Data Science 2 - SVM

March 5, 2022

Data Science 2 - SVM

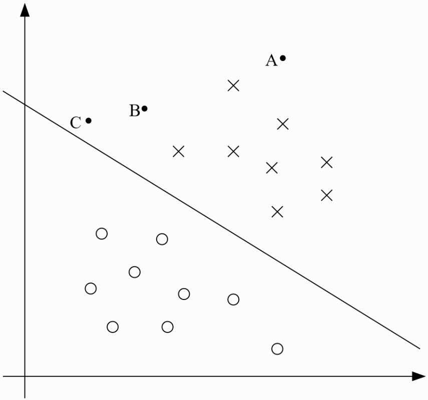
Faculty of Mathematics and Physics

Support Vector Machine
Intro

- Consider logistic regression, where the probability of event happening is modeled by $h(x) = g(\theta^\top x)$.
- We predict 1 on an input x if and only if $h(x) \geq 0.5$, or equivalently, if and only if $g(\theta^\top x) \geq 0$
- Consider a positive training example $y = 1$. The larger $\theta^\top x$ is, the higher our degree of "confidence" that the label is 1.
- We can think of our prediction as being very confident that $y = 1$ if $\theta^\top x \gg 0$. Similarly, predicting confidently $y = 0$ if $\theta^\top x \ll 0$.
- Given a training set, a good fit to the training data means we can find θ so that $\theta^\top x_i \gg 0$ whenever $y_i = 1$, and $\theta^\top x_i \ll 0$ whenever $y_i = 0$

Support Vector Machine
Separating hyperplane

Consider the classification using a separating hyperplane of $\theta^\top x = 0$:



- Point A is far from the decision boundary, point C is close to the decision boundary
- Small change to the decision boundary could cause prediction to be $y = 0$ for point C

Support Vector Machine
Notation

- Consider a linear classifier for a binary classification problem with labels y and features x.
- Let's use $y \in \{-1, 1\}$ (instead of $\{0, 1\}$) to denote the class labels.
- We will use parameters w, b for weights and intercept:

$$h_{w,b}(x) = g(w^\top x + b)$$

- Define $g(z) = 1$ if $z \geq 0$ and $g(z) = -1$ otherwise.
- Our classifier will directly predict either 1 or -1 without intermediate step of estimating probability

Support Vector Machine
Functional margin

Given observation (x_i, y_i), we define the functional margin of (w, b) with respect to the observation as:

$$\gamma_i^f = y_i(w^\top x_i + b)$$

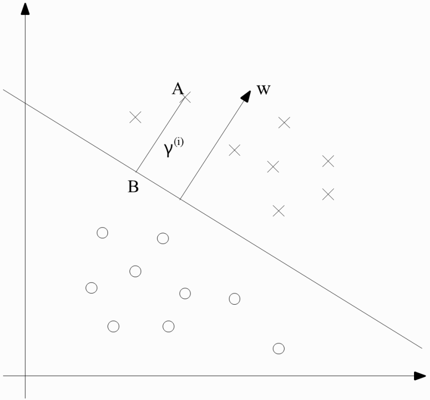
- For $y_i = 1$ the margin will be large if $(w^\top x_i + b)$ is large positive number, and, conversely, for $y_i = -1$ we need to have large negative number to get the margin high
- Not robust w.r.t. to scaling if we use the function g defined above:
  - if we replace w with $2w$ and b with $2b$, the sign of prediction remains the same, but the margin changes
  - given the freedom of scale, this can be solved by normalization of the weights

Given a training set $S = (x_i, y_i),\ i = 1, \ldots, n$, we also define the functional margin of (w, b) with respect to S as the smallest of the functional margins of the individual examples:

$$\gamma^f = \min_{i=1,\ldots,n} \gamma_i^f$$

Support Vector Machine
Geometric margin

Consider the classification using a separating hyperplane given by (w, b):



- w is orthogonal to the separating hyperplane
- geometric margin should be the distance to the point A denoted by γ_i

