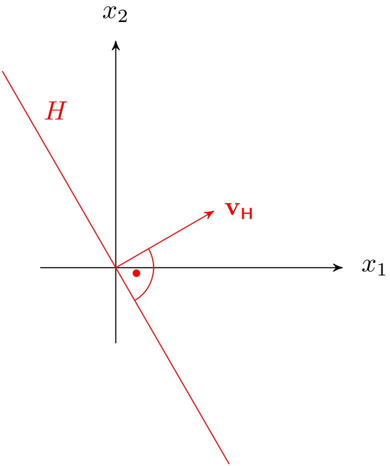
# Lecture 7: Support Vector Machines I

## Reading: Section 12.2

GU4241/GR5241 Statistical Machine Learning

Linxi Liu
February 15, 2019

# Hyperplanes




## Hyperplanes

A **hyperplane** in $\mathbb{R}^d$ is a linear subspace of dimension $(d-1)$.

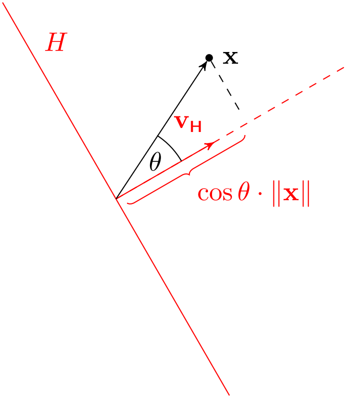
- A $\mathbb{R}^2$-hyperplane is a line, a $\mathbb{R}^3$-hyperplane is a plane.
- As a linear subspace, a hyperplane always contains the origin.

## Normal vectors

A hyperplane $H$ can be represented by a **normal vector**. The hyperplane with normal vector $\mathbf{v}_\mathsf{H}$ is the set

$$H = \{\mathbf{x} \in \mathbb{R}^d \mid \langle \mathbf{x}, \mathbf{v}_\mathsf{H} \rangle = 0\} .$$

# Which side of the plane are we on?




- The projection of $\mathbf{x}$ onto the direction of $\mathbf{v}_\mathsf{H}$ has length $\langle \mathbf{x}, \mathbf{v}_\mathsf{H} \rangle$ *measured in units of* $\mathbf{v}_H$, i.e. length $\langle \mathbf{x}, \mathbf{v}_\mathsf{H} \rangle / \|\mathbf{v}_\mathsf{H}\|$ in the units of the coordinates.
- Recall the cosine rule for the scalar product,

$$\cos\theta = \frac{\langle \mathbf{x}, \mathbf{v}_\mathsf{H} \rangle}{\|\mathbf{x}\| \cdot \|\mathbf{v}_\mathsf{H}\|} \ .$$

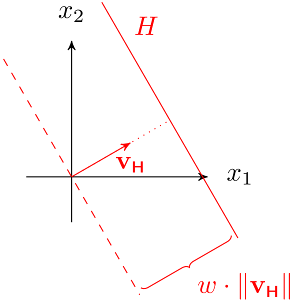
- Consequence: The distance of $\mathbf{x}$ from the plane is given by

$$d(\mathbf{x}, H) = \frac{\langle \mathbf{x}, \mathbf{v}_\mathsf{H} \rangle}{\|\mathbf{v}_\mathsf{H}\|} = \cos\theta \cdot \|\mathbf{x}\| \ .$$

- We can decide which side of the plane $\mathbf{x}$ is on using

$$\mathsf{sgn}(\cos\theta) = \mathsf{sgn}\,\langle \mathbf{x}, \mathbf{v}_\mathsf{H} \rangle \ .$$

# Affine Hyperplanes




## Affine Hyperplanes

- An **affine hyperplane** $H_\mathbf{w}$ is a hyperplane translated (shifted) by a vector $\mathbf{w}$, i.e. $H_\mathbf{w} = H + \mathbf{w}$.
- We choose $\mathbf{w}$ in the direction of $\mathbf{v_H}$, i.e. $\mathbf{w} = c \cdot \mathbf{v_H}$ for $c > 0$.

