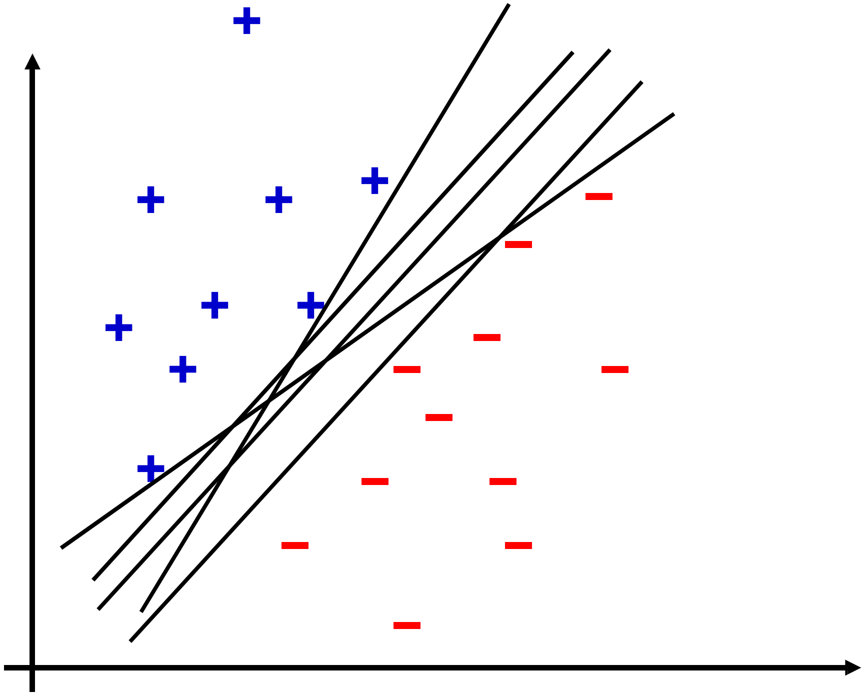
Support Vector Machines

Key concepts

- Functional and geometric margin of a classifier
- SVM objective: quadratic objective with linear constraints
- Primal and Dual problem, the KKT conditions
- Solution characteristics of SVM
- Support vectors
- Kernel SVM

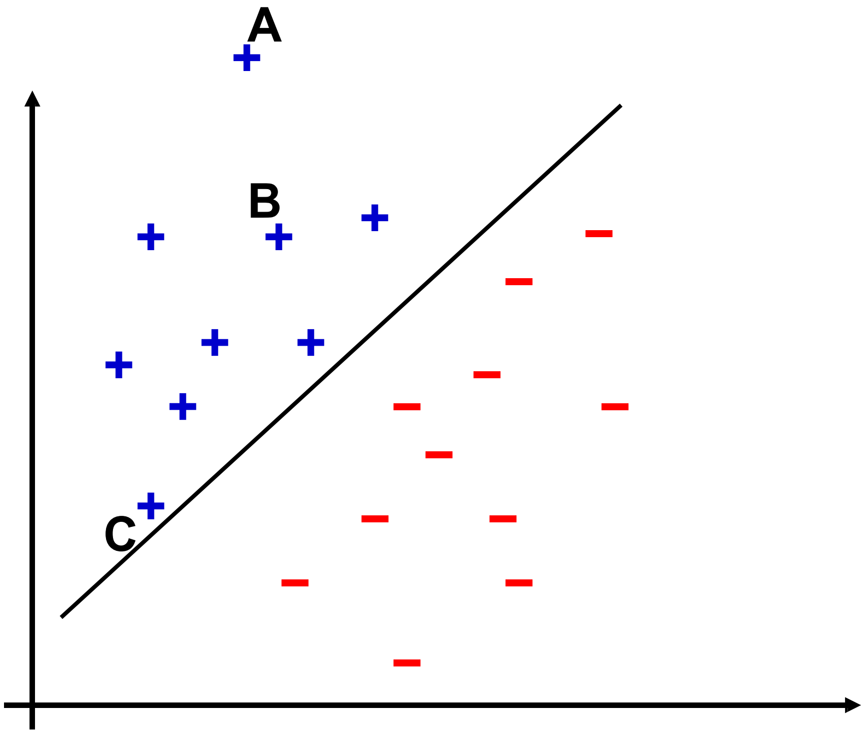
Linear Separators

- Which of the linear separators is optimal?



Intuition of Margin

- Consider points A, B, and C
- We are quite confident in our prediction for A because it is far from the decision boundary.
- In contrast, we are not so confident in our prediction for C because a slight change in the decision boundary may flip the decision.



Given a training set, we would like to make all predictions correct and confident! This leads to the concept of margin.

Functional Margin

- Given a linear classifier parameterized by $(\mathbf{w}, b)$, we define its functional margin **w.r.t training example** $(\mathbf{x}_i, y_i)$ as:

