Optimized Techniques for Semi-Supervised Support Vector Machine

Hyunjoo Bae

Department of Statistics, Korea University

July 18, 2019

Overview

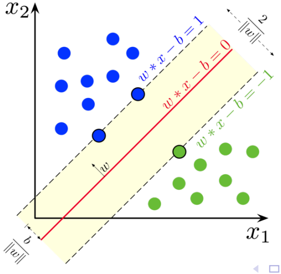
1. Introduction

Support Vector Machine

- Constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space.
- Minimize

$$\left[\frac{1}{n}\sum_{i=1}^{n}\max\left(0, 1 - y_i\left(\vec{w}\cdot\vec{x}_i - b\right)\right)\right] + \lambda\|\vec{w}\|^2 \tag{1}$$

  - λ determines the trade-off between increasing the margin size.
  - $\max(0, 1 - y_i\left(\vec{w}\cdot\vec{x}_i - b\right))$ is the hinge loss function.



2. Semi-Supervised SVM

Assumption

- Cluster Assumption for Semi-supervised learning:

Points in a data cluster have similar labels.

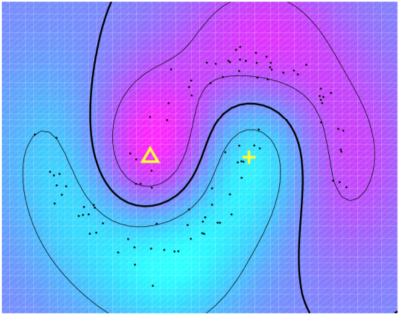


Figure: There are 2 labeled points (triangle and cross) and 100 unlabeled points.

Datasets

- training sets ($n = l + u$)
  - l labeled sets $\{(\mathbf{x}_i, y_i)\}_{i=1}^{l}, y_i = \pm 1$
  - u unlabeled sets $\{\mathbf{x}_i\}_{i=l+1}^{n}$
- Balancing constraints with the estimated ratio of positive class in the unlabeled data, r
  - $\frac{1}{u}\sum_{i=l+1}^{n} \max(y_i, 0) = r$
  - or equivalently $\frac{1}{u}\sum_{i=l+1}^{n} y_i = 2r - 1$

Semi-Supervised SVM

- Minimization problem over *both* the hyperparameters $(\mathbf{w}, b)$ and the label vector $\mathbf{y}_u := [y_{l+1} \dots y_n]^\top$

$$\min_{(\mathbf{w},b),\mathbf{y}_u} I(\mathbf{w}, b, \mathbf{y}_u) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l} V(y_i, o_i) + C^\star \sum_{i=l+1}^{n} V(y_i, o_i) \tag{2}$$

- C and $C^\star$ needs to be set different for the optimization general performance.
- $V(y_i, o_i) = \max(0, 1 - y_i o_i)^p$ is the hinge loss function. $(p = 2)$

Strategies

Combinational Optimization

Optimize over $(\mathbf{w},\ b)$ while fixing $\mathbf{y}_u$

$$J(\mathbf{y}_u) = \min_{\mathbf{w},b} I(\mathbf{w}, b, \mathbf{y}_u) \tag{3}$$

Continuous Optimization

For a fixed $(\mathbf{w}, b)$, $\arg\min_y V(y, o) = \text{sign}(o) = \text{sign}(\mathbf{w}^T\mathbf{x}_i + b)$
Non-convex optimization function will be changed into

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l} \max(0, 1 - y_i o_i)^2 + C^\star \sum_{i=l+1}^{n} \max(0, 1 - |o_i|)^2 \tag{4}$$

Strategies

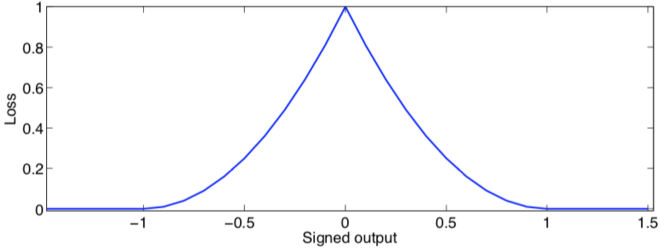


Figure: The plot of $o = \left(\mathbf{w}^\top \mathbf{x} + b\right)$ and effective loss $\max(0, 1 - |o|)^2$

3. Combinatorial Optimization

BB for global optimization

- Produce globally optimal solutions for small-sized problems

1. Perform standard SVM for labeled datasets.
2. Add additional loss for unlabeled datasets and delete the greater objective values.

