Machine Learning

Support Vector Machines

Fabio Vandin

November 20, 2017
Classification and Margin

Consider a classification problem with two classes:
- instance set $\mathcal{X} = \mathbb{R}^d$
- label set $\mathcal{Y} = \{-1, 1\}$.

Training data: $S = (((x_1, y_1), \ldots, (x_m, y_m))$

Hypothesis set $\mathcal{H} = \text{halfspaces}$

**Assumption:** data is linearly separable $\Rightarrow$ there exist a halfspace that perfectly classify the training set

**In general:** multiple separating hyperplanes: $\Rightarrow$ which one is the best choice?

![Diagram showing classification and margin concepts](image-url)
Classification and Margin

The last one seems the best choice, since it can tolerate more “noise”.

Informally, for a given separating halfspace we define its margin as its minimum distance to an example in the training set $S$.

**Intuition:** best separating hyperplane is the one with largest margin.

How do we find it?
Training set $S = ((x_1, y_1), \ldots, (x_m, y_m))$ is *linearly separable* if there exists a halfspace $(w, b)$ such that $y_i = \text{sign}(\langle w, x_i \rangle + b)$ for all $i = 1, \ldots, m$.

Equivalent to:

$$\forall i = 1, \ldots, m : y_i(\langle w, x_i \rangle + b) > 0$$

**Informally**: *margin* of a separating hyperplane is its minimum distance to an example in the training set $S$.
Given hyperplane defined by \( L = \{ \mathbf{v} : \langle \mathbf{w}, \mathbf{v} \rangle + b = 0 \} \), and given \( \mathbf{x} \), the distance of \( \mathbf{x} \) to \( L \) is

\[
d(\mathbf{x}, L) = \min\{||\mathbf{x} - \mathbf{v}|| : \mathbf{v} \in L\}
\]

Claim: if \( ||\mathbf{w}|| = 1 \) then \( d(\mathbf{x}, L) = |\langle \mathbf{w}, \mathbf{x} \rangle + b| \) (Proof: Claim 15.1 [UML])
Margin and Support Vectors

The *margin* of a separating hyperplane is the distance of the closest example in training set to it. If $||w|| = 1$ the margin is:

$$\min_{i \in \{1, \ldots, m\}} |\langle w, x_i \rangle + b|$$

The closest examples are called *support vectors*.
Support Vector Machine (SVM)

**Hard-SVM**: seek for the separating hyperplane with largest margin (only for linearly separable data)

**Computational problem:**

\[
\arg \max_{(w,b): \|w\|=1} \min_{i \in \{1, \ldots, m\}} \left| \langle w, x_i \rangle + b \right|
\]

subject to \( \forall i : y_i (\langle w, x_i \rangle + b) > 0 \)

**Equivalent formulation** (due to separability assumption):

\[
\arg \max_{(w,b): \|w\|=1} \min_{i \in \{1, \ldots, m\}} y_i (\langle w, x_i \rangle + b)
\]
Hard-SVM: Quadratic Programming Formulation

- **input**: \((x_1, y_1), \ldots, (x_m, y_m)\)
- **solve**: 
  \[
  (w_0, b_0) = \arg \min_{(w,b)} ||w||^2
  \]
  subject to \(\forall i : y_i(\langle w, x_i \rangle + b) \geq 1\)
- **output**: \(\hat{w} = \frac{w_0}{||w_0||}, \hat{b} = \frac{b_0}{||w_0||}\)

**How do we get a solution?** Quadratic optimization problem: objective is convex quadratic function, constraints are linear inequalities ⇒ Quadratic Programming solvers!

**Proposition**
The output of algorithm above is a solution to the *Equivalent Formulation* in the previous slide.

Proof: on the board and in the book (Lemma 15.2 [UML])
Equivalent Formulation and Support Vectors

Equivalent formulation (homogeneous halfspaces): assume first component of $\mathbf{x} \in \mathcal{X}$ is 1, then

$$\mathbf{w}_0 = \min_{\mathbf{w}} ||\mathbf{w}||^2 \text{ subject to } \forall i : y_i \langle \mathbf{w}, \mathbf{x}_i \rangle \geq 1$$

“Support Vectors” = vectors at minimum distance from $\mathbf{w}_0$

The support vectors are the only ones that matter for defining $\mathbf{w}_0$!