$$\hat{\gamma}_i = y_i(\mathbf{w}^T\mathbf{x}_i + b)$$

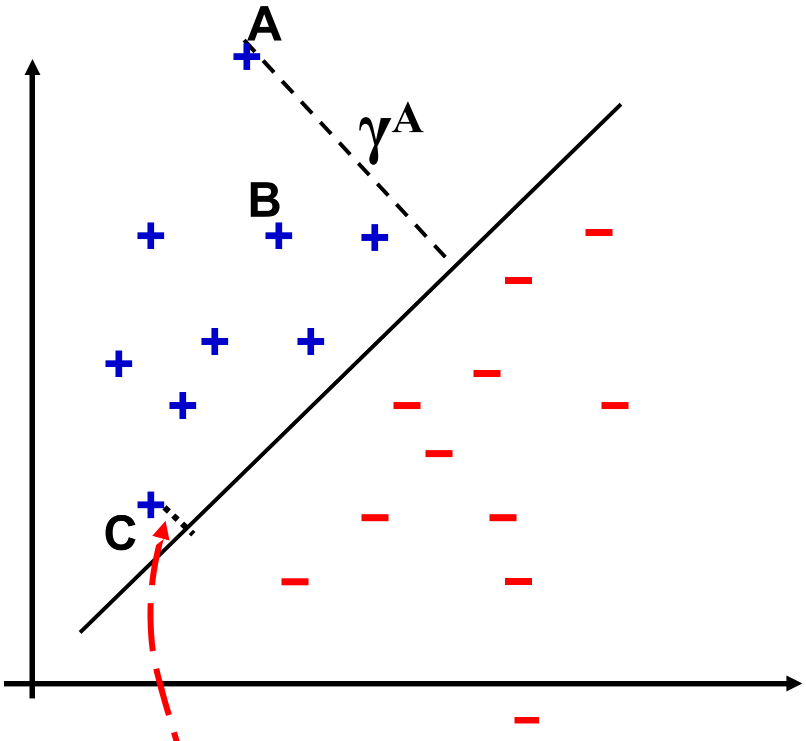
- If we rescale $(\mathbf{w}, b)$ by a factor α, functional margin gets multiplied by α
  - we can make it arbitrarily large without change anything meaningful
    - $\mathbf{w}^T\mathbf{x} + b = 0$ and $\alpha\mathbf{w}^T\mathbf{x} + \alpha b = 0$ defines the same decision boundary
  - Instead, ***geometric margin*** capture the distance between $(\mathbf{x}_i, y_i)$ and the decision boundary

Geometric Margin

- The geometric margin of $(\mathbf{w}, b)$ w.r.t. $(\mathbf{x}_i, y_i)$ is the signed distance from $\mathbf{x}_i$ to the decision boundary
  - Signed: + if point on the correct side, - otherwise

- This distance can be computed as

$$\gamma_i = \frac{y_i(\mathbf{w}^T\mathbf{x}_i + b)}{||\mathbf{w}||}$$



- Given training set $\boldsymbol{S} = \{(\mathbf{x}_i, y_i)\colon\ i = 1, \ldots, N\}$, the geometric margin of the classifier w.r.t. $\boldsymbol{S}$ is

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

Points closest to the boundary are called Support vectors – we will see that these are the points that really matters

Question time

Given a linear decision boundary defined by w and b, the functional margin of a training data point (x,y) is defined as: $y(w^T x + b)$ whereas its geometric margin is defined as $\frac{y(w^T x + b)}{|w|}$, which normalizes the functional margin by the norm of the weight vector w. Which of the following statements are true:

A. Functional margin could be positive or negative. Positive indicates correct prediction and negative indicates incorrect prediction.
B. Geometric margin could be positive or negative. Positive indicates correct prediction and negative indicates incorrect prediction.
C. Consider classifying two training data point with the same classifier (aka, same w and b), if point A has a larger functional margin than point B, point A must be further away from the decision boundary.
D. Consider classifying two training data point with the same classifier (aka, same w and b), if point A has a larger geometric margin than point B, point A must be further away from the decision boundary.

Maximum Margin Classifier

- Given a ***linearly separable*** training set $\boldsymbol{S} = \{(\mathbf{x}_i, y_i)\colon\ i = 1, \ldots, N\}$, we would like to find a linear classifier with the maximum margin.
- This can be represented as an optimization problem.

$$\max_{w,b,\gamma} \gamma$$

$$\text{subject to:} \frac{y_i(\mathbf{w}^T\mathbf{x}_i + b)}{||\mathbf{w}||} \geq \gamma$$

Nasty optimization problem! Let's make it look nicer!

- Let $\gamma' = \gamma||\mathbf{w}||$, this is equivalent to

$$\boxed{\begin{gathered}\max_{\mathbf{w},b,\gamma'} \frac{\gamma'}{||\mathbf{w}||} \\ \text{subject to}: y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq \gamma' \ \ \forall i = 1, \ldots, N\end{gathered}}$$

Maximum Margin Classifier

- Note that rescaling **w** and *b* (by $\frac{1}{\gamma'}$) will not change the classifier, we can thus further reformulate the optimization problem

$$\boxed{\begin{array}{c}\max\limits_{\mathbf{w},b,\gamma'} \dfrac{\gamma'}{\|\mathbf{w}\|} \\ \text{subject to}: \;\; y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq \gamma', \, i = 1, \ldots, N\end{array}}$$

⇩

$$\boxed{\begin{array}{l}\max\limits_{\mathbf{w},b} \dfrac{1}{\|\mathbf{w}\|} \;\; (\text{or equivalently } \min\limits_{\mathbf{w},b} \|\mathbf{w}\|^2) \\ \text{subject to}: \;\; y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \;\; i = 1, \ldots, N\end{array}}$$

Maximizing the geometric margin is equivalent to minimizing the magnitude of **w** subject to maintaining a functional margin of at least 1

Solving the Optimization Problem

$$\min_{\mathbf{w},b} ||\mathbf{w}||^2$$

Subject to $y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, i = 1, \ldots, N$

- This is a ***quadratic optimization problem*** with linear constraints.
- A well-known class of mathematical programming problems, several (non-trivial) algorithms exist.
  - One can use any of them to solve for $\mathbf{w}$ and b
- It is useful to first formulate an equivalent dual optimization problem, which serves two purposes:
  - To show that the solution for $\mathbf{w}$ can be expressed as weighted sum of subset of training examples (aka the support vectors)
  - For applying the kernel trick for nonlinear svm

Aside: Constrained Optimization

- To solve the following optimization problem

$$\min_x . f(x) \;\; s.t. \;\; g_i(x) \leq 0 \;\;\text{ for } i = 1, \ldots, m$$

- Consider the following function known as the Lagrangian

$$\mathcal{L}(x, \alpha) = f(x) + \sum_i \alpha_i g_i(x) \;\; s.t. \;\; \alpha_i \geq 0$$

- The original optimization problem is equivalent to solving the following:

$$\min_x \max_\alpha \mathcal{L}(x, \alpha) \qquad \text{subject to } \alpha_i \geq 0$$