## Which side of the plane?

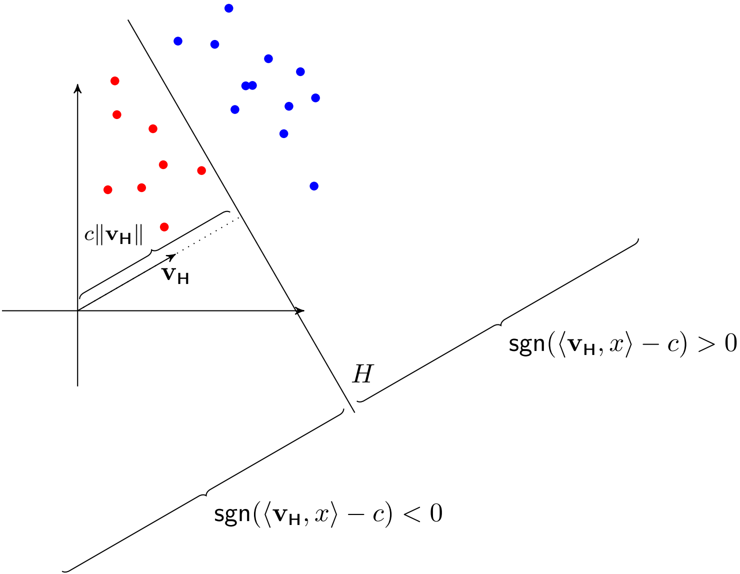
- Which side of $H_\mathbf{w}$ a point $\mathbf{x}$ is on is determined by

$$\text{sgn}(\langle \mathbf{x} - \mathbf{w}, \mathbf{v_H} \rangle) = \text{sgn}(\langle \mathbf{x}, \mathbf{v_H} \rangle - c \langle \mathbf{v_H}, \mathbf{v_H} \rangle) = \text{sgn}(\langle \mathbf{x}, \mathbf{v_H} \rangle - c \|\mathbf{v_H}\|^2) \ .$$

- If $\mathbf{v_H}$ is a unit vector, we can use

$$\text{sgn}(\langle \mathbf{x} - \mathbf{w}, \mathbf{v_H} \rangle) = \text{sgn}(\langle \mathbf{x}, \mathbf{v_H} \rangle - c) \ .$$

# Classification with Affine Hyperplanes



# Linear Classifiers

## Definition

A **linear classifier** is a function of the form

$$f_{\mathsf{H}}(\mathbf{x}) := \operatorname{sgn}(\langle \mathbf{x}, \mathbf{v}_{\mathsf{H}} \rangle - c) ,$$

where $\mathbf{v}_{\mathsf{H}} \in \mathbb{R}^d$ is a vector and $c \in \mathbb{R}_+$.

**Note:** We usually assume $\mathbf{v}_{\mathsf{H}}$ to be a unit vector. If it is not, $f_{\mathsf{H}}$ still defines a linear classifier, but $c$ describes a shift of a different length.

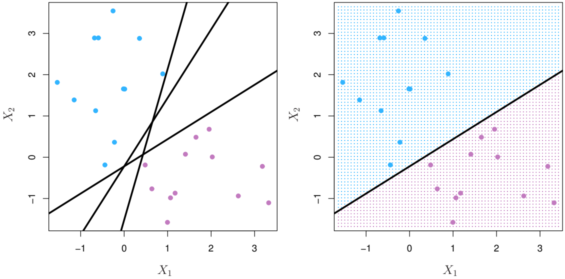
## Definition

Two sets $A, B \in \mathbb{R}^d$ are called **linearly separable** if there is an affine hyperplane $H$ which separates them, i.e. which satisfies

$$\langle \mathbf{x}, \mathbf{v}_{\mathsf{H}} \rangle - c = \begin{cases} < 0 & \text{if } \mathbf{x} \in A \\ > 0 & \text{if } \mathbf{x} \in B \end{cases}$$

# Maximum Margin Idea

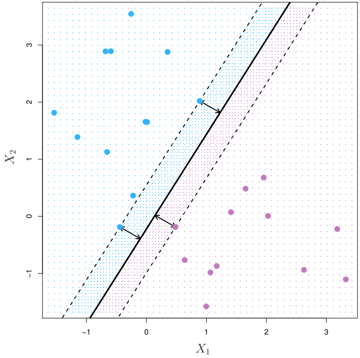
- Suppose we have a classification problem with response $Y = -1$ or $Y = 1$.
- If the classes can be separated, most likely, there will be an infinite number of hyperplanes separating the classes.



