Today

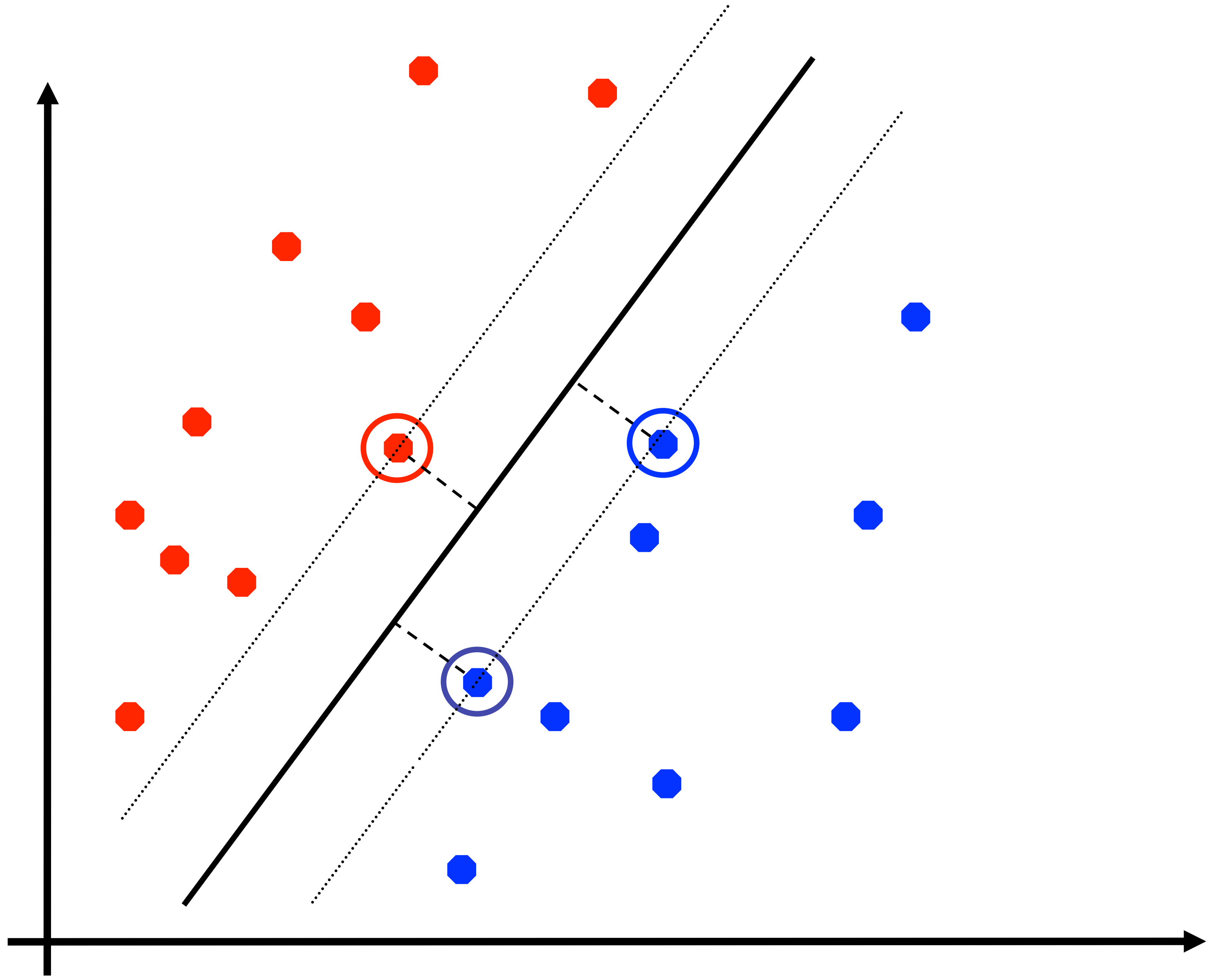
- Last time
  - Towards support vector machines
    - Perceptron algorithm
  - Optimal margin classifiers

- Today
  - Dual formulation of SVMs
    - Bishop Chapter 7
    - Hastie Chapter 12
    - http://cs229.stanford.edu/notes/cs229-notes3.pdf

- Announcements
  - Blended lab session on Wed (make sure you know your seat number)
  - Homework 1 due on Wed *before class*

