Machine Learning

A. Supervised Learning
A.8. Support Vector Machines (SVMs)

Lars Schmidt-Thieme, Nicolas Schilling

Information Systems and Machine Learning Lab (ISMLL)
Institute for Computer Science
University of Hildesheim, Germany

Outline

1. Maximum Margin Separating Hyperplanes
2. Lagrange Multipliers
3. Sequential Minimal Optimization
4. Kernel SVM

Outline

1. Maximum Margin Separating Hyperplanes

2. Lagrange Multipliers

3. Sequential Minimal Optimization

4. Kernel SVM

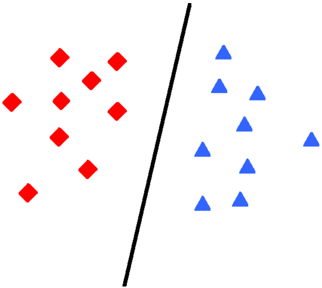
Separating Hyperplanes

For given data

$$(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$$

with a binary class label $Y \in \{-1, +1\}$
a hyperplane H_β is called **separating** if

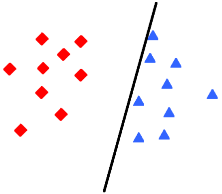
$$y_i h(x_i) > 0, \quad i = 1, \ldots, n, \quad \text{with } h(x) := \langle \beta, x \rangle + \beta_0$$



Linear Separable Data

The data is called **linear separable** if there exists such a separating hyperplane.

In general, if there is one, there are many, for example:



$\Rightarrow$ If there is a choice, we need a criterion to narrow down which one we want / is the best.

Maximum Margin Separating Hyperplanes

For linearly seperable data we wanted to solve the following problem:

$$
\begin{aligned}
\text{minimize } & \frac{1}{2}||\beta||^2 \\
\text{w.r.t. } y_i(\beta_0 + \langle \beta, x_i \rangle) \geq & 1, \quad i = 1, \ldots, n \\
\beta \in & \mathbb{R}^p \\
\beta_0 \in & \mathbb{R}
\end{aligned}
$$

This problem is a convex optimization problem that can be solved using Lagrange Multipliers.

Maximum Margin Separating Hyperplanes

For non seperable data we want to find a hyperplane that has

- few number of points on the wrong side
- wrong points very close to the hyperplane

This can be modelled using slack variables ξ_i:

$$\begin{aligned} \text{minimize } & \frac{1}{2}||\beta||^2 + C\sum_{i=1}^{n}\xi_i \\ \text{w.r.t. } y_i(\beta_0 + \langle\beta, x_i\rangle) &\geq 1 - \xi_i, \quad i = 1, \ldots, n \\ \xi &\geq 0 \\ \beta &\in \mathbb{R}^p \\ \beta_0 &\in \mathbb{R} \end{aligned}$$

for some positive constant C

Outline

1. Maximum Margin Separating Hyperplanes

2. Lagrange Multipliers

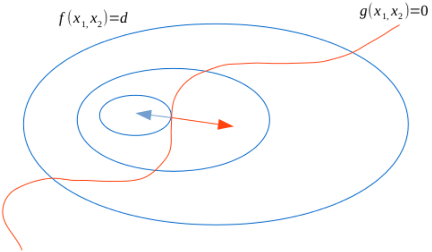
3. Sequential Minimal Optimization

4. Kernel SVM

Introduction

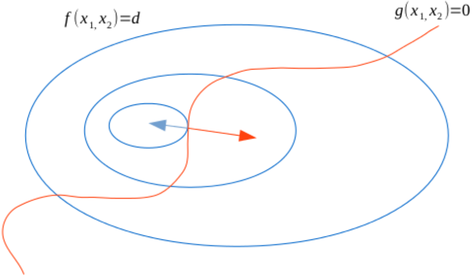
Suppose we want to maximize a function $f(x_1, x_2)$ subject to an equality constraint:

$$\max f(x_1, x_2) \quad \text{subject to} \quad g(x_1, x_2) = 0$$



- blue lines could be height lines
- red line is a hiking path
- find the highest point on the hiking path

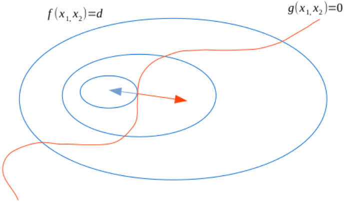
Introduction



Suppose, we walk along the red line and search for points where f does not change (candidates for maxima)

