional derivative improves the accuracy of the piecewise linear approximation in the direction of interest, accelerating convergence.

Note that all existing implicit algorithms require training the SVM problem to optimality at each iteration for each function/gradient calculation of the algorithm. Intuitively, explicit methods for solving for the hyperparameters and model parameters simultaneously have the potential of being more effective.

3 Primal Explicit Method: PBP

In contrast with the implicit methods, explicit methods solve problem (1) directly for both the model parameters and hyperparameters. We focus on the penalized bilevel programming (PBP) [12] approach, which further assumes the training objective $\mathcal{L}_{\text{trn}}$ is once differentiable such as in least squares SVM. This method replaces the primal unconstrained training problem by its optimality condition:

$$\begin{array}{ll} \min\limits_{\mathbf{w},\boldsymbol{\gamma}} & \mathcal{L}_{\text{val}}(\mathbf{w},\boldsymbol{\gamma}) \\ \text{s.t.} & \mathbf{0} = \frac{\partial \mathcal{L}_{\text{trn}}(\mathbf{w},\boldsymbol{\gamma})}{\partial \mathbf{w}} \quad \boldsymbol{\gamma} \in \Gamma. \end{array} \tag{10}$$

The new equality constraint is nonsmooth and nonconvex. This constraint is difficult to deal with, so it is penalized into the objective with penalty parameter β:

$$\begin{array}{ll} \min\limits_{\mathbf{w},\boldsymbol{\gamma}} & \mathcal{L}_{\text{val}}(\mathbf{w},\boldsymbol{\gamma}) + \beta \| \frac{\partial \mathcal{L}_{\text{trn}}(\mathbf{w},\boldsymbol{\gamma})}{\partial \mathbf{w}} \|^2 \\ \text{s.t.} & \boldsymbol{\gamma} \in \Gamma. \end{array} \tag{11}$$

This new problem is nonconvex and nonsmooth. To solve it, a locally accurate quadratic approximation function is created about a stability center. The approximation function is then minimized, and the stability center and approximation updated. This approximation is constructed such that the greatest feasible descent direction is chosen. This advanced search ensures that the algorithm optimizes the function efficiently and also determines if the necessary conditions for optimality are satisfied (no descent possible). The reader should consult [12] for full details of the PBP algorithm and its convergence to a point satisfying necessary optimality criteria.

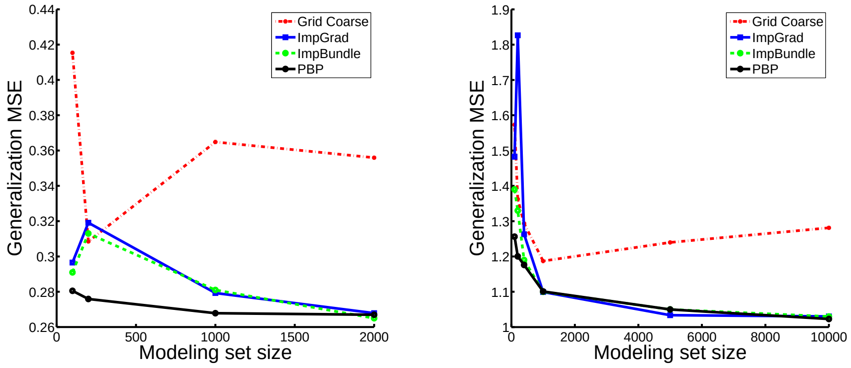
4 Results

PBP, ImpGrad and ImpBundle algorithms are compared against grid search using a pair of quantitative structure activity relationship (QSAR) datasets arising in drug design. This regression task is to predict a measure of the bioactivity of molecules, based on computed molecular descriptors. A full description of the experimental design appears in [12]. There are two datasets: pyruvate kinase and tau-fibril. Model selection was performed on dataset sizes ranging from 100 points to 1,0000 molecules.

