ML Session 3: Support Vector Machines: Binary class, 1-class, multi-class, structured SVMs

Javen Shi

6 Dec. 2018

Table of Contents I

Primal

A more popular version is (still a primal form)

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\| \mathbf{w} \|^2 + C\sum_{i=1}^{n} \xi_i,$$

$$\text{s.t.} \quad y_i(\langle \mathbf{x}_i, \mathbf{w} \rangle + b) \geq 1 - \xi_i, \xi_i \geq 0, i = 1, \cdots, n,$$

This is equivalent to the previous form and $\gamma = 1/\| \mathbf{w} \|$.

View in in ERM hinge loss $\ell_H(\mathbf{x}, y, \mathbf{w}) = \max\{0, 1 - y(\langle \mathbf{x}, \mathbf{w} \rangle + b)\}$, and $\Omega(\mathbf{w}) = \frac{1}{2}\| \mathbf{w} \|^2$ with a proper λ.

It is often solved by using Lagrange multipliers and duality.

Lagrangian function

$$L(\mathbf{w}, b, \xi, \alpha, \beta) = \frac{1}{2}\| \mathbf{w} \|^2 + C\sum_{i=1}^{n}\xi_i + \sum_{i=1}^{n}\alpha_i[1 - \xi_i - y_i(\langle \mathbf{x}_i, \mathbf{w}\rangle + b)] + \sum_{i=1}^{n}\beta_i(-\xi_i)$$

Optimise Lagrangian function — 1st order condition

To get $\inf_{\mathbf{w},b,\xi}\{L(\mathbf{w},b,\xi,\alpha,\beta)\}$, by 1st order condition

$$\frac{\partial L(\mathbf{w},b,\xi,\alpha,\beta)}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w}^* - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = 0 \tag{1}$$

$$\frac{\partial L(\mathbf{w},b,\xi,\alpha,\beta)}{\partial \xi_i} = 0 \Rightarrow C - \alpha_i - \beta_i = 0 \tag{2}$$

$$\frac{\partial L(\mathbf{w},b,\xi,\alpha,\beta)}{\partial b} = 0 \Rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{3}$$

Optimise Lagrangian function — Complementarity conditions

Complementarity conditions

$$\alpha_i[1 - \xi_i - y_i(\langle \mathbf{x}_i, \mathbf{w} \rangle + b)] = 0, \forall i \tag{4}$$

$$\beta_i \xi_i = 0, \forall i \tag{5}$$

Dual

$$
\begin{aligned}
&L(\mathbf{w}^*, b^*, \xi^*, \alpha, \beta) \\
&= \frac{1}{2}\langle \mathbf{w}^*, \mathbf{w}^* \rangle + \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \alpha_i y_i \langle \mathbf{x}_i, \mathbf{w}^* \rangle \\
&+ \sum_{i=1}^{n} \xi_i^* (C - \alpha_i - \beta_i) + b(\sum_{i=1}^{n} \alpha_i y_i) \\
&= \frac{1}{2}\langle \mathbf{w}^*, \mathbf{w}^* \rangle + \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \alpha_i y_i \langle \mathbf{x}_i, \mathbf{w}^* \rangle \quad \text{via eq(2) and eq(3)} \\
&= \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle + \sum_{i=1}^{n} \alpha_i - \sum_{i,j} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \;\; \text{via eq(1)} \\
&= \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle
\end{aligned}
$$

Dual

$\max_\alpha \inf_{\mathbf{w},b,\xi}\{L(\mathbf{w},b,\xi,\alpha,\beta)\}$ gives the dual form:

$$\max_\alpha \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j\rangle$$

$$\text{s.t.} \quad 0 \le \alpha_i \le C, i = 1, \cdots, n, \quad \text{(via eq(2))}$$

$$\sum_{i=1}^n \alpha_i y_i = 0$$

Let α^* be the solution.