- happens if we walk along a contour line of f
- happens if we reach a "level part" of f (region of constanf f)
- find the highest point on the hiking path

Introduction



If g follows a contour line of f it means

- g and a contour line of f are parallel
- then the gradients of g and f have to be parallel as well

Thus:

$$\nabla_{x_1,x_2} f(x_1, x_2) = \lambda \nabla_{x_1,x_2} g(x_1, x_2)$$

This equality still holds for the second case, if we have reached a level part of f, as then its gradient is zero and λ can be set to zero

Lagrange Function

The equality can be written within one equation:

$$\mathcal{L}(x_1, x_2, \lambda) = f(x_1, x_2) + \lambda g(x_1, x_2)$$

where we then solve:

$$\nabla_{x_1, x_2, \lambda} \mathcal{L}(x_1, x_2, \lambda) = 0$$

Thus we have a system of equations:

$$\begin{aligned} \nabla_{x_1} \mathcal{L}(x_1, x_2, \lambda) &= 0 \\ \nabla_{x_2} \mathcal{L}(x_1, x_2, \lambda) &= 0 \\ \nabla_{\lambda} \mathcal{L}(x_1, x_2, \lambda) = g(x_1, x_2) &= 0 \end{aligned}$$

Dual Lagrange Function

The dual lagrange function is defined as:

$$L(\lambda) = \inf_{x_1, x_2} \mathcal{L}(x_1, x_2, \lambda)$$

Thus, we first solve

$$\nabla_{x_1, x_2} \mathcal{L}(x_1, x_2, \lambda) = 0$$

And then substitute the resulting x_1 and x_2 (which depend still on λ) into $\mathcal{L}$ which yields the dual problem.

Example

Suppose we want to minimize:

$$f(x_1, x_2) = {x_1}^2 + {x_2}^2$$

subject to:

$$2x_1 - x_2 + 3 = 0$$

We have the following Lagrangian:

$$\mathcal{L}(x_1, x_2, \lambda) = {x_1}^2 + {x_2}^2 + \lambda(2x_1 - x_2 + 3)$$

Computing the derivatives, setting them to zero and solving yields:

$$\nabla_{x_1}\mathcal{L}(x_1, x_2, \lambda) = 2x_1 + 2\lambda = 0 \quad \Longrightarrow \quad x_1 = -\lambda$$

and

$$\nabla_{x_2}\mathcal{L}(x_1, x_2, \lambda) = 2x_2 - \lambda = 0 \quad \Longrightarrow \quad x_2 = \lambda/2$$

Example

We can input these solutions into the constraint to compute the final λ (which then yields the solution)

$$-2\lambda - \lambda/2 + 3 = 0$$

is equivalent to

$$-4\lambda - \lambda + 6 = 0$$

which yields the solution

$$\lambda = 6/5$$

and thus we obtain

$$x_1 = -6/5 \qquad x_2 = 3/5$$

Example

We can also input these solutions into the Lagrangian to obtain the dual problem:

$$L(\lambda) = (-\lambda)^2 + (\lambda/2)^2 + \lambda(-2\lambda - \lambda/2 + 3)$$

which we can then further simplify and maximize with respect to λ!

Lagrangian Function of SVM

$$
\begin{aligned}
\text{minimize } & \frac{1}{2}||\beta||^2 + C\sum_{i=1}^{n}\xi_i \\
\text{w.r.t. } y_i(\beta_0 + \langle\beta, x_i\rangle) \geq & 1 - \xi_i, \quad i = 1, \ldots, n \\
\xi \geq & 0 \\
\beta \in & \mathbb{R}^p \\
\beta_0 \in & \mathbb{R}
\end{aligned}
$$

The Lagrange function of this problem is

$$\mathcal{L} := \frac{1}{2}||\beta||^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i(y_i(\beta_0 + \langle\beta, x_i\rangle) - (1 - \xi_i)) - \sum_{i=1}^{n}\mu_i\xi_i$$

with the multipliers

$$\alpha_i \geq 0 \qquad \text{and} \qquad \mu_i \geq 0$$

Lagrangian Function of SVM

$$\mathcal{L} := \frac{1}{2}||\beta||^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i(y_i(\beta_0 + \langle\beta, x_i\rangle) - (1-\xi_i)) - \sum_{i=1}^{n}\mu_i\xi_i$$

