A homotopy approach for nonconvex disjunctive programs in machine learning

Kohei Ogawa
Nagoya Institute of Technology
Gokiso-cho, Showa-ku, Nagoya, Japan
ogawa.mllab.nit@gmail.com

Ichiro Takeuchi
Nagoya Institute of Technology
Gokiso-cho, Showa-ku, Nagoya, Japan
takeuchi.ichiro@nitech.ac.jp

Masashi Sugiyama
Tokyo Institute of Technology
O-okayama, Meguro-ku, Tokyo, Japan
sugi@cs.titech.ac.jp

Abstract

We study a class of non-convex problems that can be formulated as a disjunctive program whose feasible region is represented as a *union* of convex polytopes. We introduce a novel homotopy-type optimization framework for this class of non-convex problems. As an example of such problems, we mainly focus in this paper on semi-supervised SVM (S^3VM) and develop a homotopy-type S^3VM algorithm. Numerical experiments on S^3VM demonstrate the advantage of our framework over other non-convex optimization methods such as CCCP.

1 Introduction

We consider a class of non-convex problems that contains many important machine learning problems such as semi-supervised learning, robust learning, and multi-instance learning. A common feature of these problems is that they can be formulated as *disjunctive programs* [1]. Disjunctive program is an optimization problem over a *union* (not an *intersection*) of convex polytopes. The goal of this paper is to develop a common optimization framework for these problems by exploiting the disjunctive structure.

We introduce the *homotopy method* [2]-like approach for these problems. The homotopy method handles a family of optimization problems parametrized by a so-called homotopy parameter $\theta \in [0, 1]$. Our class of non-convex problems is written as

$$\min \quad \text{(convex function)} + \theta \text{ (non-convex function)}.$$

We also start from the solution at $\theta = 0$, where the problem is convex and the global optimal solution can be easily computed. Then, we compute a sequence of local optimal solutions as θ moves from 0 to 1, and finally obtain a local optimal solution of our target non-convex problem at $\theta = 1$.

In this paper, we implement this idea by using parametric programming (a.k.a. path-following) [2] and develop an algorithm that can compute a path of local optimal solutions from $\theta = 0$ to 1. Unlike conventional path-following methods, we have to handle *discontinuity* of the solution path when the solution moves from one convex polytope to another[1].

[1]Usually, homotopy method refers to a method that computes a continuous path of solutions. In this sense, it might be misleading to call our approach as a homotopy method. Actually, we will show that there exists no continuous path of local solutions in the class of problems we consider here.

We overcome this difficulty by examining the local optimality conditions. The conditions we derive here can tell us which polytope contains a better solution, and help us to develop an algorithm that can provably compute the entire path of local optimal solutions for all $\theta \in [0, 1]$ with a finite number of iterations.

As an example of such non-convex problems, we mainly focus in this paper on semi-supervised SVM (S^{3}VM) [3]. We discuss how S^{3}VM can be formulated as a disjunctive program, and develop a homotopy-type S^{3}VM algorithm. Numerical experiments on S^{3}VM demonstrate the advantage of our approach over other non-convex optimization algorithms such as CCCP [4].

2 Semi-supervised SVM

In this section, we formulate semi-supervised SVM (S^{3}VM) as a disjunctive program for the purpose of illustrating the class of problems to which our framework can be applied.

Consider a binary classification with n instances $\{(\boldsymbol{x}_i, y_i)\}_{i\in\mathcal{L}}$, where $\boldsymbol{x}_i$ is the input vector, $y_i \in \{-1, 1\}$ is the binary class label, and $\mathcal{L} := \{1, \ldots, n\}$. For simplicity, we consider linear decision function $f(\boldsymbol{x}) = w_0 + \boldsymbol{w}^\top \boldsymbol{x}$. The SVM is formulated as a convex optimization problem: $\min_f \frac{1}{2}||\boldsymbol{w}||_2^2 + C\sum_{i\in\mathcal{L}}[1 - y_i f(\boldsymbol{x}_i)]_+$, where C is the regularization parameter and $[h]_+ := \max(0, h)$ is the hinge-loss.