Support Vector Machine

Geometric margin

The distance from B to A is given by:

$$\gamma_i \frac{w}{\|w\|}$$

Since B is on the separating hyperplane, we have:

$$w^\top \left(x_i - \gamma_i \frac{w}{\|w\|} \right) + b = 0$$

Solving for *w* yields:

$$\gamma_i = \left(\frac{w}{\|w\|} \right)^\top x_i + \frac{b}{\|w\|}$$

For the other direction, only the sign changes, therefore, generally:

$$\gamma_i = y_i \left(\left(\frac{w}{\|w\|} \right)^\top x_i + \frac{b}{\|w\|} \right)$$

Support Vector Machine
Geometric margin

- For $\|w\| = 1$ geometric margin is equivalent to functional margin
- Geometric margin is invariant to scaling of the parameters

Given a training set $S = (x_i, y_i),\ i = 1, \ldots, n$, we also define the geometric margin of (w, b) with respect to S as the smallest of the geometric margins of the individual training examples:

$$\gamma = \min_{i=1,\ldots,n} \gamma_i$$

Support Vector Machine
Optimal margin classifier

- Let's find a decision boundary that maximizes the (geometric) margin
- This shall result in a classifier that separates the positive and the negative training examples with a gap
- Assume that we are given a training set that is linearly separable: it is possible to separate the positive and negative examples using some separating hyperplane

$$\begin{aligned} &\max_{\gamma,w,b} \gamma \\ &\text{s.t. } y_i(w^\top x_i + b) \geq \gamma, \quad i = 1, \ldots, n \\ &\quad \|w\| = 1 \end{aligned} \tag{1}$$

We should try to reformulate to get rid of the non-convex constraint $\|w\| = 1$

Support Vector Machine
Optimal margin classifier

- Plug in $\gamma = \frac{\gamma^f}{\|w\|}$:

$$
\begin{aligned}
&\max_{\gamma^f, w, b} \frac{\gamma^f}{\|w\|} \\
&\text{s.t. } y_i(w^\top x_i + b) \geq \gamma^f, \ i = 1, \dots, n
\end{aligned} \tag{2}
$$

We still have a non-convex objective function.

Support Vector Machine
Optimal margin classifier

- We know that we can have arbitrary scaling of the parameters without changing the classification
- Consider setting the margin to $\gamma^f = 1$
- Now maximizing $\frac{1}{||w||}$ is the same as minimizing $||w||^2$

$$\begin{aligned} &\min_{w,b} \frac{1}{2}||w||^2 \\ &\text{s.t. } y_i(w^\top x_i + b) \geq 1, \;\; i = 1, \dots, n \end{aligned} \tag{3}$$

This is a quadratic program (quadratic objective and linear constraints) solvable by starndard optimization solvers.

Support Vector Machine
Generalized Lagrangian

Consider primal problem

$$\begin{aligned} &\min_{w} f(w) \\ \text{s.t. } & g_i(w) \leq 0, \;\; i = 1, \dots, k \\ & h_i(w) = 0, \;\; i = 1, \dots, l \end{aligned} \tag{4}$$

Generalized Lagrangian is given by:

$$\mathcal{L}(w, \alpha, \beta) = f(w) + \sum_{i=1}^{k} \alpha_i g_i(w) + \sum_{i=1}^{l} \beta_i h_i(w)$$

Dual solution d^* is always lower than primal solution p^*:

$$d^* = \max_{\alpha, \beta : \alpha_i \geq 0} \min_{w} \mathcal{L}(w, \alpha, \beta) \leq \min_{w} \max_{\alpha, \beta : \alpha_i \geq 0} \mathcal{L}(w, \alpha, \beta) = p^*$$

Support Vector Machine
KKT conditions

Suppose that objective function f and nonequality constraints g_i are convex as well as equality constraints h_i are affine. Suppose further that the constraints g_i are (strictly) feasible; this means that there exists some w so that $g_i(w) < 0$ for all i.
Then there must exist w^*, α^*, β^* so that w^* is the solution to the primal problem, α^*, β^* are the solution to the dual problem, and the optimal solutions are the same. If some w^*, α^*, β^* satisfy KKT conditions, they are a solution to primal and dual problem:

$$
\begin{aligned}
\frac{\partial}{\partial w_i}\mathcal{L}(w^*, \alpha^*, \beta^*) &= 0, \;\; i = 1, \ldots, d \\
\frac{\partial}{\partial \beta_i}\mathcal{L}(w^*, \alpha^*, \beta^*) &= 0, \;\; i = 1, \ldots, l \\
\alpha_i^* g_i(w^*) &= 0, \;\; i = 1, \ldots, k \\
g_i(w^*) &\le 0, \;\; i = 1, \ldots, k \\
\alpha_i^* &\ge 0, \;\; i = 1, \ldots, k
\end{aligned} \tag{5}
$$