For an extremum it is required that

$$\frac{\partial\mathcal{L}}{\partial\beta} = \beta - \sum_{i=1}^{n}\alpha_i y_i x_i \overset{!}{=} 0$$

$$\Rightarrow \beta = \sum_{i=1}^{n}\alpha_i y_i x_i$$

Lagrangian Function of SVM

Moreover we have:

$$\frac{\partial \mathcal{L}}{\partial \beta_0} = -\sum_{i=1}^{n} \alpha_i y_i \overset{!}{=} 0$$

and we also have to derive with respect to ξ_i

$$\frac{\partial \mathcal{L}}{\partial \xi_i} = C - \alpha_i - \mu_i \overset{!}{=} 0$$

which yields

$$\alpha_i = C - \mu_i$$

which implies that

$$\alpha_i \in [0, C] \quad \text{as} \quad \mu_i \geq 0$$

Dual Lagrangian Function of SVM

Now we can put these solutions into the Lagrangian

$$\beta = \sum_{i=1}^{n} \alpha_i y_i x_i, \quad \sum_{i=1}^{n} \alpha_i y_i = 0, \quad \alpha_i = C - \mu_i$$

into

$$\mathcal{L} := \frac{1}{2}||\beta||^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i(y_i(\beta_0 + \langle\beta, x_i\rangle) - (1-\xi)) - \sum_{i=1}^{n}\mu_i\xi_i$$

which yields the **dual problem**

Dual Lagrangian Function of SVM

$$
\begin{aligned}
L =& \frac{1}{2}\langle \sum_{i=1}^{n} \alpha_i y_i x_i, \sum_{j=1}^{n} \alpha_j y_j x_j \rangle - \sum_{i=1}^{n} \alpha_i (y_i(\beta_0 + \langle \sum_{j=1}^{n} \alpha_j y_j x_j, x_i \rangle) - (1 - \xi_i)) \\
& + C \sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \mu_i \xi_i \\
=& \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle + \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \alpha_i y_i \beta_0 - \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \\
& - \sum_{i=1}^{n} \alpha_i \xi_i + \sum_{i=1}^{n} \alpha_i \xi_i \\
=& -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle + \sum_{i=1}^{n} \alpha_i
\end{aligned}
$$

Dual Problem

The dual problem is

$$\begin{aligned}
\text{maximize } L = &-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j\langle x_i, x_j\rangle + \sum_{i=1}^{n}\alpha_i \\
\text{w.r.t. } \sum_{i=1}^{n}\alpha_i y_i &= 0 \\
\alpha_i &\leq C \\
\alpha_i &\geq 0
\end{aligned}$$

with much simpler constraints.

Predicting with SVMs

procedure $\text{PREDICT-SVM}(\alpha \in (\mathbb{R}_0^+)^n, \beta_0 \in \mathbb{R}, \mathcal{D}^{\text{train}} := \{(x_1, y_1), \ldots, (x_n, y_n)\})$

    $\hat{y} := \beta_0$

    for $i := 1, \ldots, n$ with $\alpha_i \neq 0$ **do**

        $\hat{y} := \hat{y} + \alpha_i y_i \langle x_i, x \rangle)$

    return $\hat{y}$

Note: $\hat{y}$ yields the score/certainty factor, sign $\hat{y}$ the predicted class.
From $\mathcal{D}^{\text{train}}$, only the support vectors (x_i, y_i) (having $\alpha_i > 0$) are required.

Outline

1. Maximum Margin Separating Hyperplanes

2. Lagrange Multipliers

3. Sequential Minimal Optimization

4. Kernel SVM

Sequential Minimal Optimization (SMO)

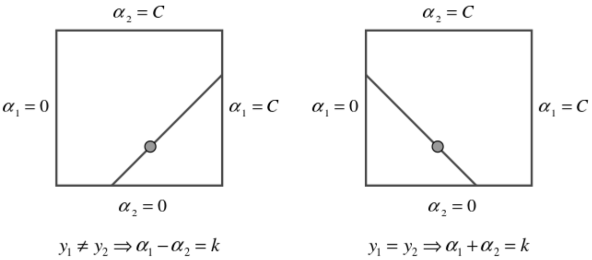
SMO (Platt, 1999) iteratively solves sub problems to finally solve the whole problem. It repeats the following:

- pick two α parameters
- optimize one α through a Newton step
- compute the second α using the reduced equality constraint
- compute the bias term β_0

Box Constraints

Let α_1 and α_2 be two chosen α. Let us first optimize α_2 and then α_1. Our constraint reduces to:

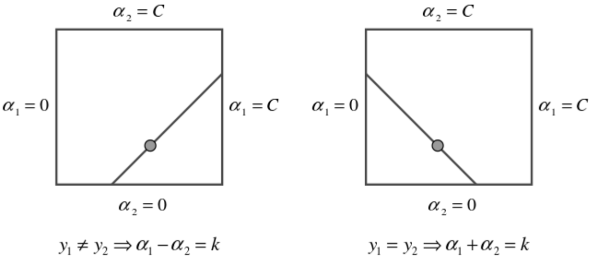
$$\alpha_1 y_1 + \alpha_2 y_2 = -\sum_{i=3}^{n} \alpha_i y_i =: k$$



There are two cases, either both associated instances have the same label $y_1 = y_2$ or they don't.

Box Constraints

Let us assume both labels are not equal (left)

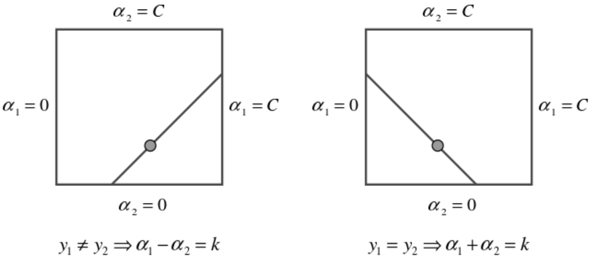


α_2 is now bounded in an interval $[L, H]$ with:

$$L = \max(0, \alpha_2 - \alpha_1) \qquad H = \min(C, C + \alpha_2 - \alpha_1)$$

Box Constraints

Let us assume both labels are equal (right)



α_2 is now bounded in an interval $[L, H]$ with:

$$L = \max(0, \alpha_2 + \alpha_1 - C) \qquad H = \min(C, \alpha_2 + \alpha_1)$$

Box Constraints

SMO then computes the minimum of L along the direction of the constraint via:

$$\alpha_2{}^{\text{new}} = \alpha_2 + \frac{y_2(E_1 - E_2)}{\eta}$$

where

$$\eta = \langle x_1, x_1 \rangle + \langle x_2, x_2 \rangle - 2\langle x_1, x_2 \rangle$$

and

$$E_i = \hat{y}_i - y_i$$

the error on the i-th training instance.

Box Constraints

In order to fulfill the interval constraints for α_2 we have to clip it:

$$\alpha_2{}^{\text{new,clipped}} = \begin{cases} H & \text{if} \quad \alpha_2{}^{\text{new}} \geq H \\ \alpha_2{}^{\text{new}} & \text{if} \quad L \leq \alpha_2{}^{\text{new}} \leq H \\ L & \text{if} \quad \alpha_2{}^{\text{new}} < H \end{cases}$$

This way, we ensure that

$$0 \leq \alpha_2{}^{\text{new,clipped}} \leq C$$

Computation of α_1

The new parameters have to fulfill the constraint:

$$\alpha_1^{\text{new}} + s\alpha_2^{\text{new,clipped}} = k = \alpha_1 + s\alpha_2$$

We can reformulate this to

$$\alpha_1^{\text{new}} = \alpha_1 + s(\alpha_2 - \alpha_2^{\text{new,clipped}})$$

Computation of the threshold

As a final step we have to compute the threshold β_0. It can be shown that $\beta_0 \in [b_1, b_2]$ is feasible for:

$$b_1 = E_1 + y_1({\alpha_1}^{\text{new}} - \alpha_1)\langle x_1, x_1\rangle + y_2({\alpha_2}^{\text{new,clipped}} - \alpha_2)\langle x_1, x_2\rangle + \beta_0$$

and

$$b_2 = E_2 + y_1({\alpha_1}^{\text{new}} - \alpha_1)\langle x_1, x_2\rangle + y_2({\alpha_2}^{\text{new,clipped}} - \alpha_2)\langle x_2, x_2\rangle + \beta_0$$

Outline

1. Maximum Margin Separating Hyperplanes

2. Lagrange Multipliers

3. Sequential Minimal Optimization

4. Kernel SVM

Support Vectors

For points on the right side of the hyperplane,

$$y_i(\beta_0 + \langle \beta, x_i \rangle) > 1, \quad \xi_i = 0$$

then L is maximized by $\alpha_i = 0$: x_i is irrelevant.