In semi-supervised learning, we also have unlabeled instances $\{\hat{\boldsymbol{x}}_i\}_{i\in\mathcal{U}}$, where $\mathcal{U}$ stands for the unlabeled instance set. The S^{3}VM is SVM-like semi-supervised classification algorithm [3]. It is formulated as minimization w.r.t. the decision function f and the predicted labels $\hat{y}_i \in \{-1, 1\}, i \in \mathcal{U}$, for the unlabeled instances:

$$\min_{f,\hat{\boldsymbol{y}}} J(f, \hat{\boldsymbol{y}}) \equiv \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_{i\in\mathcal{L}}[1 - y_i f(\boldsymbol{x}_i)]_+ + \theta C\sum_{i\in\mathcal{U}}[1 - \hat{y}_i f(\hat{\boldsymbol{x}}_i)]_+ \ \text{ s.t. } \ \hat{y}_i f(\hat{\boldsymbol{x}}_i) \geq 0, i \in \mathcal{U},$$

where $\theta \in [0, 1]$ controls the relative amount of influences of the unlabeled instances on f; $\theta = 0$ indicates no influence (the standard SVM), while $\theta = 1$ indicates the same amount of influences as the labeled instances. The constraints in (1) require that the predicted labels $\hat{\boldsymbol{y}}$ should be consistent with the decision function f. The problem (1) is a non-convex optimization problem since $\hat{y}_i f(\hat{\boldsymbol{x}}_i)$ is written as $|f(\hat{\boldsymbol{x}}_i)|$ under the constraints $\hat{y}_i f(\hat{\boldsymbol{x}}_i) \geq 0, i \in \mathcal{U}$ (see Figure 1 (left)).

To formulate S^{3}VM as a disjunctive program, for each $\boldsymbol{z} \in \{-1, 1\}^{|\mathcal{U}|}$, define a convex polytope

$$\text{pol}(\boldsymbol{z}) := \{(f, \hat{\boldsymbol{y}}) | z_i f(\hat{\boldsymbol{x}}_i) \geq 0, \hat{y}_i = z_i, \forall i \in \mathcal{U}\}. \tag{1}$$

Then, S^{3}VM is formulated as minimization over the union of the convex polytopes:

$$\min_{f,\hat{\boldsymbol{y}}} \ J(f, \hat{\boldsymbol{y}}) \ \text{ s.t. } \ (f, \hat{\boldsymbol{y}}) \in \bigcup_{\boldsymbol{z}\in\{-1,1\}^{|\mathcal{U}|}} \text{pol}(\boldsymbol{z}). \tag{2}$$

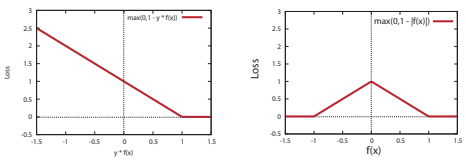

Figure 1: Convex hinge loss for labeled instances (left) and non-convex symmetric hinge loss for unlabeled instances (right) in S^{3}VM.

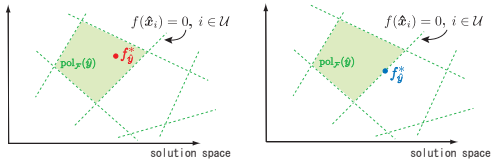

Figure 2: The conditionally optimal solution $f^*_{\hat{\boldsymbol{y}}}$ in the interior is local optimal (left), but $f^*_{\hat{\boldsymbol{y}}}$ at the boundary is NOT local optimal (right).

3 Homotopy algorithm for S^{3}VM

As we saw, the search spaces of S^{3}VM and robust SVM are represented as a union of convex polytopes. In this section, we develop a homotopy-type algorithm for S^{3}VM by exploiting the disjunctive structure. Similar algorithms for robust SVM or other non-convex problems that can be formulated as similar disjunctive programs can be constructed in the same way.