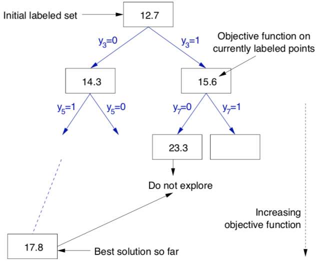


Figure: Branch-and-Bound Tree

- **lower bound at a node** and **a sequence of the unlabeled examples to branch on**

S^3VM^{light}

- Local combinational search guided by a label switching procedure
- A pair of unlabeled datasets y_i and y_j should satisfy the following condition

$$y_i = 1$$
$$y_j = -1$$
$$V(1, o_i) + V(-1, o_j) > V(-1, o_i) + V(1, o_j)$$

S^3VM^{light}

Algorithm 1 $\mathrm{S^3VM}^{light}$

Train an SVM with the labeled points. $o_i \leftarrow \mathbf{w} \cdot \mathbf{x}_i + b$.
Assign $y_i \leftarrow 1$ to the ur largest o_i, -1 to the others.
$\tilde{C} \leftarrow 10^{-5} C^\star$
while $\tilde{C} < C^\star$ **do**
  repeat
    Minimize (1) with $\{y_i\}$ fixed and $C^\star$ replaced by $\tilde{C}$.
    if $\exists(i, j)$ satisfying (6) **then**
      Swap the labels y_i and y_j
    end if
  until No labels have been swapped
  $\tilde{C} \leftarrow \min(1.5C, C^\star)$
end while

Figure: Algorithm of S^3VM^{light}

Deterministic Annealing Algorithm

- In non-convex cases, there are multiple minima and doesn't exist any local to global inference.
- Deterministic Annealing (DA) solves a related but simpler problem where simpler problem converges to the original one.
- DA computes the expectation of global quantities with respect to the Gibbs distribution.
- EM algorithm is special version of DA.

Deterministic Annealing S^3VM

- Relax the discrete label variables $\mathbf{y}_u$ to probabilities $\mathbf{p}_u$
- Objective function is changed into

$$
\begin{aligned}
I''(\mathbf{w}, b, \mathbf{p}_u; T) &= I'(\mathbf{w}, b, \mathbf{p}_u) - TH(\mathbf{p}_u) \text{ where} \\
H(\mathbf{p}_u) &= -\sum_i p_i \log p_i + (1 - p_i) \log (1 - p_i) \\
I'(\mathbf{w}, b, \mathbf{p}_u) &= E[I(\mathbf{w}, b, \mathbf{y}_u)] \\
&= \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l} V(y_i, o_i) \\
&+ C^\star \sum_{i=l+1}^{n} p_i V(1, o_i) + (1 - p_i) V(-1, o_i)
\end{aligned}
$$

Deterministic Annealing S^3VM

- Balancing constraint is

$$\frac{1}{u}\sum_{i=l+1}^{n} p_i = r$$

- $T >= 0$ is 'temperature'
    - $T = 0$, I'' reduces to $I'(\mathbf{w}, b, \mathbf{p}_u)$
    - $T = \infty$, I'' is dominated by the entropy $H(\mathbf{p}_u)$

Deterministic Annealing S^3VM

Alternating Minimization (DA)

1. Keeping $\mathbf{p}_u$ fixed, train SVM.
2. Keeping $(\mathbf{w}, u)$ fixed, I'' is minimized subject to the balance constraint $\frac{1}{u}\sum_{i=l+1}^{n} p_i = r$.
3. Then, $p_i = \frac{1}{1+e^{(g_i-\mathrm{v})/T}}$ where $g_i = C^\star \left[V(1, o_i) - V(-1, o_i)\right]$

- The alternating optimization proceeds until $\mathbf{p}_u$ stabilizes in a KL divergence.