For points in the margin as well as on the wrong side of the hyperplane,

$$y_i(\beta_0 + \langle \beta, x_i \rangle) = 1 - \xi_i, \quad \xi_i > 0$$

α_i is some finite value.

For points on the margin, i.e.,

$$y_i(\beta_0 + \langle \beta, x_i \rangle) = 1, \quad \xi_i = 0$$

α_i is some finite value.

The data points x_i with $\alpha_i > 0$ are called **support vectors**.

Decision Function

Due to

$$\beta = \sum_{i=1}^{n} \alpha_i y_i x_i,$$

the decision function

$$\hat{y}(x) = \operatorname{sign} \beta_0 + \langle \beta, x \rangle$$

can be expressed using the training data:

$$\hat{y}(x) = \operatorname{sign} \beta_0 + \sum_{i=1}^{n} \alpha_i y_i \langle x_i, x \rangle$$

Only support vectors are required, as only for them $\alpha_i \neq 0$.

Both, the learning problem and the decision function can be expressed using an inner product / a similarity measure / a kernel $\langle x, x' \rangle$.

High-Dimensional Embeddings / The "kernel trick"

Example:

we map points from R^2 into the higher dimensional space $\mathbb{R}^6$ via

$$h : \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \mapsto \begin{pmatrix} 1 \\ \sqrt{2}x_1 \\ \sqrt{2}x_2 \\ x_1^2 \\ x_2^2 \\ \sqrt{2}x_1 x_2 \end{pmatrix}$$

Then the inner product

$$\langle h(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}), h(\begin{pmatrix} x'_1 \\ x'_2 \end{pmatrix})\rangle = 1 + 2x_1x'_1 + 2x_2x'_2 + x_1^2{x'_1}^2 + x_2^2{x'_2}^2 + 2x_1x_2x'_1x'_2$$
$$= (1 + x_1x'_1 + x_2x'_2)^2$$

can be computed without having to compute h explicitely !

Popular Kernels

Some popular kernels are:

linear kernel:

$$K(x, x') := \langle x, x' \rangle := \sum_{i=1}^{n} x_i x_i'$$

polynomial kernel of degree d:

$$K(x, x') := (1 + \langle x, x' \rangle)^d$$

radial basis kernel / gaussian kernel :

$$K(x, x') := e^{-\frac{||x-x'||^2}{c}}$$

neural network kernel / sigmoid kernel :

$$K(x, x') := \tanh(a\langle x, x' \rangle + b)$$

Predicting with SVMs

1: **procedure** PREDICT-SVM($\alpha \in (\mathbb{R}_0^+)^n, \beta_0 \in \mathbb{R}, \mathcal{D}^{\text{train}} := \{(x_1, y_1), \ldots, (x_n, y_n)\}, K$)
2: $\hat{y} := \beta_0$
3: **for** $i := 1, \ldots, n$ with $\alpha_i \neq 0$ **do**
4: $\hat{y} := \hat{y} + \alpha_i y_i K(x_i, x)$
5: **return** $\hat{y}$

Note: $\hat{y}$ yields the score/certainty factor, sign $\hat{y}$ the predicted class.
From $\mathcal{D}^{\text{train}}$, only the support vectors (x_i, y_i) (having $\alpha_i > 0$) are required.

Summary (1/2)

- Binary classification problems with linear decision boundaries can be rephrased as finding a **separating hyperplane**.
- In the **linear separable case**, there are simple algorithms like **perceptron** learning to find such a separating hyperplane.
- If one requires the additional property that the hyperplane should have **maximal margin**, i.e., maximal distance to the closest points of both classes, then a quadratic optimization problem with inequality constraints arises.

Summary (2/2)

- Optimal hyperplanes can also be formulated for the **linear inseparable case** by allowing some points to be on the wrong side of the margin, but penalize for their distance from the margin. This also can be formulated as a quadratic optimization problem with inequality constraints.

- The final decision function can be computed in terms of inner products of the query points with some of the data points (called **support vectors**), which allows to bypass the explicit computation of high dimensional embeddings (**kernel trick**).