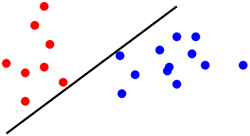
# Maximum Margin Idea

**Idea:**

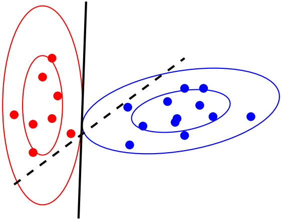
- Draw the largest possible empty margin around the hyperplane.
- Out of all possible hyperplanes that separate the 2 classes, choose the one such that distance to closest point in each class is maximal. This distance is called the *margin*.



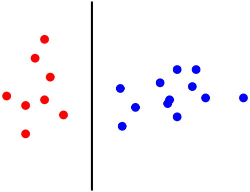
# Generalization Error



Possible solution



Good generalization under a specific distribution (here: Gaussian)



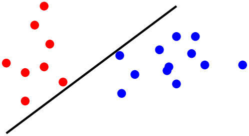
Maximum margin solution

## Example: Gaussian data

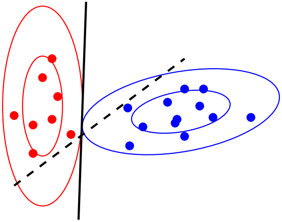
- The ellipses represent lines of constant standard deviation (1 and 2 STD respectively).
- The 1 STD ellipse contains $\sim 65\%$ of the probability mass ($\sim 95\%$ for 2 STD; $\sim 99.7\%$ for 3 STD).

**Optimal generalization:** Classifier should cut off as little probability mass as possible from either distribution.

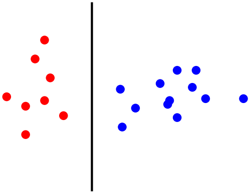
# Generalization Error



Possible solution



Good generalization under a specific distribution (here: Gaussian)



Maximum margin solution

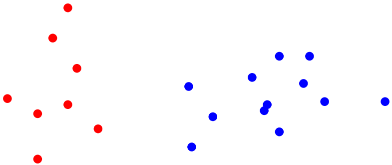
## Without distributional assumption: Max-margin classifier

- Philosophy: Without distribution assumptions, best guess is symmetric.
- In the Gaussian example, the max-margin solution would *not* be optimal.

# Substituting convex sets

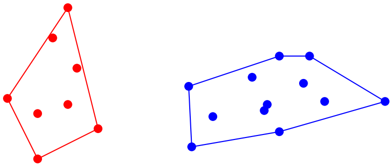
## Observation

Where a separating hyperplane may be placed depends on the "outer" points on the sets. Points in the center do not matter.



## In geometric terms

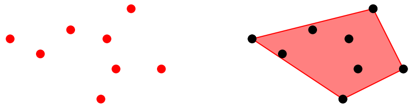
Substitute each class by the smallest convex set which contains all point in the class:



# Substituting convex sets

## Definition

If $C$ is a set of points, the smallest convex set containing all points in $C$ is called the **convex hull** of $C$, denoted $\mathrm{conv}(C)$.

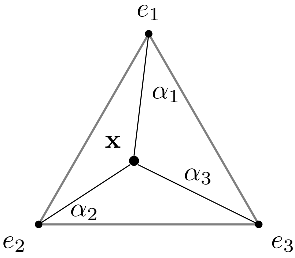


Corner points of the convex set are called **extreme points**.

## Barycentric coordinates

Every point $x$ in a convex set can be represented as a convex combination of the extreme points $\{e_1, \ldots, e_m\}$. There are weights $\alpha_1, \ldots, \alpha_m \in \mathbb{R}_+$ such that

$$\mathbf{x} = \sum_{i=1}^{m} \alpha_i e_i \quad \text{and} \quad \sum_{i=1}^{m} \alpha_i = 1 \ .$$

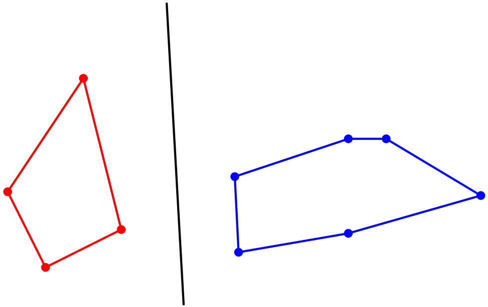



The coefficients $\alpha_i$ are called **barycentric coordinates** of $x$.

# Convex Hulls and Classification

## Key idea

A hyperplane separates two classes if and only if it separates their convex hull.