Gradient Method (∇DA)

1. Substitue the optimal $\mathbf{p}_u$ into p_i in Alternating Minimization.
2. The gradient techniques can be used on $\mathcal{S}(\mathbf{w}, b) := \min_{\mathbf{p}_u} I''(\mathbf{w}, b, \mathbf{p}_u; T)$ which is a function of $(\mathbf{w}, b)$.

Deterministic Anealing S^3VM

Algorithm 2 DA/∇DA

Initialize $p_i = r \quad i = l+1, \ldots, n$
Set $T = 10C^\star$, $R = 1.5$, $\varepsilon = 10^{-6}$.
while $H(\mathbf{p}_{uT}) > \varepsilon$ **do**
  Solve $(\mathbf{w}_T, b_T, \mathbf{p}_{uT}) = \text{argmin}_{(\mathbf{w},b),\mathbf{p}_u} I''(\mathbf{w}, b, \mathbf{p}_u; T)$ subject to: $\frac{1}{u}\sum_{i=l+1}^{n} p_i = r$
  (find local minima starting from previous solution—alternating optimization or gradient methods can be used.)
  $T = T/R$
end while
Return $\mathbf{w}_T, b_T$

Figure: Algorithm of DA S^3VM^{light}

Deterministic Anealing S^3VM

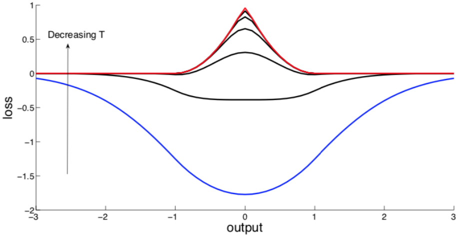


Figure: The loss functions when using T

- DA parameterizes a family of loss functions over unlabeled examples.
- As $T \to 0$, the loss function goes to the original loss function.
- $\nabla\mathrm{DA}$ is faster than DA.

- Objective function of S^3VM:

$$I(\mathbf{w}, b, \mathbf{y}_u) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l}\max(0, 1 - y_i o_i)^2$$

$$+ C^\star \sum_{i=l+1}^{n} \max(0, 1 - |o_i|)^2$$

$$= \min_{(\mathbf{w},b),\mathbf{y}_u} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l}\xi_i^2 + C^\star \sum_{i=l+1}^{n} \xi_i^2$$

subject to: $y_i o_i \geq 1 - \xi_i,\ i = 1, \ldots, n$

Convex Relaxation

- This objective function can be transformed into dual problem:

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

$$\text{subject to: } \sum_{i=1}^{n} \alpha_i y_i = 0, \alpha_i \geq 0$$

$$K_{ij} = \mathbf{x}_i^\top \mathbf{x}_j + D_{ij}$$

D is a diagonal matrix given by $D_{ii} = \dfrac{1}{2C}, i = 1 \ldots, n$

and $D_{ii} = \dfrac{1}{2C^\star}, i = l + 1, \ldots, n$

Convex Relaxation

- Optimization problem can be reformulated as:

$$\min_{\Gamma} \max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j \Gamma_{ij} K_{ij}$$

$$\text{under constraints } \sum_{i=1}^{n} \alpha_i y_i = 0, \alpha_i \geq 0, \Gamma = yy^\top$$

- The objective function above is convex.
- The constraints above are not convex, so replace the constraint $\Gamma = yy^\top$ by the following set of convex constraints.

$$\begin{aligned}
&\Gamma \succeq 0 \\
&\Gamma_{ij} = y_i y_j, \quad 1 \leq i,j \leq l \\
&\Gamma_{ii} = 1, \quad l+1 \leq i \leq n \\
&\frac{1}{u^2} \sum_{i,j=l+1}^{n} \Gamma_{ij} = (2r-1)^2
\end{aligned}$$

4. Continuous Optimization

Issues

Balancing Constraints

- Enforce a linear constraint

$$\frac{1}{u}\sum_{i=l+1}^{n} \mathbf{w}^\top \mathbf{x}_i + b = 2\tilde{r} - 1$$

- By standardization of the unlabeled points, $\sum_{i=l+1}^{n} \mathbf{x}_i = 0$
- The constraint is equivalent to $b = 2\tilde{r} - 1$ and unconstrained optimization problem on $\mathbf{w}$.