Support Vector Machine
Lagrangian

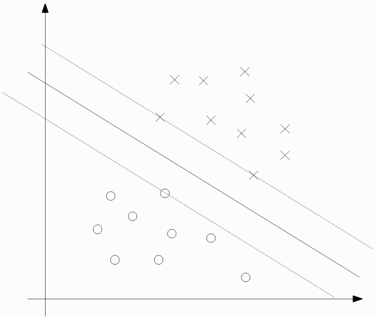
- We have constraints:

$$g_i = -y_i(w^\top x_i + b) + 1 \leq 0$$

- Lagrangian is then given by:

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i \left(y_i(w^\top x_i + b) - 1\right)$$

- From KKT conditions, we know that corresponding multipliers $\alpha_i > 0$ only if $g_i = 0$
- The corresponding points are the *supporting vectors*



Support Vector Machine

Dual Problem

Let's use Lagrange duality to obtain the dual problem, minimize the Lagrangian w.r.t. *w* and *b*:

$$\begin{aligned}
\frac{\partial}{\partial w}\mathcal{L}(w,b,\alpha) &= w - \sum_{i=1}^{n} \alpha_i y_i^\top x_i = 0 \\
w &= \sum_{i=1}^{n} \alpha_i y_i^\top x_i \\
\frac{\partial}{\partial b}\mathcal{L}(w,b,\alpha) &= \sum_{i=1}^{n} \alpha_i y_i = 0 \\
\mathcal{L}(w,b,\alpha) &= -\frac{1}{2}\sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j x_i^\top x_j + \sum_{i=1}^{n}\alpha_i - b\sum_{i=1}^{n}\alpha_i y_i \\
\mathcal{L}(w,b,\alpha) &= \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} y_i y_k \alpha_j \alpha_j x_i^\top x_j
\end{aligned} \tag{6}$$

Support Vector Machine
Dual problem

Our dual problem is then given by:

$$
\begin{aligned}
\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_k \alpha_j \alpha_j \langle x_i, x_j \rangle \\
\text{s.t.} \;\; \alpha_i \geq 0, \;\; i = 1, \dots, n \\
\sum_{i=1}^{n} \alpha_i y_i = 0
\end{aligned}
\tag{7}
$$

- KKT conditions are satisfied so we can solve the dual problem instead of primal problem
- once we have final weights α^*, we get the other parameters as:

$$
\begin{aligned}
w^* &= \sum_{i=1}^{n} \alpha_i^* y_i^\top x_i \\
b^* &= -\frac{\max_{i:y_i=-1} (w^*)^\top x_i + \min_{i:y_i=1} (w^*)^\top x_i}{2}
\end{aligned}
\tag{8}
$$

Support Vector Machine
Dual problem

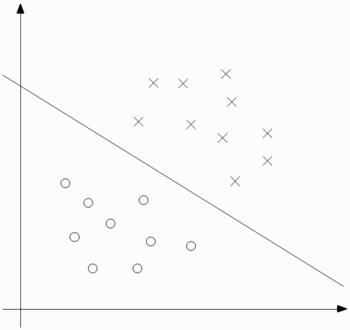
To apply fitted SVM for a new prediction, we calculate:

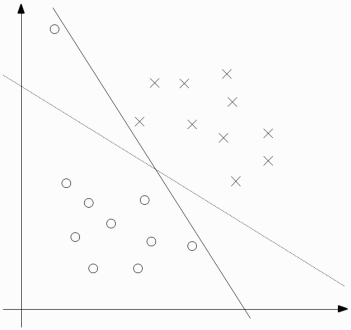
$$
\begin{aligned}
w^\top x + b &= \left(\sum_{i=1}^{n} \alpha_i^* y_i^\top x_i \right)^\top x + b \\
&= \sum_{i=1}^{n} \alpha_i^* y_i \langle x_i, x \rangle + b
\end{aligned}
\tag{9}
$$