From dual to primal variables

How to compute $\mathbf{w}^*, b^*$ from α^*?
Via eq(1), we have

$$\mathbf{w}^* = \sum_{i=1}^{n} \alpha_i^* y_i \mathbf{x}_i. \tag{6}$$

Via comp condition eq(4), we have $\alpha_i[1 - \xi_i - y_i(\langle \mathbf{x}_i, \mathbf{w} \rangle + b)] = 0, \forall i$. When $\alpha_i > 0$, we know $1 - \xi_i - y_i(\langle \mathbf{x}_i, \mathbf{w} \rangle + b) = 0$. It will be great if $\xi_i = 0$ too. When will it happen? $\beta_i > 0 \Rightarrow \xi_i = 0$ because of comp condition eq(5). Since $C - \alpha_i - \beta_i = 0$ (2), $\beta_i > 0$ means $\alpha < C$. For any i, s.t. $0 < \alpha_i < C$, $1 - y_i(\langle \mathbf{x}_i, \mathbf{w} \rangle + b) = 0$, so (multiple y_i on both sides, and the fact that $y_i^2 = 1$)

$$b^* = y_i - \langle \mathbf{x}_i, \mathbf{w}^* \rangle \tag{7}$$

Numerically wiser to take the average over all such training points (Burges tutorial).

Support Vectors

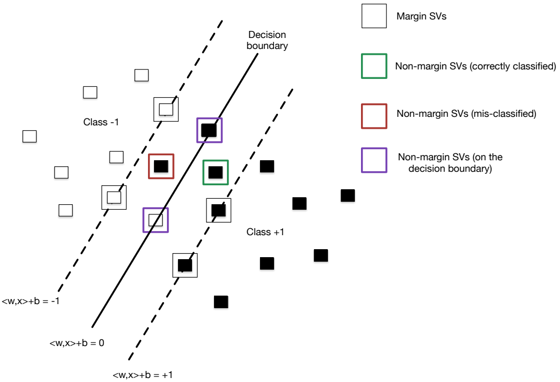
$y^* = \text{sign}(\langle \mathbf{x}, \mathbf{w}^* \rangle + b^*) = \text{sign}(\sum_{i=1}^{n} \alpha_i^* y_i \langle \mathbf{x}_i, \mathbf{x} \rangle + b^*)$.

It turns out many $\alpha_i^* = 0$. Those $\mathbf{x}_j$ with $\alpha_j^* > 0$ are called support vectors. Let $S = \{j : \alpha_j^* > 0\}$

$$y^* = \text{sign}(\sum_{j \in S} \alpha_j^* y_j \langle \mathbf{x}_j, \mathbf{x} \rangle + b^*)$$

Note now y can be predicted without explicitly expressing $\mathbf{w}$ as long as the support vectors are stored.

Support Vectors



Two types of SVs:

- Margin SVs: $0 < \alpha_i < C$ ($\xi_i = 0$, on the dash lines)
- Non-margin SVs: $\alpha_i = C$ ($\xi_i > 0$, thus violating the margin. More specifically, when $1 > \xi_i > 0$, correctly classified; when $\xi_i > 1$, it's mis-classified; when $\xi_i = 1$, on the decision boundary)

Dual

All derivation holds if one replaces $\mathbf{x}_j$ with $\phi(\mathbf{x}_j)$ and let kernel function $\kappa(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle$. This gives

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \kappa(\mathbf{x}_i, \mathbf{x}_j)$$

$$\text{s.t.} \quad 0 \leq \alpha_i \leq C, i = 1, \cdots, n$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

$$y^* = \text{sign}[\sum_{j \in S} \alpha_j^* y_j \kappa(\mathbf{x}_j, \mathbf{x}) + b^*].$$

This leads to non-linear SVM and more generally kernel methods (will be covered in later lectures).

Theoretical justification

An example of generalisation bounds is below (just to give you an intuition, no need to fully understand it for now).

Theorem (VC bound)

Denote h as the VC dimension, for all $n \geq h$, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$, $\forall g \in \mathcal{G}$

$$R(g) \leq R_n(g) + 2\sqrt{2\frac{h \log \frac{2en}{h} + \log(\frac{2}{\delta})}{n}}.$$

Margin $\gamma = 1/\| \mathbf{w} \|$, $h \leq \min\{D, \lceil \frac{4R^2}{\gamma^2} \rceil\}$, where the radius $R^2 = \max_{i=1}^n \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_i) \rangle$ (assuming data are already centered)

Theoretical justification

Other tighter bounds such as Rademacher bounds, PAC-Bayes bounds *etc.*.

Novelty detection

Motivation: data from one class are easy to collect, and data from the rest class(es) are hard (or disastrous) to collect, or too few to be statistical meaningful.

Novelty detection

Motivation: data from one class are easy to collect, and data from the rest class(es) are hard (or disastrous) to collect, or too few to be statistical meaningful.