**Proposition**

Let $\mathbf{w}_0$ be as above. Let $I = \{ i : |\langle \mathbf{w}_0, \mathbf{x} \rangle| = 1 \}$. Then there exist coefficients $\alpha_1, \ldots, \alpha_m$ such that

$$\mathbf{w}_0 = \sum_{i \in I} \alpha_i \mathbf{x}_i$$

“Support vectors” = $\{ \mathbf{x}_i : i \in I \}$

**Note:** Solving Hard-SVM is equivalent to find $\alpha_i$ for $i = 1, \ldots, m$, and $\alpha_i \neq 0$ only for support vectors
Soft-SVM

Hard-SVM works if data is linearly separable.

What if data is not linearly separable? ⇒ soft-SVM

**Idea**: modify constraints of Hard-SVM to allow for some violation, but take into account violations into objective function
Soft-SVM Constraints

Hard-SVM constraints:

\[ y_i(\langle w, x_i \rangle + b) \geq 1 \]

Soft-SVM constraints:

- slack variables: \( \xi_1, \ldots, \xi_m \geq 0 \Rightarrow \text{vector } \xi \)
- for each \( i = 1, \ldots, m \): \( y_i(\langle w, x_i \rangle + b) \geq 1 - \xi_i \)
- \( \xi_i \): how much constraint \( y_i(\langle w, x_i \rangle + b) \geq 1 \) is violated

Soft-SVM minimizes combinations of

- norm of \( w \)
- average of \( \xi_i \)

Tradeoff among two terms is controlled by a parameter \( \lambda \in \mathbb{R}, \lambda > 0 \)
Soft-SVM: Optimization Problem

- **input**: \((x_1, y_1), \ldots, (x_m, y_m)\), parameter \(\lambda > 0\)
- **solve**:

  \[
  \min_{w, b, \xi} \left( \lambda \|w\|^2 + \frac{1}{m} \sum_{i=1}^{m} \xi_i \right)
  \]

  subject to \(\forall i : y_i(\langle w, x_i \rangle + b) \geq 1 - \xi_i\) and \(\xi_i \geq 0\)

- **output**: \(w, b\)

**Equivalent formulation**: consider the *hinge loss*

\[
\ell_{\text{hinge}}((w, b), (x, y)) = \max\{0, 1 - y(\langle w, x \rangle + b)\}
\]

Given \((w, b)\) and a training \(S\), the empirical risk \(L_s^{\text{hinge}}((w, b))\) is

\[
L_s^{\text{hinge}}((w, b)) = \frac{1}{m} \sum_{i=1}^{m} \ell_{\text{hinge}}((w, b), (x_i, y_i))
\]
Soft-SVM as RLM

Soft-SVM: solve

$$\min_{w,b,\xi} \left( \lambda \|w\|^2 + \frac{1}{m} \sum_{i=1}^{m} \xi_i \right)$$

subject to $\forall i : y_i (\langle w, x_i \rangle + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$

Equivalent formulation with hinge loss:

$$\min_{w,b} \left( \lambda \|w\|^2 + L_{\text{hinge}}(w, b) \right)$$

that is

$$\min_{w,b} \left( \lambda \|w\|^2 + \frac{1}{m} \sum_{i=1}^{m} \ell_{\text{hinge}}((w, b), (x_i, y_i)) \right)$$

Note:

- $\lambda \|w\|^2$: $\ell_2$ regularization
- $L_{\text{hinge}}(w, b)$: empirical risk for hinge loss
Soft-SVM: Solution

We need to solve:

\[
\min_{w, b} \left( \lambda \|w\|^2 + \frac{1}{m} \sum_{i=1}^{m} \ell_{\text{hinge}}((w, b), (x_i, y_i)) \right)
\]

How?