- Since most of α_i are zero, we have to calculate only few items of the sum (only on supporting vectors)
- Kernel functions can be applied instead of the inner product to efficiently increase the feature space and make the separation possible
  - Polynomial kernels $k(x_i, x_j) = \langle x_i, x_j \rangle^d$
  - Gaussian kernels $k(x_i, x_j) = \exp\{-\gamma \|x_i, x_j\|^2\}$
  - Hyperbolic tangent $k(x_i, x_j) = \tanh\{-\kappa \langle x_i, x_j \rangle + c\}$
- Optimization problem can be solved by interior point methods or by specialized algoritgm (SMO = sequential minimal optimization)

Support Vector Machine
Regularization

Separating hyperplanes can be very sensitive to outliers:





Support Vector Machine

Regularization

- We add *L*1 regularization which will make our algorithm more robust
- This will allow to cover the case when observations are linearly non-separable
- Hyperparameter *C* to control the weight of regularization

$$
\begin{aligned}
&\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i \\
&\text{s.t. } y_i(w^\top x_i + b) \geq 1 - \xi_i, \;\; i = 1, \ldots, n \\
&\qquad \xi_i \geq 0, \;\; i = 1, \ldots, n
\end{aligned}
\tag{10}
$$

Support Vector Machine
SMO algorithm

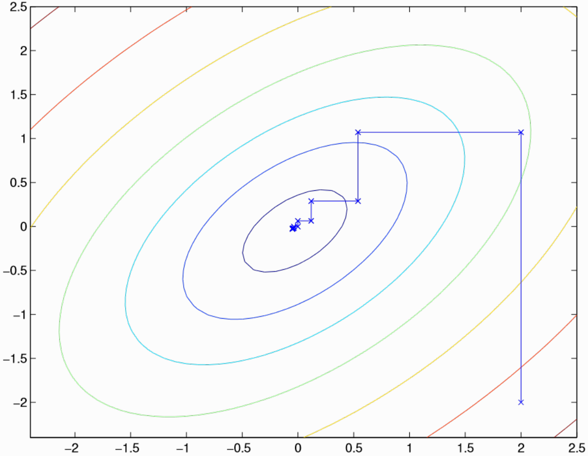
Our dual problem with regularization is given by:

$$\begin{aligned} \max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_k \alpha_j \alpha_j \langle x_i, x_j \rangle \\ \text{s.t. } 0 \leq \alpha_i \leq C, \;\; i = 1, \ldots, n \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{aligned} \tag{11}$$

- Denote $W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_k \alpha_j \alpha_j \langle x_i, x_j \rangle$
- Coordinate ascent algorithms pick one parameter at a time (α_i) and maximize the function $W(\alpha)$ with other parameters fixed
- We repeat the process for all coordinates until convergence criterion is satisfied

Support Vector Machine
Coordinate ascent

Coordinate ascent algorithm example:



Support Vector Machine

SMO algorithm

- We cannot chance single α_i because of the constraint $\sum_{i=1}^{n} \alpha_i y_i = 0$

$$\alpha_j y_j = -\sum_{i \neq j} \alpha_i y_i$$

$$\alpha_j = -y_j \sum_{i \neq j} \alpha_i y_i$$

- Repeat until KKT conditions are satisfied with some tolerance:
  1. Select a pair α_i and α_j to update next (using a heuristic)
  2. Reoptimize $W(\alpha)$ with respect to α_i and α_j , while all the other α_k, $k \notin \{i, j\}$ are fixed.
- Since α_j can be written as a linear function of α_i, we can put them to the objective and find out that we have quadratic function
- Finding new values is then very fast, maximizing of quadratic function which could be clipped at some boundaries to ensure both α_i and α_j are greater than zero and lower than C

Support Vector Machine
Usage

- Text and hypertext categorization
- Classification of images
- Hand-written characters recognition
- Speech recognition
- Outlier detection

Multiclass problems are usually reduced into multiple binary classification problems.

Thank you!

TARAN

ADVISORY IN DATA & ANALYTICS