Local optimality conditions Let us start from rewriting the problem (2) as follows:

$$\min_{\hat{\boldsymbol{y}}}\big\{\min_{f} J(f,\hat{\boldsymbol{y}}) \text{ s.t. } f \in \mathrm{pol}_{\mathcal{F}}(\hat{\boldsymbol{y}})\big\}, \tag{3}$$

where $\mathrm{pol}_{\mathcal{F}}(\hat{\boldsymbol{y}}) := \{f | \hat{y}_i f(\hat{\boldsymbol{x}}_i) \geq 0, \forall i \in \mathcal{U}\}$ is a convex polytope obtained by projecting $\mathrm{pol}(\boldsymbol{z})$ onto the space of f after fixing $\boldsymbol{z} = \hat{\boldsymbol{y}}$, i.e., $\mathrm{pol}_{\mathcal{F}}(\hat{\boldsymbol{y}})$ is the feasible region of f when $\hat{\boldsymbol{y}}$ is fixed. The next definition states that we have a convex optimization problem over each $\mathrm{pol}_{\mathcal{F}}(\hat{\boldsymbol{y}})$.

Definition 1 (conditionally optimal solution $f^*_{\hat{\boldsymbol{y}}}$) *The inner optimization problem in (3) is convex, and we call its optimal solution $f^*_{\hat{\boldsymbol{y}}} := \arg\min_f J(f,\hat{\boldsymbol{y}})$ s.t. $f \in \mathrm{pol}_{\mathcal{F}}(\hat{\boldsymbol{y}})$ as a conditionally optimal solution for $\hat{\boldsymbol{y}}$.*

Note that there are $2^{|\mathcal{U}|}$ conditionally optimal solutions for each $\hat{\boldsymbol{y}} \in \{-1,1\}^{|\mathcal{U}|}$. By examining the necessary and sufficient conditions for the local optimality, we can characterize which conditionally optimal solutions are local optimal solutions of the original problem (3) (see Figure 2).

Theorem 2 *For each $\hat{\boldsymbol{y}}$, if the conditionally optimal solution $f^*_{\hat{\boldsymbol{y}}}$ is in the strict interior of the convex polytope $\mathrm{pol}_{\mathcal{F}}(\hat{\boldsymbol{y}})$, then $(f^*_{\hat{\boldsymbol{y}}},\hat{\boldsymbol{y}})$ is a local optimal solution of (3). On the other hand, if $f^*_{\hat{\boldsymbol{y}}}$ is at the boundary of the convex polytope $\mathrm{pol}_{\mathcal{F}}(\hat{\boldsymbol{y}})$, then $(f^*_{\hat{\boldsymbol{y}}},\hat{\boldsymbol{y}})$ is NOT a local optimal solution.*

In case a conditionally optimal solution is located at the boundary of the current convex polytope, we must move to another convex polytope for finding a local optimal solution. The following theorem tells us which convex polytope we should move to.

Theorem 3 *Suppose that the conditionally optimal solution $f^*_{\hat{\boldsymbol{y}}}$ for $\hat{\boldsymbol{y}}$ is at the boundary of the convex polytope $\mathrm{pol}_{\mathcal{F}}(\hat{\boldsymbol{y}})$, i.e., $\hat{y}_i f(\hat{\boldsymbol{x}}_i) = 0$ for at least one $i \in \mathcal{U}$. If we define*

$$\hat{\boldsymbol{y}}'_i := \begin{cases} -\hat{y}_i, & \text{if } \hat{y}_i f(\hat{\boldsymbol{x}}_i) = 0, \\ \hat{y}_i, & \text{if } \hat{y}_i f(\hat{\boldsymbol{x}}_i) > 0, \end{cases} \tag{4}$$

then, $(f^*_{\hat{\boldsymbol{y}}'},\hat{\boldsymbol{y}}')$ is a strictly better feasible solution of the original problem (3) than $(f^*_{\hat{\boldsymbol{y}}},\hat{\boldsymbol{y}})$, i.e., $J(f^*_{\hat{\boldsymbol{y}}'},\hat{\boldsymbol{y}}') < J(f^*_{\hat{\boldsymbol{y}}},\hat{\boldsymbol{y}})$.