**Next:** We have to formalize what it means for a hyperplane to be "in the middle" between to classes.

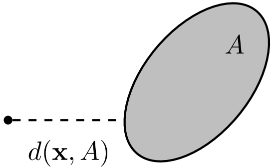
# Distances to sets

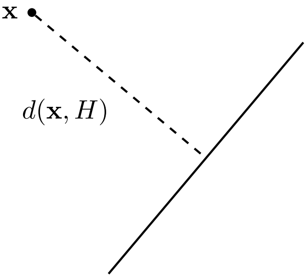
## Definition

The **distance** between a point $\mathbf{x}$ and a set $A$ the Euclidean distance between $x$ and the closest point in $A$:

$$d(\mathbf{x}, A) := \min_{\mathbf{y} \in A} \|\mathbf{x} - \mathbf{y}\|$$

In particular, if $A = H$ is a hyperplane, $d(\mathbf{x}, H) := \min_{\mathbf{y} \in H} \|\mathbf{x} - \mathbf{y}\|$.



# Margin

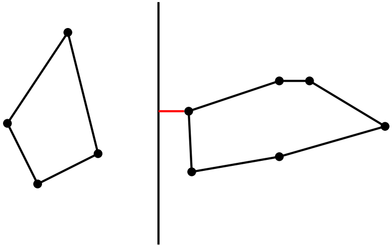
## Definition

The **margin** of a classifier hyperplane $H$ given two training classes $\mathcal{X}_\ominus, \mathcal{X}_\oplus$ is the shortest distance between the plane and any point in either set:

$$\text{margin} = \min_{x \in \mathcal{X}_\ominus \cup \mathcal{X}_\oplus} d(x, H)$$

Equivalently: The shortest distance to either of the convex hulls.

$$\text{margin} = \min\{d(H, \text{conv}(\mathcal{X}_\ominus)), d(H, \text{conv}(\mathcal{X}_\oplus))\}$$



Idea in the following: $H$ is "in the middle" when margin maximal.

# Linear Classifier with Margin

## Recall: Specifying affine plane

Normal vector $\mathbf{v}_{\mathsf{H}}$.

$$\langle \mathbf{v}_{\mathsf{H}}, \mathbf{x} \rangle - c \begin{cases} > 0 & \mathbf{x} \text{ on positive side} \\ < 0 & \mathbf{x} \text{ on negative side} \end{cases}$$

Scalar $c \in \mathbb{R}$ specifies shift (plane through origin if $c = 0$).

## Plane with margin

Demand

$$\langle \mathbf{v}_{\mathsf{H}}, \mathbf{x} \rangle - c > 1 \text{ or } < -1$$

$\{-1, 1\}$ on the right works for any margin: Size of margin determined by $\|\mathbf{v}_{\mathsf{H}}\|$. To increase margin, scale down $\mathbf{v}_{\mathsf{H}}$.

## Classification

Concept of margin applies only to training, not to classification.
Classification works as for any linear classifier. For a test point $\mathbf{x}$:

$$y = \operatorname{sign}\left(\langle \mathbf{v}_{\mathsf{H}}, \mathbf{x} \rangle - c\right)$$

# Support Vector Machine

## Finding the hyperplane

For $n$ training points $(\tilde{\mathbf{x}}_i, \tilde{y}_i)$ with labels $\tilde{y}_i \in \{-1, 1\}$, solve optimization problem:

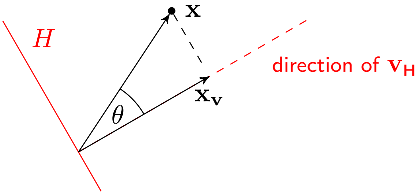
$$\begin{array}{lll} \min\limits_{\mathbf{v}_{\mathsf{H}}, c} & \|\mathbf{v}_{\mathsf{H}}\| & \\ \text{s.t.} & \tilde{y}_i(\langle \mathbf{v}_{\mathsf{H}}, \tilde{\mathbf{x}}_i \rangle - c) \geq 1 & \text{for } i = 1, \dots, n \end{array}$$

## Definition

The classifier obtained by solving this optimization problem is called a **support vector machine**.

# Why *minimize* $\|\mathbf{v}_\mathrm{H}\|$?

We can project a vector $\mathbf{x}$ (think: data point) onto the direction of $\mathbf{v}_\mathrm{H}$ and obtain a vector $\mathbf{x}_\mathbf{v}$.