Issues

Primal Optimization

- Solve (4) with a non-convex loss function over unlabeled datasets using "kernel trick"
  - Method 1

1. Find $\mathbf{z}_i$ such that $\mathbf{z}_i \cdot \mathbf{z}_j = k(\mathbf{x}_i, \mathbf{x}_j)$
2. Let B the matrix having columns $\mathbf{z}_i$ and $K = B^\top B$.

3.1. Use Cholesky factor of K.

3.2. Perform eigen decomposition of K as $K = B^\top B$ ("kernel PCA map")

  - Method 2

1. Let $\mathbf{w} = \sum_{i=1}^{n} \beta_i \phi(\mathbf{x}_i)$
2. By the Representer theorem, the optimal solution has the above form.
3. Substitute this form in (4) and use the kernel function.

Concave Convex Procedure

- Concave Convex Procedure (CCCP) decomposes non-convex function f into a convex component f_{vex} and a concave component f_{cave}.
- Concave part is replaced by a linear function and the sum of this linear function and the convex part is minimized.

Algorithm 3 CCCP for minimizing $f = f_{vex} + f_{cave}$

Require: Starting point $\mathbf{x}_0$
  $t \leftarrow 0$
  while $\nabla f(\mathbf{x}_t) \neq 0$ **do**
    $\mathbf{x}_{t+1} \leftarrow \arg\min_{\mathbf{x}} f_{vex}(\mathbf{x}) + \nabla f_{cave}(\mathbf{x}_t) \cdot \mathbf{x}$
    $t \leftarrow t + 1$
  end while

Figure: Algorithm of CCCP for minimizing $f = f_{vex} + f_{cave}$

Concave Convex Procedure

- Objective function

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l}\max\left(0, 1 - y_i o_i\right)^2 + C^\star \sum_{i=l+1}^{n}\max\left(0, 1 - |o_i|\right)^2$$

- The first two terms are convex.
- Split the last term into convex and concave function.

$$\max(0, 1 - |t|)^2 = \underbrace{\max(0, 1 - |t|)^2 + 2|t|}_{\text{convex}} \; \underbrace{-2|t|}_{\text{concave}}$$

$$\text{where } t = y_i(\mathbf{w} \cdot \mathbf{x}_i + b)$$

- The effective loss on the unlabeled point will be

$$\tilde{L}(t) = \begin{cases} 0 & \text{if } t \geq 1 \\ (1-t)^2 & \text{if } |t| < 1 \\ -4t & \text{if } t \leq -1 \end{cases}$$

Concave Convex Procedure

Algorithm 4 CCCP for S^{3}VMs

Starting point: Use the $\mathbf{w}$ obtained from the supervised SVM solution.
repeat
$y_i \leftarrow \text{sign}(\mathbf{w} \cdot \mathbf{x}_i + b), \quad l+1 \leq i \leq n.$
$(\mathbf{w}, b) = \arg\min \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l} \max(0, 1 - y_i(\mathbf{w} \cdot \mathbf{x}_i + b))^2 + C^\star \sum_{i=l+1}^{n} \tilde{L}(y_i(\mathbf{w} \cdot \mathbf{x}_i + b)).$
until convergence of $y_i, \; l+1 \leq i \leq n.$

Figure: Algorithm of CCCP for S^3VMs

$\nabla S^3 VM$

- Minimize the objective function directly by gradient descent.
- Since $t \mapsto \max(0, 1 - |t|)^2$ is not differentiable, it is replaced by $t \longmapsto \exp\left(-st^2\right)$.
- Changed objective function:

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l} \max\left(0, 1 - y_i\left(\mathbf{w}\cdot\mathbf{x}_i + b\right)\right)^2 + C^\star \sum_{i=l+1}^{n} \exp\left(-s\left(\mathbf{w}\cdot\mathbf{x}_i + b\right)^2\right) \tag{5}$$

- s is chosen and ∇S^3**VM** performs annealing in an outer loop on $C^\star$.