Sketches of the proofs of Theorems 2 and 3 We can prove Theorem 2 by comparing the KKT optimality conditions of the two convex optimization problems defined over $\mathrm{pol}_{\mathcal{F}}(\hat{\boldsymbol{y}})$ and $\mathrm{pol}_{\mathcal{F}}(\hat{\boldsymbol{y}}')$. From these optimality conditions, we can show that, if the conditionally optimal solution $f^*_{\hat{\boldsymbol{y}}}$ is at the boundary of $\mathrm{pol}_{\mathcal{F}}(\hat{\boldsymbol{y}})$, it is a primal feasible *sub*optimal solution of the latter problem defined over $\mathrm{pol}_{\mathcal{F}}(\hat{\boldsymbol{y}}')$ by the fact that $f^*_{\hat{\boldsymbol{y}}}$ does not satisfy the KKT optimality of the latter convex problem. It suggests that the latter convex problem defined over $\mathrm{pol}_{\mathcal{F}}(\hat{\boldsymbol{y}}')$ has a strictly better feasible solution than $f^*_{\hat{\boldsymbol{y}}}$, i.e., $f^*_{\hat{\boldsymbol{y}}}$ is not a local optimal solution of the original problem. Theorem 3 is a direct consequence of the above discussion. When $f^*_{\hat{\boldsymbol{y}}}$ is at the boundary of $\mathrm{pol}_{\mathcal{F}}(\hat{\boldsymbol{y}})$, $f^*_{\hat{\boldsymbol{y}}'}$, the conditionally optimal solution of the convex problem obtained by flipping the labels as in (4), is strictly better than $f^*_{\hat{\boldsymbol{y}}}$.

Non-convex homotopy algorithm for S^{3}VM Our homotopy algorithm for the disjunctive program (2) consists of two steps: the *continuous path (CP)* step and the *discrete jump (DJ)* step. In the CP-step, a path of local solutions is computed within a single convex polytope. Since the inner problem of (3) is a convex parametric QP, we can efficiently compute the solution path by exploiting its piecewise-linearity [2][2] If the piecewise-linear solution path within the current convex polytope intersects with one of its boundaries, we go to the DJ-step. In the DJ-step, we use Theorem 3 for determining which convex polytope we should jump to. Thanks to the strict improvement property in Theorem 3, a path of local optimal solutions for the entire range of $\theta \in [0,1]$ can be computed with a finite number of iterations (see Algorithm 1).

[2]In adverse circumstances, parametric QP has exponential number of steps (as in the simplex method in LP). However, it has been empirically demonstrated in many past studies that the number of steps is of linear order in the number of variables.

Algorithm 1 S^3VM Homotopy

Inputs: Labeled and unlabeled instances $\{(\boldsymbol{x}_i, y_i)\}_{i\in\mathcal{L}}$ and $\{\hat{\boldsymbol{x}}_i\}_{i\in\mathcal{U}}$, regularization parameter C;
Initialize: $\theta \leftarrow 0$; $f \leftarrow$ the standard SVM solution; $\hat{y}_i \leftarrow \mathrm{sgn}(f(\hat{\boldsymbol{x}}_i))$, $i \in \mathcal{U}$;
while $\theta < 1$
  Compute $\mathcal{S} \leftarrow \{i \in \mathcal{U} \mid \hat{y}_i f(\hat{\boldsymbol{x}}_i) = 0\}$;
  if $\mathcal{S} = \emptyset$ **then** Execute the **CP-step**;
  else Execute the **DJ-step**;
Outputs: Local optimal solution path for $\theta \in [0, 1]$.

4 Numerical experiments for S^3VM

Here, we compare our homotopy-type algorithm for S^3VM ($S^3\mathrm{VM}^{\mathrm{homo}}$) with $S^3\mathrm{VM}^{\mathrm{light}}$ [3], deterministic annealing (DA) [6], and CCCP [5] on four UCI benchmark data sets: `DIGIT1` (#D1), `Breast Cancer Diagnostic` (#D2), `USPS2` (#D3), and `ESET2` (#D4) as summarized in Table 1.