- If $H$ has no offset ($c = 0$), the Euclidean distance of $\mathbf{x}$ from $H$ is

  $$d(\mathbf{x}, H) = \|\mathbf{x}_\mathbf{v}\| = \cos\theta \cdot \|\mathbf{x}\| \ .$$

  It does not depend on the length of $\mathbf{v}_\mathrm{H}$.
- The scalar product $\langle \mathbf{x}, \mathbf{v}_\mathrm{H} \rangle$ *does* increase if the length of $\mathbf{v}_\mathrm{H}$ increases.
- To compute the distance $\|\mathbf{x}_\mathbf{v}\|$ from $\langle \mathbf{x}, \mathbf{v}_\mathrm{H} \rangle$, we have to scale out $\|\mathbf{v}_\mathrm{H}\|$:

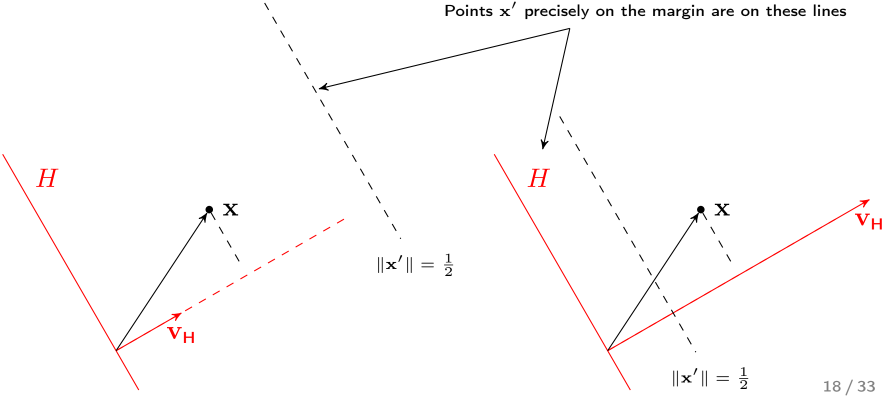
  $$\|\mathbf{x}_\mathbf{v}\| = \cos\theta \cdot \|\mathbf{x}\| = \frac{\langle \mathbf{x}, \mathbf{v}_\mathrm{H} \rangle}{\|\mathbf{v}_\mathrm{H}\|}$$

# Why *minimize* $\|\mathbf{v}_\mathsf{H}\|$?

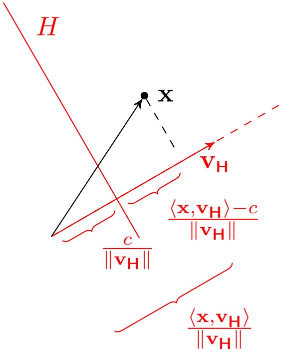
If we scale $\mathbf{v}_\mathsf{H}$ by $\alpha$, we have to scale $\mathbf{x}$ by $1/\alpha$ to keep $\langle \mathbf{v}_\mathsf{H}, \mathbf{x} \rangle$ constant, e.g.:

$$1 = \langle \mathbf{v}_\mathsf{H}, \mathbf{x} \rangle = \left\langle \alpha \mathbf{v}_\mathsf{H}, \frac{1}{\alpha}\mathbf{x} \right\rangle .$$

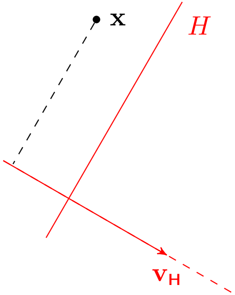
A point $\mathbf{x}'$ is precisely on the margin if $\langle \mathbf{x}', \mathbf{v}_\mathsf{H} \rangle = 1$.
Look at what happens if we scale $\mathbf{v}_\mathsf{H}$:



# Distance With Offset




For an affine plane, we have to substract the offset.



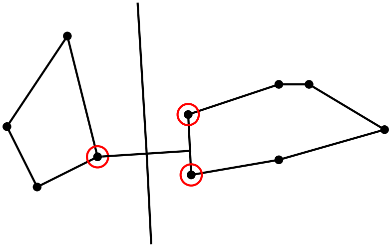

The optimization algorithm can also rotate the vector $\mathbf{v_H}$, which rotates the plane.

# Support Vectors

## Definition

Those extreme points of the convex hulls which are closest to the hyperplane are called the **support vectors**.

There are at least two support vectors, one in each class.