Example:

- Operational status of a nuclear plant as "normal"

Novelty detection

Motivation: data from one class are easy to collect, and data from the rest class(es) are hard (or disastrous) to collect, or too few to be statistical meaningful.

Example:

- Operational status of a nuclear plant as "normal"
- Seeing a baby elephant $\Rightarrow$ elephants are small?

Novelty detection

- Only "normal data" in your training dataset (thus seen all as 1-class).

Novelty detection

- Only "normal data" in your training dataset (thus seen all as 1-class).
- for a testing data point, to predict if it's "normal" (*i.e.* belong to that class or not).

Novelty detection

Q: Since belonging to one class or not, why not a binary classification problem?

Novelty detection

Q: Since belonging to one class or not, why not a binary classification problem?

A: In novelty detection there are no "abnormal" data (*i.e.* 2nd class data) in the training dataset for you to train on.

Novelty detection

Q: Since belonging to one class or not, why not a binary classification problem?

A: In novelty detection there are no "abnormal" data (*i.e.* 2nd class data) in the training dataset for you to train on.

Other names: one-class classification, unary classification, outlier detection, anomaly detection

1-Class SVM

- One-Class SVM by Scholkopf etal (NIPS 2000)
- One-Class SVM by Tax and Duin (J.ML 2004)

One-Class SVM by Scholkopf etal (a.k.a. ν-SVM)

Support Vector Method for Novelty Detection

Bernhard Schölkopf*, Robert Williamson§,
Alex Smola§, John Shawe-Taylor†, John Platt*

* Microsoft Research Ltd., 1 Guildhall Street, Cambridge, UK
§ Department of Engineering, Australian National University, Canberra 0200
† Royal Holloway, University of London, Egham, UK
* Microsoft, 1 Microsoft Way, Redmond, WA, USA
bsc/jplatt@microsoft.com, Bob.Williamson/Alex.Smola@anu.edu.au, john@dcs.rhbnc.ac.uk

Primal

$$\min_{w \in F, \boldsymbol{\xi} \in \mathbb{R}^{\ell}, \rho \in \mathbb{R}} \quad \frac{1}{2}\|w\|^2 + \frac{1}{\nu\ell}\sum_i \xi_i - \rho$$

$$\text{subject to} \quad (w \cdot \Phi(\mathbf{x}_i)) \geq \rho - \xi_i, \;\; \xi_i \geq 0.$$

ℓ is the number of training examples. ν is a hyper-parameter (often chosen by human).

- ν is an upper bound on the fraction of outliers
- ν is a lower bound on the number of training examples used as Support Vector

Also known as ν-SVM.

Dual

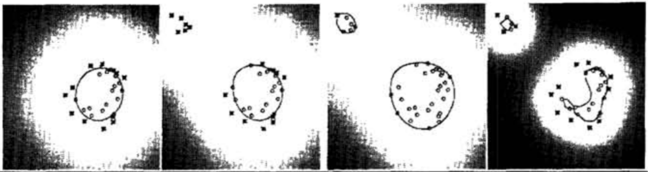
$$\min_{\boldsymbol{\alpha}} \frac{1}{2} \sum_{ij} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) \quad \text{subject to } 0 \leq \alpha_i \leq \frac{1}{\nu \ell}, \quad \sum_i \alpha_i = 1.$$

Decision function (predication)

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_i \alpha_i k(\mathbf{x}_i, \mathbf{x}) - \rho\right)$$

The offset ρ can be recovered by exploiting that for any α_i which is not at the upper or lower bound, the corresponding pattern $\mathbf{x}_i$ satisfies $\rho = (w \cdot \Phi(\mathbf{x}_i)) = \sum_j \alpha_j k(\mathbf{x}_j, \mathbf{x}_i)$.

Toy results



ν, width c	0.5, 0.5	0.5, 0.5	0.1, 0.5	0.5, 0.1
frac. SVs/OLs	0.54, 0.43	0.59, 0.47	0.24, 0.03	0.65, 0.38
margin $\rho/\|w\|$	0.84	0.70	0.62	0.48

Figure 1: *First two pictures:* A single-class SVM applied to two toy problems; $\nu = c = 0.5$, domain: $[-1, 1]^2$. Note how in both cases, at least a fraction of ν of all examples is in the estimated region (cf. table). The large value of ν causes the additional data points in the upper left corner to have almost no influence on the decision function. For smaller values of ν, such as 0.1 *(third picture)*, the points cannot be ignored anymore. Alternatively, one can force the algorithm to take these 'outliers' into account by changing the kernel width (2): in the *fourth picture*, using $c = 0.1, \nu = 0.5$, the data is effectively analyzed on a different length scale which leads the algorithm to consider the outliers as meaningful points.