- By exchanging the order of min and max, we get the **dual problem**:

$$\max_\alpha \min_x \mathcal{L}(x, \alpha) \qquad \text{subject to } \alpha_i \geq 0$$

Aside: Constrained Optimization

$$\mathcal{L}(x,\alpha) = f(x) + \sum_i \alpha_i g_i(x) \ \ s.t. \ \ \alpha_i \geq 0$$

Primal : $f^* = \min_{x} \max_{\alpha \geq 0} \mathcal{L}(x,\alpha)$

Dual: $d^* = \max_{\alpha \geq 0} \min_{x} \mathcal{L}(x,\alpha)$

Let $x^*$and α^* be the optimal and dual solution respectively, $f^* = d^*$ if $f(x)$ is convex and x^* and α^* satisfy the KKT conditions:

1. $\nabla L(x^*, \alpha^*) = 0$ --- zero gradient
2. $g(x^*) \leq 0$ --- primal feasibility
3. $\alpha^* \geq 0$ --- dual feasibility
4. $\alpha^* g(x^*) = 0$ --- complementary slackness

Back to the Original Problem

$$\text{Minimize } \frac{1}{2}||\mathbf{w}||^2$$

$$\text{subject to}: 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b) \leq 0, i = 1, \dots, N$$

The Lagrangian is

$$\mathcal{L}(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + \sum_{i=1}^{N} \alpha_i(1 - y_i(\mathbf{w}^T\mathbf{x}_i + b))\ s.t., \alpha_i \geq 0$$

- We want to solve the dual problem: $\max_{\alpha\geq 0} \min_{\mathbf{w},b} \mathcal{L}(\mathbf{w}, b, \alpha)$
- Setting the gradient of $\mathcal{L}$ w.r.t. $\mathbf{w}$ and b to zero:

$$\mathbf{w} - \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i = 0 \ \Rightarrow \ \mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i$$

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

The Dual Problem

$$\mathcal{L}(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + \sum_{i=1}^{N} \alpha_i\left(1 - y_i(\mathbf{w}^T\mathbf{x}_i + b)\right)$$

- Substitute $\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i$ into $\mathcal{L}$:

$$L(\alpha)$$

$$= \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j + \sum_{i=1}^{N}\alpha_i - \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j - b\boxed{\sum_{i=1}^{N}\alpha_i y_i}$$

$$= 0$$

$$= \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j$$

The Dual Problem

- The new objective function is in terms of α_i, known as the dual problem
- The original problem is known as the primal problem
- The objective function of the dual problem needs to be maximized!
- The dual problem is therefore:

$$\max L(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\text{subject to} \quad \alpha_i \geq 0, i = 1, \dots, N \qquad \sum_{i=1}^{N} \alpha_i y_i = 0$$

Properties of α_i when we introduce the Lagrange multipliers

The result when we differentiate the original Lagrangian w.r.t. b

The Dual Problem

$$\max L(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\text{subject to} \quad \alpha_i \geq 0, i = 1, \dots, N \qquad \sum_{i=1}^{N} \alpha_i y_i = 0$$

- This is also a quadratic programming (QP) problem
  - A global maximum of α_i can always be found
- **w** can be recovered by $\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i$
- b can also be recovered as well (wait for a bit)

Characteristics of the Solution

- Many of the α_i are zero --- sparse solution
- **w** is a linear combination of only a small number of data points
- The KKT conditions requires that:

$$\alpha_i \geq 0, i = 1, \ldots, N$$

Dual feasibility

$$y_i \left(\sum_{j=1}^{N} \alpha_j y_j \mathbf{x}_j^T \mathbf{x}_i + b \right) \geq 1 \, , i = 1, \ldots, N$$

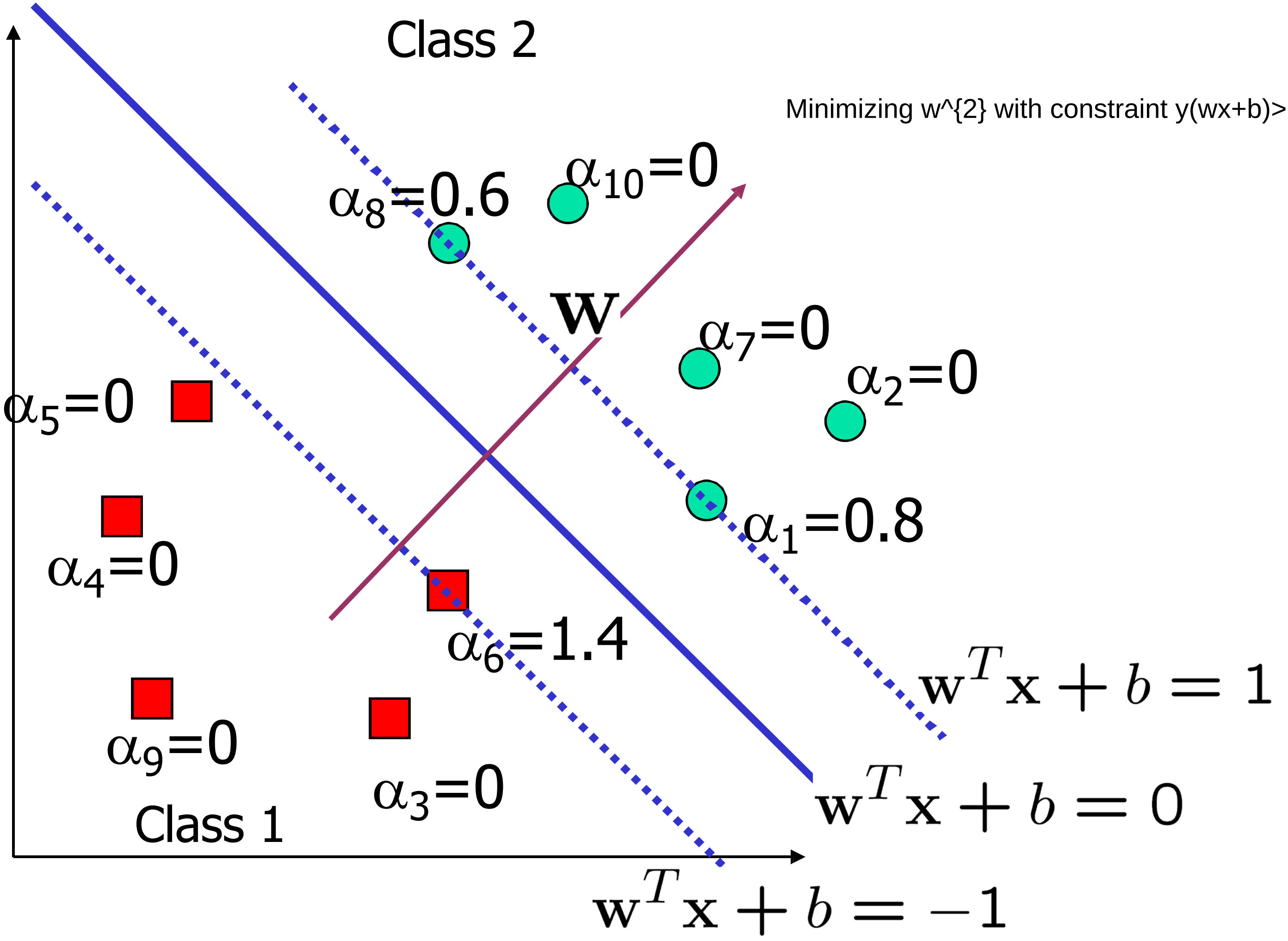
Primal feasibility: Functional margin ≥ 1