## Implications

- The maximum-margin criterion focuses all attention to the area closest to the decision surface.
- Small changes in the support vectors can result in significant changes of the classifier.
- In practice, the approach is combined with "slack variables" to permit overlapping classes. As a side effect, slack variables soften the impact of changes in the support vectors.

# Dual Optimization Problem

Solving the SVM opimization problem

$$\begin{aligned} \min_{\mathbf{v_H}, c} \quad & \|\mathbf{v_H}\| \\ \text{s.t.} \quad & \tilde{y}_i(\langle \mathbf{v_H}, \tilde{\mathbf{x}}_i \rangle - c) \geq 1 \quad \text{for } i = 1, \ldots, n \end{aligned}$$

is difficult, because the constraint is a function. It is possible to transform this problem into a problem which seems more complicated, but has simpler constraints:

$$\begin{aligned} \max_{\boldsymbol{\alpha} \in \mathbb{R}^n} \quad & W(\boldsymbol{\alpha}) := \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j \tilde{y}_i \tilde{y}_j \langle \tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j \rangle \\ \text{s.t.} \quad & \sum_{i=1}^{n} \tilde{y}_i \alpha_i = 0 \\ & \alpha_i \geq 0 \quad \text{for } i = 1, \ldots, n \end{aligned}$$

This is called the optimization problem **dual** to the minimization problem above. It is usually derived using Lagrange multipliers. We will use a more geometric argument.

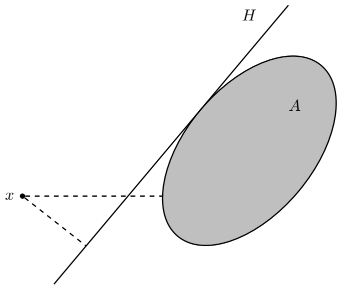
# Convex Duality

## Sets and Planes

Many dual relations in convex optimization can be traced back to the following fact:

*The closest distance between a point $\mathbf{x}$ and a convex set $A$ is the maximum over the distances between $\mathbf{x}$ and all hyperplanes which separate $\mathbf{x}$ and $A$.*

$$d(\mathbf{x}, A) = \sup_{H \text{ separating}} d(\mathbf{x}, H)$$



# Deriving the Dual Problem

## Idea

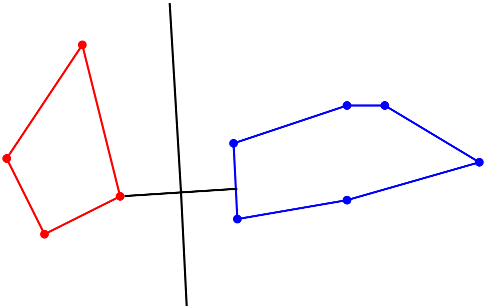
As a consequence of duality on previous slide, we can find the maximum-margin plane as follows:

1. Find shortest line connecting the convex hulls.
2. Place classifier orthogonal to line in the middle.

Convexity of sets ensures that this classifier has correct orientation.

## As optimization problem

$$\min_{\substack{\mathbf{u} \in \mathsf{conv}(\mathcal{X}_\ominus) \\ \mathbf{v} \in \mathsf{conv}(\mathcal{X}_\oplus)}} \quad \|\mathbf{u} - \mathbf{v}\|^2$$



# Barycentric Coordinates

## Dual optimization problem

$$\min_{\substack{\mathbf{u}\in\mathsf{conv}(\mathcal{X}_\ominus)\\ \mathbf{v}\in\mathsf{conv}(\mathcal{X}_\oplus)}} \qquad \|\mathbf{u}-\mathbf{v}\|^2$$

As points in the convex hulls, $\mathbf{u}$ and $\mathbf{v}$ can be represented by barycentric coordinates:

$$\mathbf{u} = \sum_{i=1}^{n_1} \alpha_i \tilde{\mathbf{x}}_i \qquad \mathbf{v} = \sum_{i=n_1+1}^{n_1+n_2} \alpha_i \tilde{\mathbf{x}}_i \qquad\qquad \text{(where } n_1 = |\mathcal{X}_\ominus|, n_2 = |\mathcal{X}_\oplus|\text{)}$$

The extreme points suffice to represent any point in the sets. If $\tilde{\mathbf{x}}_i$ is not an extreme point, we can set $\alpha_i = 0$.