The learning task chosen is multiSVR modeling with 10 hyperparameters [12]. This is a generalized version of ϵ-insensitive regression where each dataset is split into 5 evenly-sized groups according to sample quality. MultiSVR builds regression models with the flexibility of different hyperparameter values for each group. Hence, the multiSVR problem involves 10 hyperparameters. Computing a normal fine grid search with 10 hyperparameters for many possible values per hyperparameter is far beyond our computational powers. Therefore, we have restricted grid search to a *coarse grid* search, where for each of the hyperparameters, only two possible values are used.

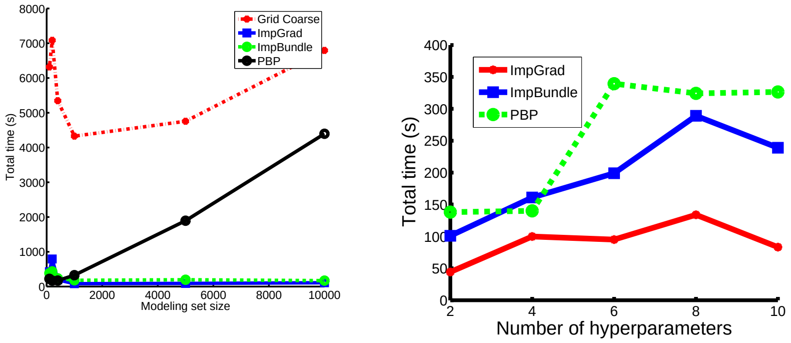
Figure 1 presents generalization results for both datasets. Note that PBP find the smallest generalization error for all model sizes (or is essentially tied). Further, as the sample size increases, ImpGrad and ImpBundle improve their generalization error with respect to PBP to attain a negligible difference on the larger datasets. Similarly, ImpGrad and ImpBundle perform very comparably on all runs except the 200 data point tau-fibril dataset. Likely ImpGrad was “stuck” here. Finally, Grid search returned the worst generalization errors. This was expected as it is limited to a discrete sampling of hyperparameter combinations; the other algorithms are not so constrained.

We seek to develop algorithms that scale linearly with sample size. Figure 2A assesses the empirical scalability of the algorithms presented in this paper. Coarse grid search uses more computational time than the other algorithms at 100 modeling points, and then its computational time begins to grow as expected after 1,000 points. ImpGrad, ImpBundle and PBP scale modesty. PBP is consistent, but grows at a higher rate than ImpGrad and ImpBundle. For larger datasets ImpGrad was the fastest, followed closely by ImpBundle. This suggests a hybrid approach may be applicable. Small datasets would benefit from better generalization error using PBP and large datasets would benefit from the speed of ImpBundle.



(A) (B)

Figure 1: Generalization results for the (A) pyruvate kinase and (B) tau-fibril dataset using multiSVR with 5-groups. The plots show that the coarse grid does poorly and PBP does the best. Notice that PBP, ImpGrad and ImpBundle converge for the large datasets, but for smaller tau-fibril datasets ImpGrad is less reliable.



(A) (B)

Figure 2: Scalability in the (A) sample and (B) hyperparameter sizes for grid search, ImpGrad, ImpBundle and PBP with 10 hyperparameters for the tau-fibril dataset as measured by CPU time. Note the coarse grid is only selecting over two choices per hyperparameters; a full grid search is impractical.

Figure 2B addresses the algorithms hyperparameter scalability using the tau-fibril $1,000$ molecule datasets. The plot shows that as the number of hyperparameters (groups) are increased from two hyperparameters (one group) to 10 hyperparameters (five groups), the computation time grows slowly in the hyperparameter size. Grid search CPU time grows exponentially and is omitted.