$$\alpha_i \left(y_i \left(\sum_{j=1}^{N} \alpha_j y_j \mathbf{x}_j^T \mathbf{x}_i + b \right) - 1 \right) = 0, i = 1, \ldots, N$$

Complemetary slackness: α is nonzero only when functional margin = 1

A Geometrical Interpretation

Minimizing w^{2} with constraint y(wx+b)>=1 will keep minimizir



Support Vectors

- $\mathbf{x}^i$ with non-zero $\alpha' s$ are called support vectors (SV)
- The decision boundary is determined only by the SV's

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i$$

- Note that we know that for support vectors the functional margin = 1
- We can use this information to solve for b

Classifying new examples

For classifying with a new input **x**

- Compute $$\mathbf{w}^T\mathbf{x} + b = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + b$$

- Note: no need to form **w** explicitly, rather, classify **x** by taking a weighted sum of **its dot products with the support vectors** (useful for generalizing from inner product to kernels)

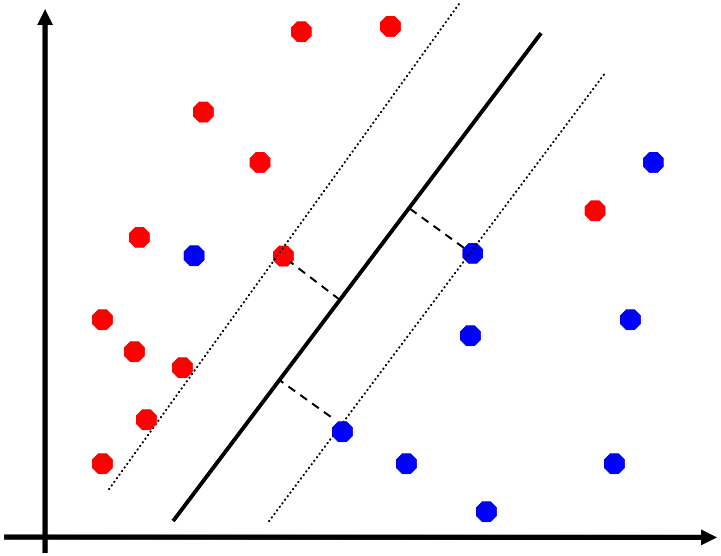
Solving the QP optimization problem

- Many approaches have been proposed for QP
  - Loqo, cplex, etc. (see http://www.numerical.rl.ac.uk/qp/qp.html)
- Early work focuses on “interior-point” methods
  - Start with an initial solution that can violate the constraints
  - Improve this solution by optimizing the objective function and/or reducing the amount of constraint violation
- Stochastic sub-gradient descent has been shown to lead to extremely efficient primal solver for large scale problems
- In practice, one can just regard the QP solver as a “black-box” without bothering how it works, but depending on the scale of the problem some solvers might be more appropriate than others

Non-separable Data

What if the data is not linearly separable?

- The solution does not exist
- i.e., the set of linear constraints are not satisfiable
- But we should still be able to find a good decision boundary



Solution:

- Project the data onto higher dimensional space
- Via kernel function

Kernel SVM

Linear SVM:

$$\max L(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\text{subject to} \quad \alpha_i \geq 0, i = 1, \ldots, N, \qquad \sum_{i=1}^{N} \alpha_i y_i = 0$$

Replace dot product with kernel function

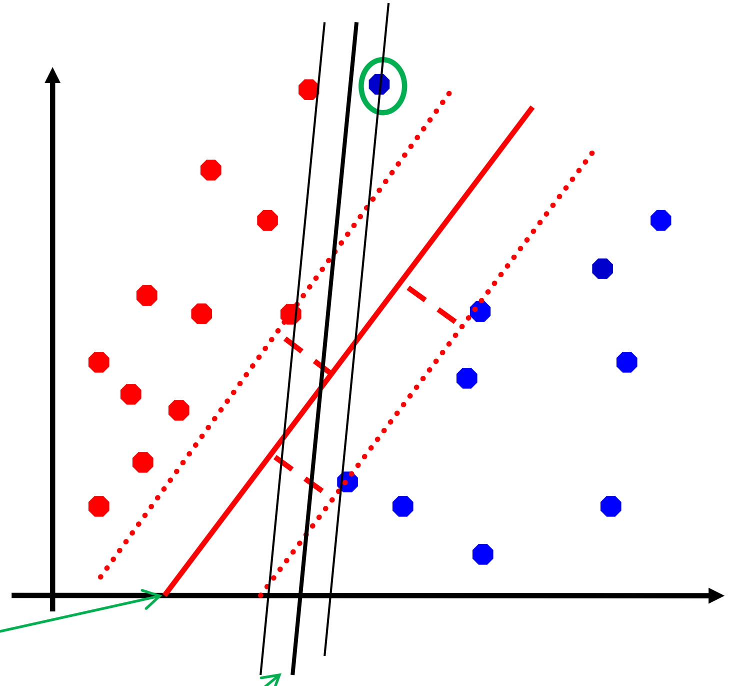
Kernel SVM:

$$\max L(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \kappa(\mathrm{x}_i, \mathbf{x}_j)$$

$$\text{subject to} \quad \alpha_i \geq 0, i = 1, \ldots, N, \qquad \sum_{i=1}^{N} \alpha_i y_i = 0$$

Maximum margin overfits to outliers

Consider the blue point circled out. It is an outlier that is labeled as blue but really should belong to red

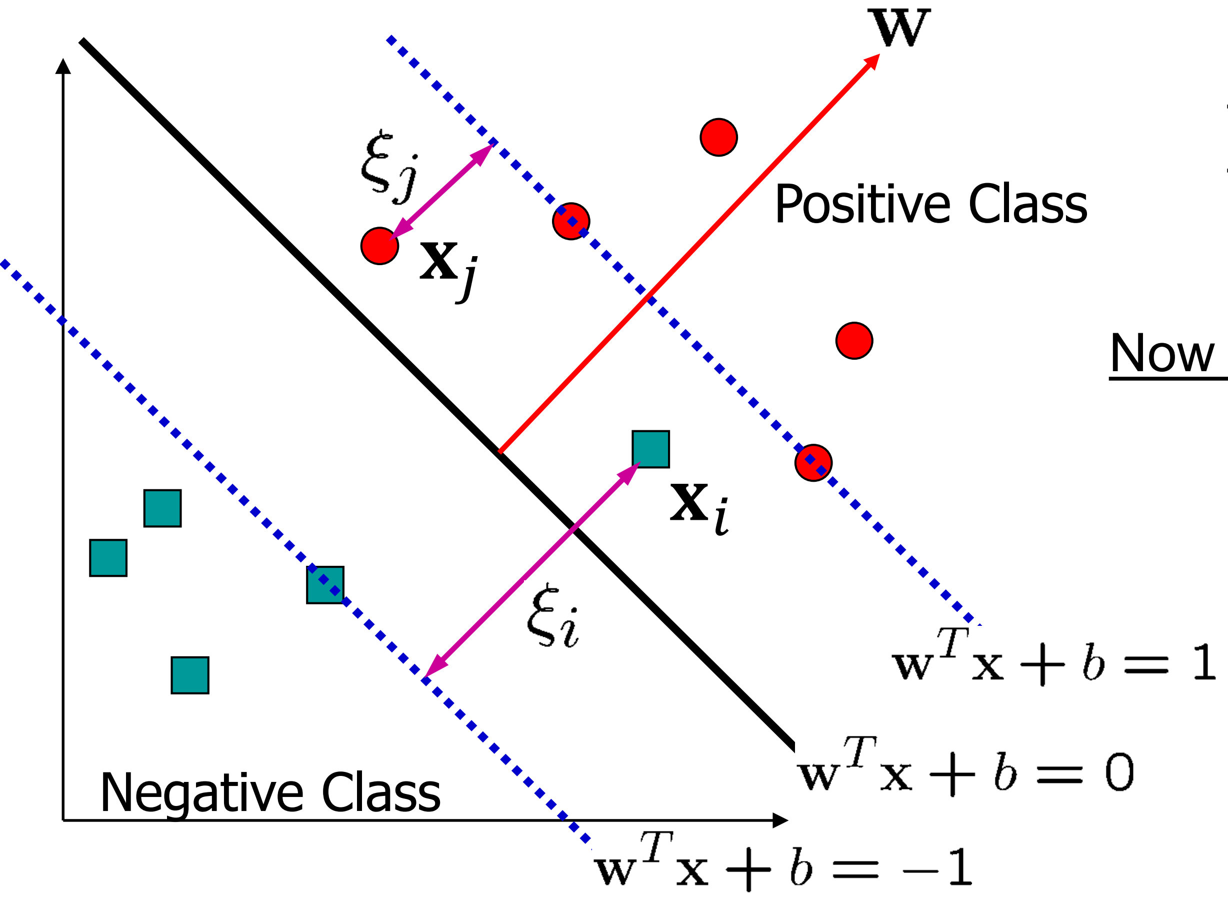


We would like to learn a boundary that ignores the outliers

But the margin will be defined by the outlier and we instead learn a boundary that overfit to the outliers

Soft Margin

- Allow functional margins to be less than 1


$\mathbf{w}$
ξ_j
$\mathbf{x}_j$
Positive Class
$\mathbf{x}_i$
ξ_i
$\mathbf{w}^T\mathbf{x} + b = 1$
$\mathbf{w}^T\mathbf{x} + b = 0$
Negative Class
$\mathbf{w}^T\mathbf{x} + b = -1$

Originally functional margins need to satisfy:

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$$

Now we allow it to be less than 1:

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i$$
$$\xi_i \geq 0$$

The objective changes to:

$$\min_{\mathbf{w},b,\xi_i} \|\mathbf{w}\|^2 + c\sum_{i=1}^{N}\xi_i$$

Soft-Margin Maximization

$$\min_{\mathbf{w},b} \|\mathbf{w}\|^2$$
$$\text{subject to}: \; y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \quad i = 1, \cdots, N$$

Slack variables

$$\min_{\mathbf{w},b} \|\mathbf{w}\|^2 + c\sum_{i=1}^{N} \xi_i$$
$$\text{subject to}: \; y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1, \cdots, N$$
$$\xi_i \geq 0, \quad i = 1, \cdots, N$$