Substitute into minimization problem:

$$\begin{aligned} \min_{\alpha_1,\ldots,\alpha_n} \qquad & \Big\| \sum_{i\in\mathcal{X}_\ominus} \alpha_i \tilde{\mathbf{x}}_i - \sum_{i\in\mathcal{X}_\oplus} \alpha_i \tilde{\mathbf{x}}_i \Big\|_2^2 \\ \text{s.t.} \qquad & \sum_{i\in\mathcal{X}_\ominus} \alpha_i = \sum_{i\in\mathcal{X}_\oplus} \alpha_i = 1, \quad \alpha_i \geq 0 \end{aligned}$$

# Dual optimization problem

## Dual problem

$$
\begin{aligned}
\| \sum_{i \in \mathcal{X}_\ominus} \alpha_i \tilde{\mathbf{x}}_i - \sum_{i \in \mathcal{X}_\oplus} \alpha_i \tilde{\mathbf{x}}_i \|_2^2 &= \| \sum_{i \in \mathcal{X}_\ominus} \tilde{y}_i \alpha_i \tilde{\mathbf{x}}_i + \sum_{i \in \mathcal{X}_\oplus} \tilde{y}_i \alpha_i \tilde{\mathbf{x}}_i \|_2^2 \\
&= \left\langle \sum_{i=1}^{n} \tilde{y}_i \alpha_i \tilde{\mathbf{x}}_i \, , \, \sum_{i=1}^{n} \tilde{y}_i \alpha_i \tilde{\mathbf{x}}_i \right\rangle \\
&= \sum_{i,j} \tilde{y}_i \tilde{y}_j \alpha_i \alpha_j \left\langle \tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j \right\rangle
\end{aligned}
$$

Note: Minimizing this term under the constraints is equivalent to *maximizing*

$$
\sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \tilde{y}_i \tilde{y}_j \alpha_i \alpha_j \left\langle \tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j \right\rangle
$$

under the same constraints, since $\sum_i \alpha_i = 2$ is constant. That is just the dual problem defined four slides back.

# Computing $c$

## Output of dual problem

$$\mathbf{v}_{\mathsf{H}}^* := \mathbf{v}^* - \mathbf{u}^* = \sum_{i=1}^{n} \tilde{y}_i \alpha_i^* \tilde{\mathbf{x}}_i$$

This vector describes a hyperplane through the origin. We still have to compute the offset.

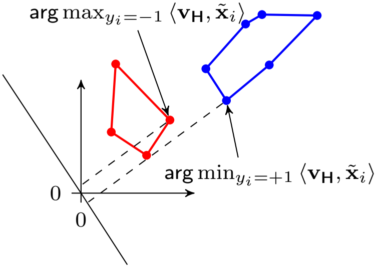
## Computing the offset

$$c^* := \frac{\max_{\tilde{y}_i=-1} \langle \mathbf{v}_{\mathsf{H}}^*, \tilde{\mathbf{x}}_i \rangle + \min_{\tilde{y}_i=+1} \langle \mathbf{v}_{\mathsf{H}}^*, \tilde{\mathbf{x}}_i \rangle}{2}$$

# Computing $c$

## Explanation

- The max and min are computed with respect to the $\mathbf{v_H}$ plane *containing the origin*.
- That means the max and min determine a support vector in each class.
- We then compute the shift as the mean of the two distances.



# Resulting Classification Rule

## Output of dual optimization

- Optimal values $\alpha_i^*$ for the variables $\alpha_i$
- If $\tilde{\mathbf{x}}_i$ support vector: $\alpha_i^* > 0$, if not: $\alpha_i^* = 0$

Note: $\alpha_i^* = 0$ holds even if $\tilde{\mathbf{x}}_i$ is an extreme point, but not a support vector.

## SVM Classifier

The classification function can be expressed in terms of the variables $\alpha_i$:

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{n} \tilde{y}_i \alpha_i^* \langle \tilde{\mathbf{x}}_i, \mathbf{x} \rangle - c^*\right)$$

Intuitively: To classify a data point, it is sufficient to know which side of each support vector it is on.

# Soft-Margin Classifiers

**Soft-margin classifiers** are maximum-margin classifiers which permit some points to lie on the wrong side of the margin, or even of the hyperplane.

## Motivation 1: Nonseparable data

SVMs are linear classifiers; without further modifications, they cannot be trained on a non-separable training data set.