Support vector machines

- SVMs (Vapnik, 1990's) choose the linear separator with the **largest margin**



- Good according to intuition, theory, practice

[David Sontag]

Notation

- Class labels $y \in \{-1,1\}$
- Parametrize with w, b rather than θ (intercept treated separately)

$$h_\theta(x) = g(w^T x + b)$$

with

$$g(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases}$$

Functional margin

- Define functional margin of training sample $(x^{(i)}, y^{(i)})$ w.r.t. (w, b) as
$$\hat{\gamma}^{(i)} = y^{(i)}(w^T x + b)$$
  - If $y^{(i)} = 1$, then need $w^T x + b \gg 0$ for $\hat{\gamma}^{(i)}$ large
  - If $y^{(i)} = -1$, then need $w^T x + b \ll 0$ for $\hat{\gamma}^{(i)}$ large
  - Holds $y^{(i)}(w^T x + b) > 0$, then prediction correct
- Large functional margin = confident + correct prediction
- Scaling
  - For our choice $g(z) = 1 ? z \geq 0 : 0$ have
$$g(2w^T x + 2b) = g(w^T x + b)$$
which means that h_θ is invariant under scaling even though $\hat{\gamma}^{(i)}$ is not
—> normalize by enforcing $\|\boldsymbol{w}\| = 1$

Geometric margin

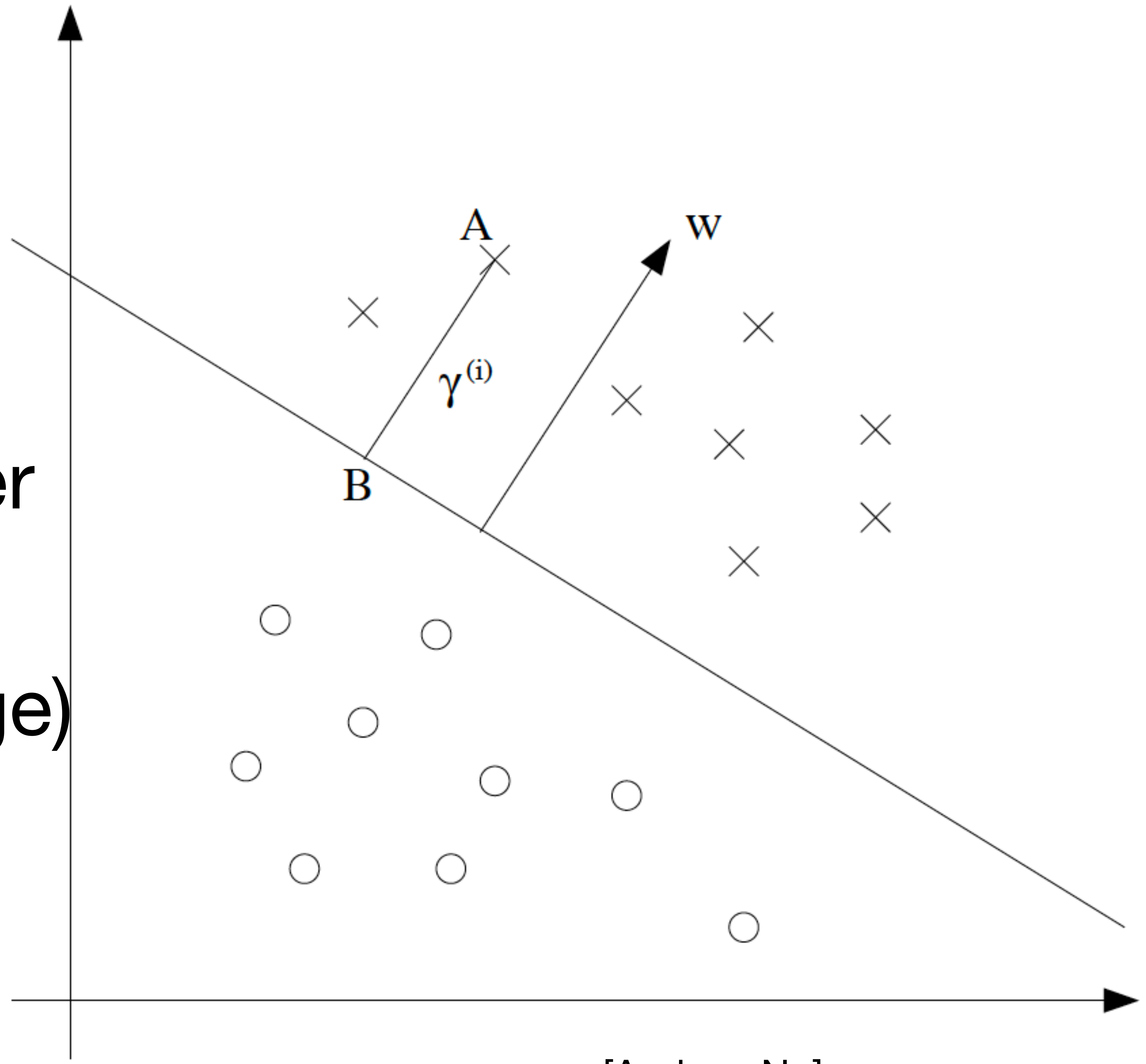
- Geometric margin of (w, b) w.r.t. $(x^{(i)}, y^{(i)})$