First, we compare the optimization performances based on the objective values $J(f, \hat{\boldsymbol{y}})$, where f is obtained by each algorithm and $\hat{y}_i = \mathrm{sgn}(f(\hat{\boldsymbol{x}}_i)), i \in \mathcal{U}$. Figure 3 shows the results for #D1 and #D2. These plots imply that our algorithm tends to find better local optimal solutions. Next, we compare the computation time of each algorithm. Figure 4 shows the results for #D3 and #D4. In semi-supervised learning, the amount of unlabeled data influence $\theta \in [0, 1]$ must be chosen by model selection. The horizontal axis in the two plots indicate the number of model selection candidates on θ. Since our algorithm can compute the path of solutions for all $\theta \in [0, 1]$, we gain a computational advantage when the number of candidates is large. Finally, we compare the generalization performance on unlabeled and test instances. Table 1 shows the results. We see that our algorithm provides slightly better performances than others.

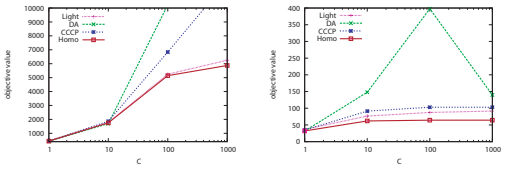

Figure 3: Objective Values (#D1 and #D2)

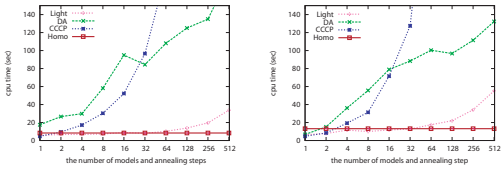

Figure 4: Computational Costs (#D3 and #D4)

Table 1: Data set descriptions and the average mis-classification performances on unlabeled and test instances. For each data set, $d/n/\ell/u/v/t$ indicates input dimension(d), the numbers of total(n), labeled(ℓ), unlabeled(u), validation(v), and test(t) instances, respectively. Boldface indicates the best (and those not significantly different from the best) performance based on t-test ($p < 0.05$).

Data	$d/n/\ell/u/v/t$	SVM	$S^3\mathrm{VM}^{\mathrm{light}}$	DA	CCCP	$S^3\mathrm{VM}^{\mathrm{homo}}$
		u/t	u/t	u/t	u/t	u/t
#D1	241/1500/45/1200/70/185	9.2/9.2	9.8/**8.9**	13.3/12.3	9.0/9.1	**7.5/7.4**
#D2	30/569/20/350/10/189	11.9/12.5	9.1/9.8	**10.5**/9.0	9.2/8.6	**8.1/7.1**
#D3	241/1500/150/1000/80/270	9.2/8.6	9.0/**8.2**	8.7/7.8	8.2/7.4	**7.2/6.6**
#D4	617/2700/80/1200/60/1360	13.1/12.9	**9.7/9.0**	**10.7/10.6**	10.3/9.8	**8.3/8.3**

References

[1] E. Balas. Disjunctive programming: properties of the convex hull of feasible points. *Discrete Applied Mathematics*, 89:3–44, 1998.

[2] E. L. Allgower and K. George. Continuation and path following. *Acta Numerica*, 2:1–63, 1993.

[3] T. Joachims. Transductive inference for text classification using support vector machines. *International Conference on Machine Learning*, 1999.

[4] A. L. Yuille and A. Rangarajan. The concave-convex procedure (cccp). In *Advances in Neural Information Processing Systems*, volume 14, 2002.

[5] R. Collobert, F. Sinz, J. Weston, and L. Bottou. Trading convexity for scalability. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 201–208, 2006.

[6] V. Sindhwani, S. Keerthi, and O. Chapelle. Deterministic annealing for semi-supervised kernel machines. *International Conference on Machine Learning*, 2006.