- This allows some functional margins < 1 (could even be < 0)
- The ξ_i 's can be viewed as the “errors” of our *fat* decision boundary
- Adding ξ_i 's to the objective function to minimize errors
- We have a tradeoff between making the decision boundary fat and minimizing the error
- Parameter ***c*** controls the tradeoff:
  - Large c: ξ_i's incur large penalty, so the optimal solution will try to avoid them
  - Small c: small cost for ξ_i's, we can sacrifice some training examples to have a large classifier margin

Soft Margin SVM: Regularized Hinge loss

$$\min_{\mathbf{w},b} ||\mathbf{w}||^2 + c \sum_{i=1}^{N} \xi_i$$

$$\text{subject to } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i,$$

$$\xi_i \geq 0, \forall i = 1, \dots, N$$

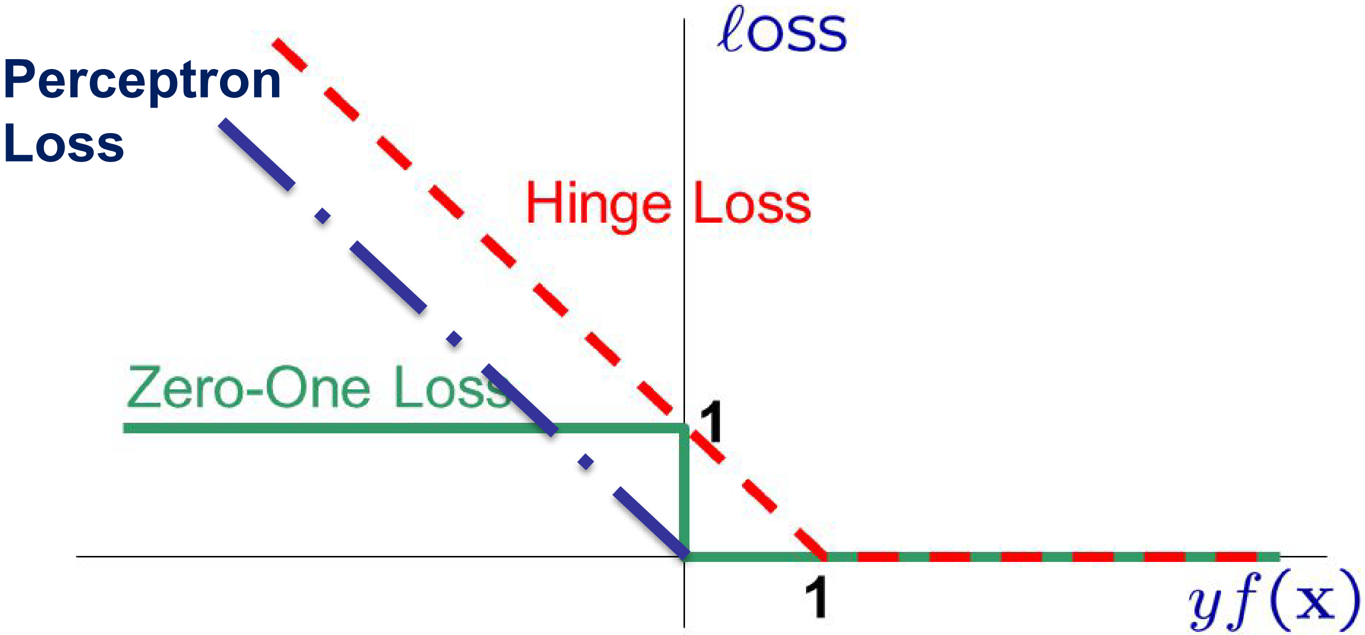
Is equivalent to:

$$\min_{w,b} ||\mathbf{w}||^2 + \cancel{c} \sum_{i}^{N} \underbrace{\max\big(0,1 - y_i(\mathbf{w}^T\mathbf{x}_i + b)\big)}_{\text{Hinge loss}}$$

λ

L_2 Regularization

Different Loss functions



Solutions to soft-margin SVM

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i, \quad \text{s.t.} \sum_{i=1}^{N} \alpha_i y_i = 0$$

No soft margin

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i$$

$$\text{s.t.} \sum_{i=1}^{N} \alpha_i y_i = 0 \ \text{ and } 0 \le \alpha_i \le c \ \forall i = 1, \ldots, N$$

With soft margin

- *c* effectively puts a **box constraint** on α, the weights of the support vectors
- It limits the influence of individual support vectors (maybe outliers)
- In practice, c is a hyper-parameter to be tuned and can be set using cross-validation

Question time

As we change the parameter C for soft-margin SVM, what do you expect to happen to the number of support vectors?