$$\gamma^{(i)} = y^{(i)} \left(\frac{w^T}{||w||} x^{(i)} + \frac{b}{||w||} \right)$$

- Geometric margin is invariant under scaling of w, b (e.g., replace w, b with $2w, 2b$ then $\gamma^{(i)}$ doesn't change)

- Geometric margin w.r.t. set $\mathscr{D}$

$$\gamma = \min_{i=1,\ldots,N} \gamma^{(i)}$$



[Andrew Ng]

Optimal margin classifier

- Find decision boundary that maximizes geometric margin
- **Assumption: Training set $\mathscr{D}$ is linearly separable**
- Pose optimization problem

$$
\begin{aligned}
\max_{\gamma,w,b} \quad & \gamma \\
\textbf{s.t.} \quad & y^{(i)}(w^T x^{(i)} + b) \geq \gamma\,, \quad i = 1,\ldots,N \\
& \|w\| = 1
\end{aligned}
$$

- Constraint $\|w\| = 1$ ensures that functional margin $((\hat{\gamma}^{(i)} =)y^{(i)}(w^T x^{(i)} + b))$ is equal to geometric margin
- Constraint $\|w\| = 1$ is non-convex (nasty to optimize)

Optimal margin classifier (cont'd)

- Note that functional margin $\hat{\gamma}$ and geometric margin γ are related as

$$\gamma = \hat{\gamma}/||w||$$

- Optimize normalized functional margin

$$\max_{\hat{\gamma},w,b} \quad \frac{\hat{\gamma}}{||w||}$$

$$\textbf{s.t.} \quad y^{(i)}(w^T x^{(i)} + b) \geq \hat{\gamma}, \quad i = 1,\ldots,N$$

- Got rid of constraint $||w|| = 1$ but introduced objective $\hat{\gamma}/||w||$

Optimal margin classifier (cont'd)

- Invoke that functional margin $\hat{\gamma}$ depends on scaling
  - Multiplying w, b by constant, multiplies $\hat{\gamma}$ by that constant
- Introducing constraint $\hat{\gamma} = 1$, which indeed is a scaling constraint on w, b and obtain

$$
\begin{aligned}
\min_{w,b} \quad & \frac{1}{2}\|w\|^2 \\
\textbf{s.t.} \quad & y^{(i)}(w^T x^{(i)} + b) \geq 1\,, \quad i = 1,\ldots,N
\end{aligned}
$$

- Note: maximizing $\hat{\gamma}/\|w\|$ (with $\hat{\gamma} = 1$) is same as minimizing $\|w\|^2$
- Convex quadratic objective, linear constraints
- The solution is the optimal margin classifier