- standard solvers for optimization problems
- **Stochastic Gradient Descent**
Gradient Descent

General approach for minimizing a differentiable convex function $f(w)$

Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$ be a differentiable function

Definition

The gradient $\nabla f(w)$ of $f$ at $w = (w_1, \ldots, w_d)$ is

$$\nabla f(w) = \left( \frac{\partial f(w)}{\partial w_1}, \ldots, \frac{\partial f(w)}{\partial w_d} \right)$$

Intuition: the gradient points in the direction of the greatest rate of increase of $f$ around $w$
Let $\eta \in \mathbb{R}, \eta > 0$ be a parameter.

Gradient Descent (GD) algorithm:

\[ w^{(0)} \leftarrow 0; \]
\[ \text{for } t \leftarrow 0 \text{ to } T - 1 \text{ do} \]
\[ w^{(t+1)} \leftarrow w^{(t)} - \eta \nabla f(w^{(t)}); \]
\[ \text{return } \bar{w} = \frac{1}{T} \sum_{t=1}^{T} w^{(t)}; \]

Notes:

- output vector could also be $w^{(T)}$ or $\arg \min_{t \in \{1, \ldots, T\}} f(w^{(t)})$
- returning $\bar{w}$ is useful for nondifferentiable functions (using subgradients instead of gradients...) and for stochastic gradient descent...

Guarantees: Let $f$ be convex and $\rho$-Lipschitz continuos; let $w^* \in \arg \min_{w: ||w|| \leq B} f(w)$. Then for every $\epsilon > 0$, to find $\bar{w}$ such that $f(\bar{w}) - f(w^*) \leq \epsilon$ it is sufficient to run the GD algorithm for $T \geq B^2 \rho^2 / \epsilon^2$ iterations.
Stochastic Gradient Descent (SGD)

Idea: instead of using exactly the gradient, we take a (random) vector with expected value equal to the gradient direction.

SGD algorithm:
\[ w^{(0)} \leftarrow 0; \]
\[
\text{for } t \leftarrow 1 \text{ to } T \text{ do }
\]
\[
\quad \text{choose } v_t \text{ at random from a distribution such that }
\quad E[v_t|w^{(t)}] \in \nabla f(w^{(t)});
\]
\[
\quad w^{(t+1)} \leftarrow w^{(t)} - \eta v_t;
\]

return \[ \bar{w} = \frac{1}{T} \sum_{t=1}^{T} w^{(t)}; \]
\textbf{Guarantees:} Let $f$ be convex and $\rho$-Lipschitz continuos; let $w^* \in \arg \min_{w: ||w|| \leq B} f(w)$. Then for every $\epsilon > 0$, to find $\bar{w}$ such that $E[f(\bar{w})] - f(w^*) \leq \epsilon$ it is sufficient to run the SGD algorithm for $T \geq B^2 \rho^2 / \epsilon^2$ iterations.
Why should we use SGD instead of GD?

**Question**: when do we use GD in the first place?

**Answer**: for example to find \( w \) that minimizes \( L_S(w) \)

That is \( f(w) = L_S(w) \)

\[ \nabla f(w) \text{ depends on all pairs } (x_i, y_i) \in S, i = 1, \ldots, m: \text{ may require long time to compute it!} \]

**What about SGD?**

We need to pick \( v_t \) such that \( E[v_t|w(t)] \in \nabla f(w(t)): \text{ how?} \)

Pick a random \( (x_i, y_i) \Rightarrow \text{ pick } v_t \in \nabla \ell(w(t), (x_i, y_i)): \)

- satisfies the requirement!
- requires much less computation than GD

Analogously we can use SGD for regularized losses, etc.
We want to solve

$$\min_{\mathbf{w}} \left( \frac{\lambda}{2} \| \mathbf{w} \|^2 + \frac{1}{m} \sum_{i=1}^{m} \max\{0, 1 - y \langle \mathbf{w}, \mathbf{x}_i \rangle\} \right)$$

**Note:** it's standard to add a $\frac{1}{2}$ in the regularization to simplify some computations.

Algorithm:

$\theta^{(1)} \leftarrow \mathbf{0}$;

for $t \leftarrow 1$ to $T$ do

    let $\mathbf{w}^{(t)} \leftarrow \frac{1}{\lambda t} \theta^{(t)}$;

    choose $i$ uniformly at random from $\{1, \ldots, m\}$;

    if $y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle < 1$ then $\theta^{(t+1)} \leftarrow \theta^{(t)} + y_i \mathbf{x}_i$;

    else $\theta^{(t+1)} \leftarrow \theta^{(t)}$;

return $\bar{\mathbf{w}} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{w}^{(t)}$;
Duality

We now present (Hard-)SVM in a different way which is very useful for kernels.