A. Larger C value leads to larger number of support vectors.

B. Larger C value leads to smaller number of support vectors.

C. If C=0, all points become support vectors.

D. If C=∞ , all points become support vectors.

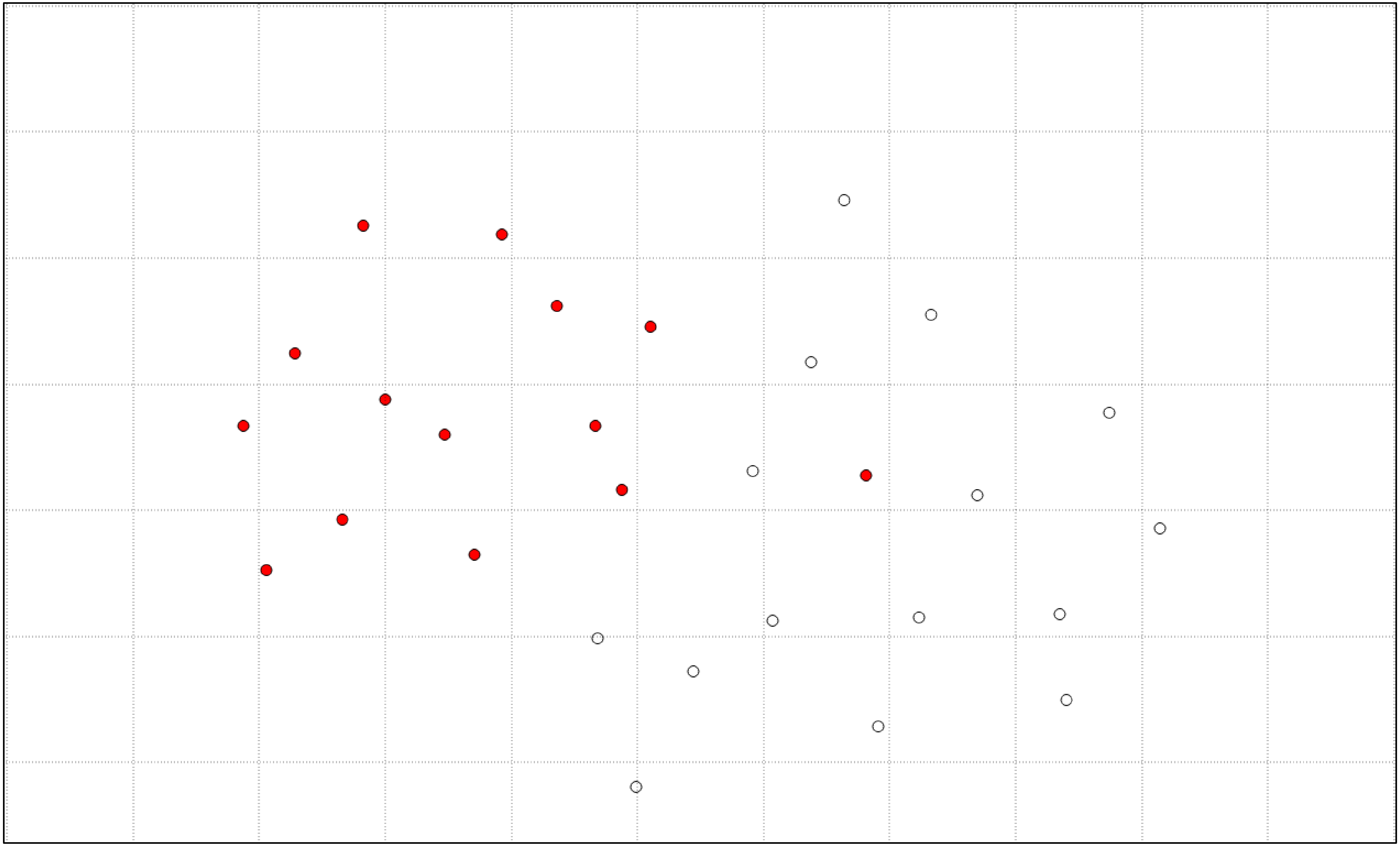
Kernel SVM with soft margin

$$\max L(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \kappa(\mathrm{x}_i, \mathbf{x}_j)$$

$$\text{subject to} \quad 0 \le \alpha_i \le c, \forall i = 1,\ldots,N, \quad \sum_{i=1}^{N} \alpha_i y_i = 0$$