Primal

$$\min_{R,\,\mathbf{a}} R^2 + C\sum_{i=1}^{n}\xi_i$$

subject to:

$$\|x_i - \mathbf{a}\|^2 \leq R^2 + \xi_i \qquad \text{for all } i = 1, \dots, n$$

$$\xi_i \geq 0 \qquad \text{for all } i = 1, \dots, n$$

Dual

$$L = \sum_i \alpha_i(\mathbf{x}_i \cdot \mathbf{x}_i) - \sum_{i,j} \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j)$$

$$0 \leq \alpha_i \leq C$$

Decision function

To test an object $\mathbf{z}$, the distance to the center of the sphere has to be calculated. A test object $\mathbf{z}$ is accepted when this distance is smaller or equal than the radius:

$$\|\mathbf{z}-\mathbf{a}\|^2 = (\mathbf{z}\cdot\mathbf{z}) - 2\sum_i \alpha_i(\mathbf{z}\cdot\mathbf{x}_i) + \sum_{i,j}\alpha_i\alpha_j(\mathbf{x}_i\cdot\mathbf{x}_j) \leq R^2 \quad (14)$$

Multi-class SVM

$$\min_{\mathbf{w},\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i \tag{8a}$$

$$\text{s.t.} \quad \forall i, y, \quad \langle \mathbf{w}, \Phi(\mathbf{x}_i, y_i) - \Phi(\mathbf{x}_i, y)\rangle \geq 1 - \xi_i. \tag{8b}$$

Using whiteboard for derivation.

SVM-struct

In order to allow some outliers, they use slack variables ξ_i and maximise the minimum margin, $F(\mathbf{x}_i, \mathbf{y}_i) - \max_{\mathbf{y} \in \mathcal{Y} - \mathbf{y}_i} F(\mathbf{x}_i, \mathbf{y})$, across training instances i. Equivalently,

$$\min_{\mathbf{w}, \xi} \ \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{m} \xi_i \tag{9a}$$

$$\text{s.t.} \quad \forall i, \mathbf{y}, \quad \langle \mathbf{w}, \Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \mathbf{y}) \rangle \geq \Delta(\mathbf{y}_i, \mathbf{y}) - \xi_i, \xi_i \geq 0. \tag{9b}$$

SVM-struct

In order to allow some outliers, they use slack variables ξ_i and maximise the minimum margin, $F(\mathbf{x}_i, \mathbf{y}_i) - \max_{\mathbf{y} \in \mathcal{Y} - \mathbf{y}_i} F(\mathbf{x}_i, \mathbf{y})$, across training instances i. Equivalently,

$$\min_{\mathbf{w}, \xi} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{m} \xi_i \tag{9a}$$

$$\text{s.t.} \quad \forall i, \mathbf{y}, \quad \langle \mathbf{w}, \Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \mathbf{y}) \rangle \geq \Delta(\mathbf{y}_i, \mathbf{y}) - \xi_i, \xi_i \geq 0. \tag{9b}$$

- How many constraints here for each i?

SVM-struct

In order to allow some outliers, they use slack variables ξ_i and maximise the minimum margin, $F(\mathbf{x}_i, \mathbf{y}_i) - \max_{\mathbf{y} \in \mathcal{Y} - \mathbf{y}_i} F(\mathbf{x}_i, \mathbf{y})$, across training instances i. Equivalently,

$$\min_{\mathbf{w}, \xi} \ \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{m} \xi_i \tag{9a}$$

$$\text{s.t.} \quad \forall i, \mathbf{y}, \quad \langle \mathbf{w}, \Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \mathbf{y}) \rangle \geq \Delta(\mathbf{y}_i, \mathbf{y}) - \xi_i, \xi_i \geq 0. \tag{9b}$$

- How many constraints here for each i?
- Reduce to only one constraint per i — finding the most violating constraint (a MAP inference problem).

Using whiteboard for derivation.

That's all

Thanks!