Continuation S^3VM

- Continuation method for minimize (5)
- $C_\star$ is fixed and a continuation technique is used to transform the objective function.

Algorithm 5 Continuation method for solving $\min_{\mathbf{x}} f(\mathbf{x})$

Require: Function $f : \mathbb{R}^d \mapsto \mathbb{R}$, initial point $\mathbf{x}_0 \in \mathbb{R}^d$
Require: Sequence $\gamma_0 > \gamma_1 > \dots \gamma_{p-1} > \gamma_p = 0$.
  Let $f_\gamma(\mathbf{x}) = (\pi\gamma)^{-d/2} \int f(\mathbf{x} - \mathbf{t}) \exp(-\|\mathbf{t}\|^2/\gamma) d\mathbf{t}$.
  for $i = 0$ to p **do**
    Starting from $\mathbf{x}_i$, find local minimizer $\mathbf{x}_{i+1}$ of f_{γ_i}.
  end for

Figure: Algorithm of Continuation S^3VM

Continuation S^3VM

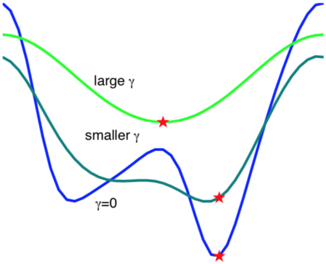


Figure: Illustration of the continuation method

- The original objective function has 2 local minima.
- By reducing the smoothing, the minimum goes toward the global minimum of the original function.

Continuation S^3VM

- In the algorithm, smoothing is achived by convolution with Gaussian.
- Other smoothing functions can also be used.
- The unlabeled part of the objective function vanishes and the optimization is identical to a standard SVM.
  1. With enough smoothing, the global minimum can be easily found.
  2. The smoothing is decreased in steps and the minimum is tracked.

Newton S^3VM

- ∇S^3VM and Continuation S^3VM requires $O\left(n^3\right)$.
- Newton S^3VM make efficiency by minimization on β where $\mathbf{w} = \sum_{i=1}^{n} \beta_i \phi\left(\mathbf{x}_i\right)$.
- Objective function is transformed into:

$$\min_{\beta} \frac{1}{2}\beta^\top K\beta + C\sum_{i=1}^{l} \ell_L\left(y_i\left(K_i^\top\beta + b\right)\right) + C^\star \sum_{i=l+1}^{n} \ell_U\left(K_i^\top\beta + b\right) \tag{6}$$

K is the kernel matrix and $K_{ij} = k\left(\mathbf{x}_i, \mathbf{x}_j\right)$
ℓ_L and ℓ_U is the general loss functions for the labeled points and the unlabeled points.

Newton S^3VM

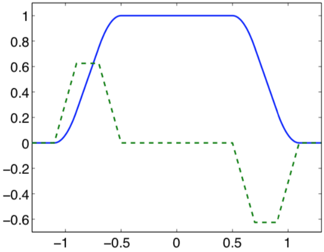


Figure: Piecewise quadratic loss function ℓ_U

Newton S^3VM

- Gradient of objective function
  $$K\mathbf{g} \text{ with } g_i = \begin{cases} \beta_i + C\ell'_L\left(y_i\left(K_i^\top\beta + b\right)\right)y_i & 1 \le i \le l \\ \beta_i + C^\star\ell'_U\left(K_i^\top\beta + b\right) & l+1 \le i \le n \end{cases}$$