We want to solve

$$\mathbf{w}_0 = \min_{\mathbf{w}} \frac{1}{2} \| \mathbf{w} \|^2 \text{ subject to } \forall i : y_i \langle \mathbf{w}, \mathbf{x}_i \rangle \geq 1$$

Define

$$g(\mathbf{w}) = \max_{\alpha \in \mathbb{R}^m : \alpha \geq 0} \sum_{i=1}^{m} \alpha_i (1 - y_i \langle \mathbf{w}, \mathbf{x}_i \rangle)$$

Note that

$$g(\mathbf{w}) = \begin{cases} 0 & \text{if } \forall i : y_i \langle \mathbf{w}, \mathbf{x}_i \rangle \geq 1 \\ +\infty & \text{otherwise} \end{cases}$$

Then the problem above can be rewritten as:

$$\min_{\mathbf{w}} \left( \frac{1}{2} \| \mathbf{w} \|^2 + g(\mathbf{w}) \right)$$
\[
\min_w \left( \frac{1}{2} \|w\|^2 + g(w) \right)
\]

Rerranging a bit the terms we obtain the equivalent problem:

\[
\min_w \max_{\alpha \in \mathbb{R}^m: \alpha \geq 0} \left( \frac{1}{2} \|w\|^2 + \sum_{i=1}^{m} \alpha_i (1 - y_i \langle w, x_i \rangle) \right)
\]

One can prove that

\[
\min_w \max_{\alpha \in \mathbb{R}^m: \alpha \geq 0} \left( \frac{1}{2} \|w\|^2 + \sum_{i=1}^{m} \alpha_i (1 - y_i \langle w, x_i \rangle) \right) = \max_{\alpha \in \mathbb{R}^m: \alpha \geq 0} \min_w \left( \frac{1}{2} \|w\|^2 + \sum_{i=1}^{m} \alpha_i (1 - y_i \langle w, x_i \rangle) \right)
\]

Note: the result above is called strong duality
We therefore define the dual problem

\[
\max_{\alpha \in \mathbb{R}^m : \alpha \geq 0} \min_w \left( \frac{1}{2} \|w\|^2 + \sum_{i=1}^{m} \alpha_i (1 - y_i \langle w, x_i \rangle) \right)
\]

**Note:** once \( \alpha \) is fixed

- the optimization with respect to \( w \) is unconstrained
- the objective is differentiable

\( \Rightarrow \) at the optimum the gradient is equal to 0:

\[
w - \sum_{i=1}^{m} \alpha_i y_i x_i = 0 \Rightarrow w = \sum_{i=1}^{m} \alpha_i y_i x_i
\]

Replacing in the dual problem we get:

\[
\max_{\alpha \in \mathbb{R}^m : \alpha \geq 0} \left( \frac{1}{2} \left\| \sum_{i=1}^{m} \alpha_i y_i x_i \right\|^2 + \sum_{i=1}^{m} \alpha_i \left( 1 - y_i \left\langle \sum_{j=1}^{m} \alpha_j y_j x_j, x_i \right\rangle \right) \right)
\]
\[
\max_{\alpha \in \mathbb{R}^m : \alpha \geq 0} \left( \frac{1}{2} \left\| \sum_{i=1}^{m} \alpha_i y_i x_i \right\|^2 + \sum_{i=1}^{m} \alpha_i \left(1 - y_i \left\langle \sum_{j=1}^{m} \alpha_j y_j x_j, x_i \right\rangle \right) \right)
\]

rearranging we get the problem:

\[
\max_{\alpha \in \mathbb{R}^m : \alpha \geq 0} \left( \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \left\langle x_j, x_i \right\rangle \right)
\]

Note:

- solution is the vector \( \alpha \) which defines the support vectors \( \{x_i : \alpha_i \neq 0\} \)
- dual problem requires only to compute inner products \( \left\langle x_j, x_i \right\rangle \), does not need to consider \( x_i \) by itself