5 Conclusions

This paper provides an insight into selecting model hyperparameters using the bilevel optimization framework. Both implicit and explicit gradient methods have been developed that work directly on the primal nonsmooth training objectives versus prior methods that used smoothed SVM formulations. Appropriate choice and use of subgradients is key for robust performance in both implicit and explicit algorithms. The simple ImpGrad descent algorithm can fail for smaller problems. By incorporating directional derivatives within a nonsmooth nonconvex bundle framework, ImpBundle improves generalization with no increase in computational cost. A deeper theoretical investigation may lead to further improvements in ImpBundle. The theoretically well-founded explicit PBP algorithm achieves the best generalization overall but at greater computational costs. The implicit strategy automatically decomposes the problem into separate problems for each CV fold, while the explicit PBP algorithm solves for all CV folds simultaneously. A decomposition approach based on the CV folds should significantly speed up the explicit algorithm and make it more comparable to the implicit methods in terms of efficiency. For the very large problems there is no real difference in the generalization of the approaches, suggesting cheaper and simple optimization algorithms without accurate subgradient calculations can be quite effective given sufficient data. The results suggest that model selection using bilevel gradient methods is preferable over grid search for problems with many hyperparameters.

References

[1] K. P. Bennett, J. Hu, X. Ji, G. Kunapuli, and J.-S. Pang. Model selection via bilevel optimization. *International Joint Conference on Neural Networks*, pages 1922–1929, 2006.

[2] K. P. Bennett, G. Kunapuli, J. Hu, and J.-S. Pang. Bilevel optimization and machine learning. In J. Zurada and et al, editors, *Computational Intelligence: Research Frontiers: IEEE WCCI 2008, Hong Kong, China, June 1-6, 2008 : Plenary/invited Lectures*, volume 5050 of *Lecture Notes in Computer Science*, pages 25–47. Springer, 2008.

[3] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1–3):131–159, 2002.

[4] C. Do, C.-S. Foo, and A. Ng. Efficient multiple hyperparameter learning for log-linear models. In *Advances in Neural Information Processing Systems 20*, pages 377–384, 2008.

[5] K. Duan, S. Keerthi, and A. Poo. Evaluation of simple performance measures for tuning SVM hyperparameters. *Neurocomputing*, 51:41–59, 2003.

[6] A. Fuduli, M. Gaudioso, and G. Giallombardo. Minimizing nonconvex nonsmooth functions via cutting planes and proximity control. *SIAM J. on Optimization*, 14(3):743–756, 2003.

[7] G. Golub, M. Heath, and G. Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223, 1979.

[8] T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu. The entire regularization path for the support vector machine. *Jour. of Machine Learning Research*, 5:1391–1415, 2004.

[9] S. Keerthi, V. Sindhwani, and O. Chapelle. An efficient method for gradient-based adaptation of hyperparameters in SVM models. In *Neural Information Processing Systems 18*, pages 673–680, 2006.

[10] G. Kunapuli, K. P. Bennett, and J.-S. Pang. Bilevel model selection for support vector machines. In P. Pardalos and P. Hansen, editors, *Data Mining and Mathematical Programming*, volume 45, pages 129–158. AMS, 2008.

[11] J.T. Kwok and I.W. Tsang. Linear dependency between ϵ and the input noise in ϵ-support vector regression. *Neural Networks*, 14(3):544–553, 2003.

[12] G. Moore, C. Bergeron, and K. Bennett. Model selection for primal SVM. In *Machine Learning*, to appear.

[13] J.-S. Pang and D. Sun. First-order sensitivity of linearly constrained strongly monotone composite variational inequalities. Technical report, 2008.

[14] R. Tyrrell Rockafellar and Roger J-B Wets. *Variational Analysis*. Springer, New York, 1998.

[15] S. Rosset. Tracking curved regularized optimization solution paths. In *Advances in Neural Information Processing Systems*. MIT Press, 2004.

[16] M. Seeger. Bayesian model selection for support vector machines, Gaussian processes and other kernel classifiers. *Advances in Neural Information Processing Systems*, pages 603–609, 1999.

[17] C.H. Teo, A. Smola, SVN Vishwanathan, and Q.V. Le. A scalable modular convex solver for regularized risk minimization. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 727–736. ACM, 2007.

[18] V. N. Vapnik. *The Nature of Statistical Learning Theory, Second Edition*. Springer-Verlag, New York, 2000.

[19] J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani. 1-norm support vector machines. In *Advances in Neural Information Processing Systems 16*, pages 49–56, 2004.