- Hessian of objective function $K + KDK$,
  $$\text{with } D \text{ diagonal, } D_{ii} = \begin{cases} C\ell''_L\left(y_i\left(K_i^\top\beta + b\right)\right) & 1 \le i \le l \\ C^\star\ell''_U\left(K_i^\top\beta + b\right) & l+1 \le i \le n \end{cases}$$

- Update β as $\beta \leftarrow \beta - (K + KDK)^{-1}Kg$

Newton S^3VM

- If Hessian is not positive definite, the step might not be a descent direction.
- Use Levenberg-Marquardt Algorithm

Algorithm 6 Levenberg-Marquardt method

$\beta \leftarrow 0.$
$\lambda \leftarrow 1.$
repeat
  Compute $\mathbf{g}$ and D using (16) and (17)
  $\mathsf{sv} \leftarrow \{i,\ D_{ii} \neq 0\}$ and $\mathsf{nsv} \leftarrow \{i,\ D_{ii} = 0\}$.
  $A_{\mathsf{sv}} \leftarrow$ Cholesky decomposition of K_{sv}.
  Do the Cholesky decomposition of $\lambda I_{n_{\mathsf{sv}}} + A_{\mathsf{sv}} D_{\mathsf{sv}} A_{\mathsf{sv}}^\top$. If it fails, $\lambda \leftarrow 4\lambda$ and try again.
  Compute the step $\mathbf{s}$ as given by (18).
  $\rho \leftarrow \frac{\Omega(\beta+\mathbf{s})-\Omega(\beta)}{\frac{1}{2}\mathbf{s}^\top(K+KDK)\mathbf{s}+\mathbf{s}^\top K\mathbf{g}}$.   % If the obj fun Ω were quadratic, ρ would be 1.
  If $\rho > 0$, $\beta \leftarrow \beta + s$.
  If $\rho < 0.25$, $\lambda \leftarrow 4\lambda$.
  If $\rho > 0.75$, $\lambda \leftarrow \min(1, \frac{\lambda}{2})$.
until $\mathrm{Norm}(\mathbf{g}) \leq \varepsilon$

Figure: Algorithm of Levenberg-Marquardt Method

Newton S^3VM

- Similar with Newton minimization but a large enough ridge is added to the Hessian, so that it becomes positive definite.
- Choose a $\lambda \geq 1$ such that

$$\lambda K + KDK = A^\top \left(\lambda I_n + ADA^\top \right) A$$

A is the Cholesky decomposition of K. $(A^T A = K)$
K and A is invertible.

- $\lambda K + KDK > 0 \Leftrightarrow B := \lambda I_{n_{\mathrm{sv}}} + A_{\mathrm{sv}} D_{\mathrm{sv}} A_{\mathrm{sv}}^{\top} > 0$

$$-(\lambda K + KDK)^{-1} K\mathbf{g} = \begin{pmatrix} A_{\mathrm{sv}}^{-1} B^{-1} A_{\mathrm{sv}} \left(\mathbf{g}_{\mathrm{sv}} - \frac{1}{\lambda} D_{\mathrm{sv}} K_{\mathrm{sv,nsv}} \mathbf{g}_{\mathrm{nsv}}\right) \\ \frac{1}{\lambda}\mathbf{g}_{\mathrm{nsv}} \end{pmatrix}$$

n_{sv}: the number of support vectors

A_{sv}: the Cholesky decomposition of K_{sv}

K_{sv}: the matrix formed using the first n_{sv} rows and columns of K

5. Conclusion

Conclusion

- Change loss function
  - Logistic loss
  - Cross-entropy loss
- Select techniques
  - Convex Relaxation
  - Newton S^3VM
  - Continuation S^3VM

References

- Chapelle, Olivier, Vikas Sindhwani, and Sathiya S. Keerthi. "Optimization techniques for semi-supervised support vector machines." Journal of Machine Learning Research 9.Feb (2008): 203-233.
- Ueda, Naonori, and Ryohei Nakano. "Deterministic annealing EM algorithm." Neural networks 11.2 (1998): 271-282.
- Yuille, Alan L., and Anand Rangarajan. "The concave-convex procedure (CCCP)." Advances in neural information processing systems. 2002.