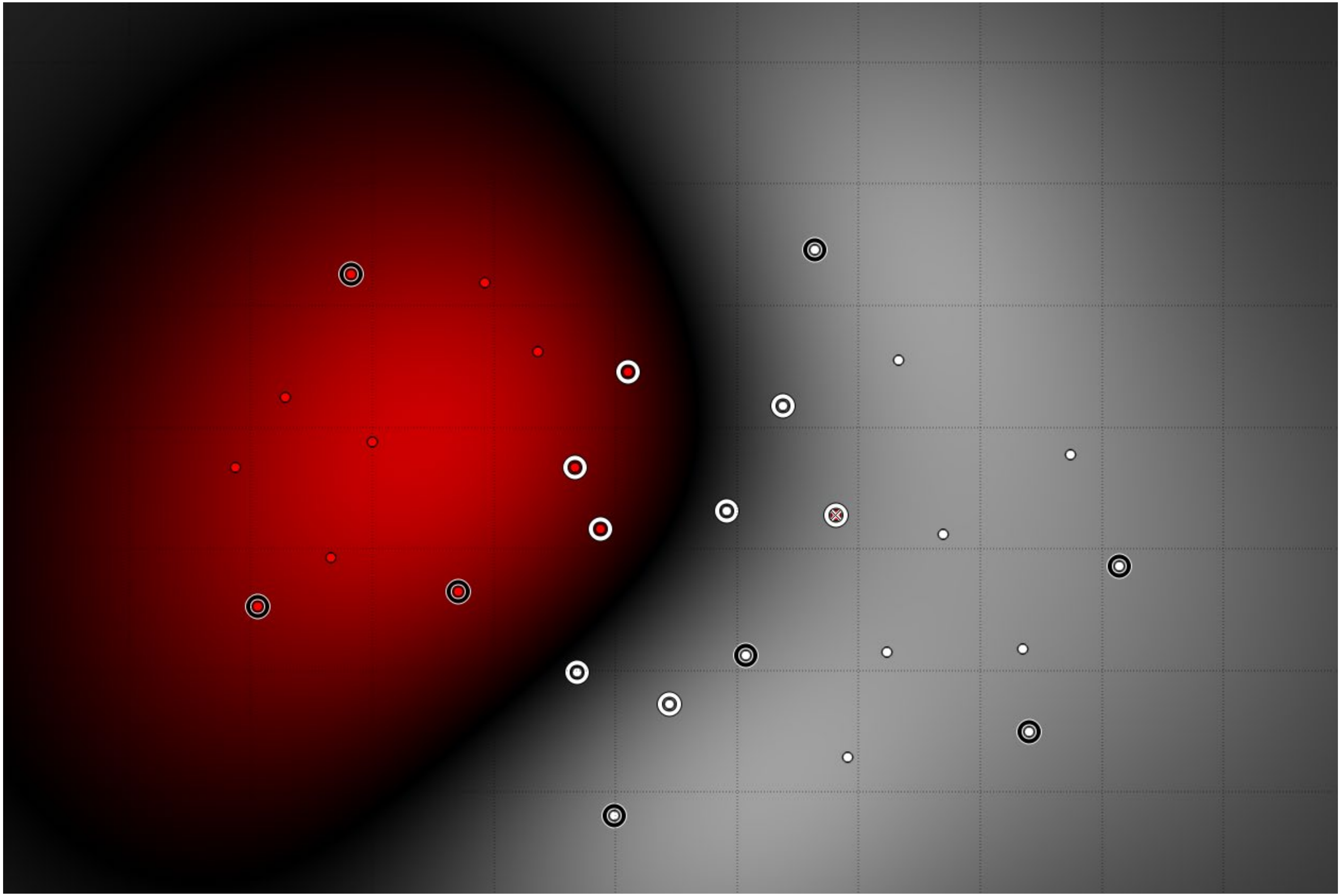
Summary of SVM

- SVM aims to find the max margin linear separator
  - Geometric margin vs functional margin
- Solution of SVM is a weighted combination of training examples
  - Most weights are zero. None-zero ones are called support vectors
- Hard Margin SVM issues:
  - has no solution for nonlinearly separable data
  - Overfits to outliers
- Soft margin SVM can be interpreted as:
  - Introducing slack to the hard margin constraints
  - Minimizing $L2$ regularized hinge loss
- Parameter C controls the trade off between fitting the data (ie. small slack) and having large margin (i.e., small $||\mathbf{w}||^2$)
  - It introduces a box bound on the α values
  - Large C (or equiv. small λ) increases overfitting, decreases # of support vectors
  - Small C (or equiv. large λ) reduces overfitting, increase # of support vectors
- Applying the kernel trick, we can learn max margin separator in the mapped nonlinear space
  - Different kernels lead to different decision boundaries

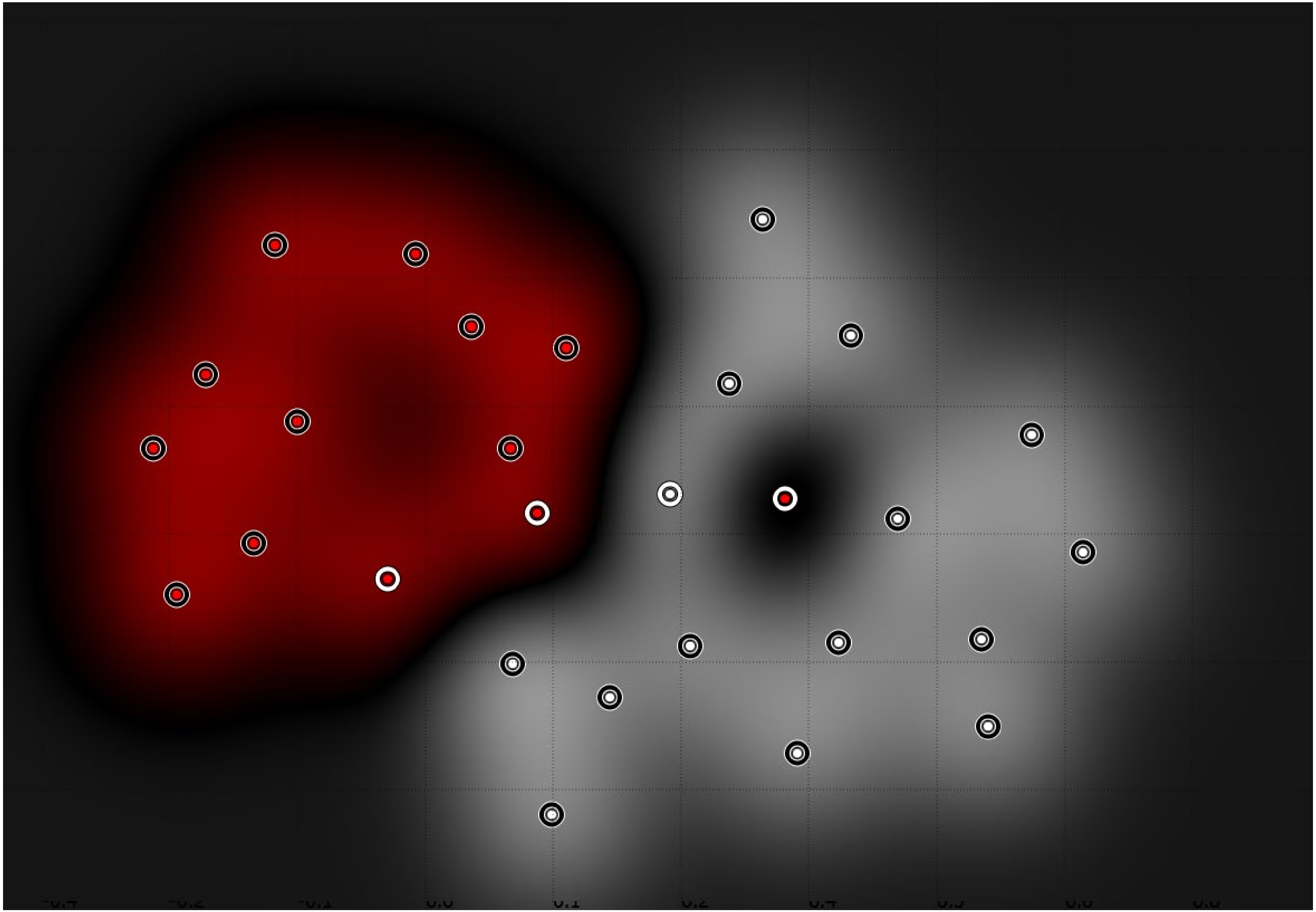


SVM demo

RBF kernel, $\sigma = 0.1, \mathrm{c} = 1$



RBF Kernel $\sigma = 0.01, c = 1$



RBF kernel $\sigma = 0.01, c = 10$