## Motivation 2: Robustness

- Recall: Location of SVM classifier depends on position of (possibly few) support vectors.
- Suppose we have two training samples (from the same joint distribution on $(X, Y)$) and train an SVM on each.
- If locations of support vectors vary significantly between samples, SVM estimate of $\mathbf{v}_\mathbf{H}$ is "brittle" (depends too much on small variations in training data). $\longrightarrow$ Bad generalization properties.
- Methods which are not susceptible to small variations in the data are often referred to as **robust**.

# Slack Variables

## Idea

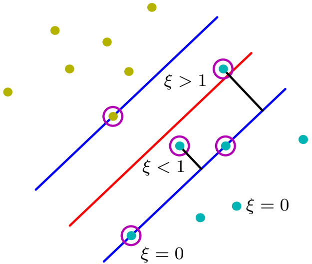
Permit training data to cross the margin, but impose cost which increases the further beyond the margin we are.

## Formalization

We replace the training rule $\tilde{y}_i(\langle \mathbf{v}_{\mathsf{H}}, \tilde{\mathbf{x}}_i \rangle - c) \geq 1$ by

$$\tilde{y}_i(\langle \mathbf{v}_{\mathsf{H}}, \tilde{\mathbf{x}}_i \rangle - c) \geq 1 - \xi_i$$

with $\xi_i \geq 0$. The variables $\xi_i$ are called **slack variables**.



# Soft-Margin SVM

## Soft-margin optimization problem

$$
\begin{aligned}
\min_{\mathbf{v_H}, c, \xi} \quad & \|\mathbf{v_H}\|^2 + \gamma \sum_{i=1}^{n} \xi_i^2 \\
\text{s.t.} \quad & \tilde{y}_i(\langle \mathbf{v_H}, \tilde{\mathbf{x}}_i \rangle - c) \geq 1 - \xi_i \quad \text{for } i = 1, \ldots, n \\
& \xi_i \geq 0, \quad \text{for } i = 1, \ldots, n
\end{aligned}
$$

The training algorithm now has a **parameter** $\gamma > 0$ for which we have to choose a "good" value. $\gamma$ is usually set by *cross validation* (discussed later). Its value is fixed before we start the optimization.

## Role of $\gamma$

- Specifies the "cost" of allowing a point on the wrong side.
- If $\gamma$ is very small, many points may end up beyond the margin boundary.
- For $\gamma \to \infty$, we recover the original SVM.

# Soft-Margin SVM

## Soft-margin dual problem

The slack variables vanish in the dual problem.

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^n} \quad W(\boldsymbol{\alpha}) := \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j \tilde{y}_i \tilde{y}_j \left( \langle \tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j \rangle + \frac{1}{\gamma}\mathbb{I}\{i=j\}\right)$$

$$\text{s.t.} \quad \sum_{i=1}^{n} \tilde{y}_i \alpha_i = 0$$

$$\alpha_i \geq 0 \quad \text{for } i = 1, \ldots, n$$

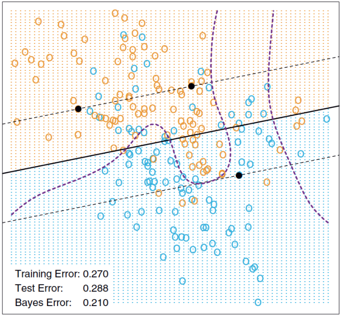
## Soft-margin classifier

The classifier looks exactly as for the original SVM:

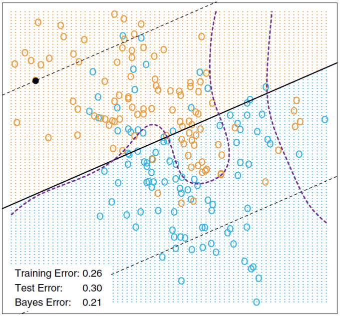
$$f(\mathbf{x}) = \operatorname{sgn}\left( \sum_{i=1}^{n} \tilde{y}_i \alpha_i^* \langle \tilde{\mathbf{x}}_i, \mathbf{x} \rangle - c \right)$$

Note: Each point on wrong side of the margin is an additional support vector ($\alpha_i^* \neq 0$), so the ratio of support vectors can be substantial when classes overlap.

# Influence of Margin Parameter




$\gamma = 100000$




$\gamma = 0.01$

Changing $\gamma$ significantly changes the classifier (note how the slope changes in the figures). We need a method to select an appropriate value of $\gamma$, in other words: to learn $\gamma$ from data.