Why is this called “support vector machines”? - Dual formulation of SVMs

board

Support vector

- Write constraints of problem

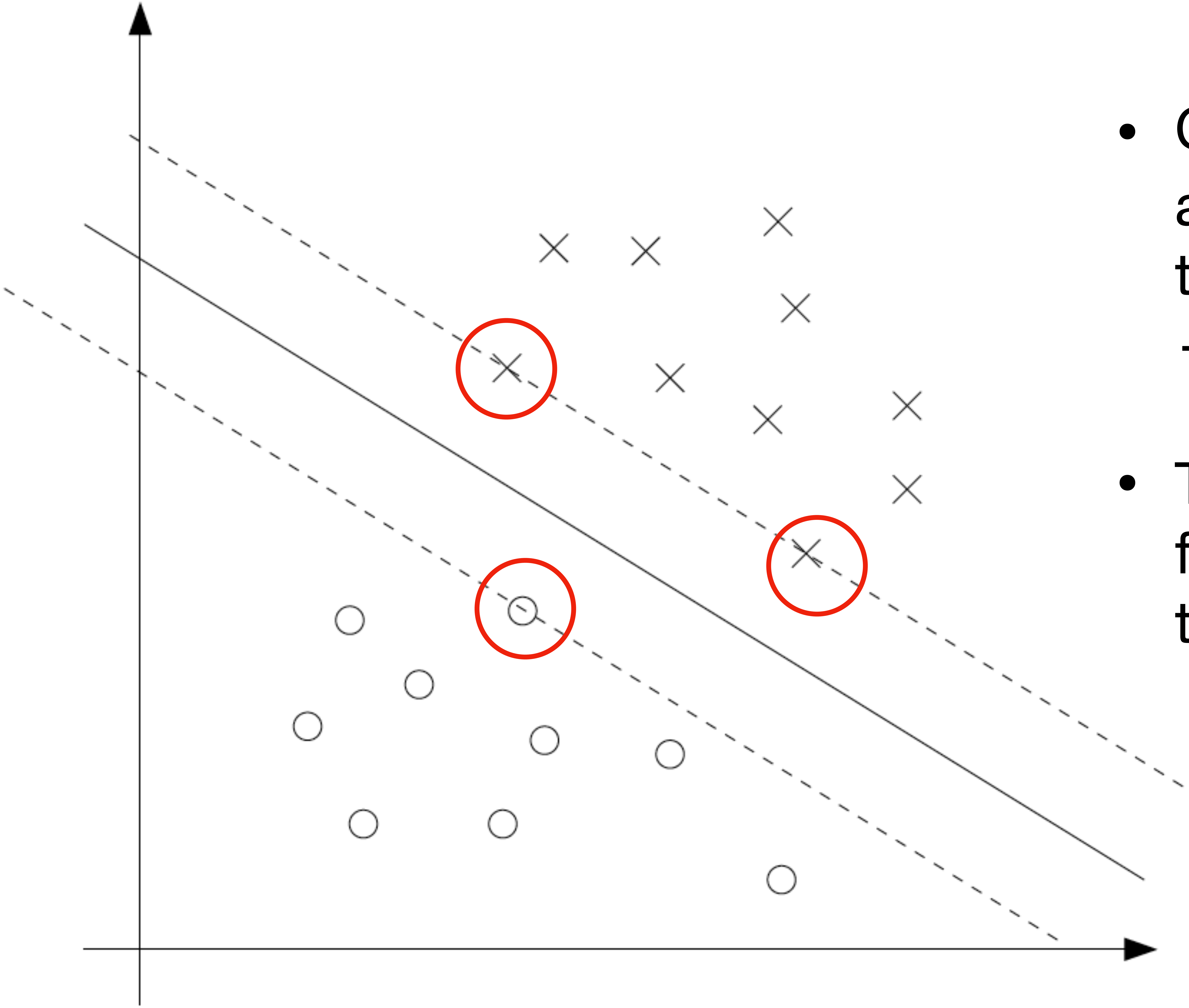
$$\min_{w,b} \quad \frac{1}{2}\|w\|^2$$

$$\text{s.t.} \quad y^{(i)}(w^T x^{(i)} + b) \geq 1\,, \quad i = 1,\ldots,N$$

as

$$g_i(w) = -y^{(i)}(w^T x^{(i)} + b) + 1 \leq 0$$

- Know from KKT conditions that g_i active for $\alpha_i^* > 0$
- Corresponds to training points with functional margin $\hat{\gamma}^{(i)} = 1$
- These training points are called the **support vectors**



- Constraint g_i active for few training points only —> support vectors
- Typically have much fewer support vectors than training points

Dual formulation

- Lagrangian of our optimization problem is

$$\mathscr{L}(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \alpha_i \left(y^{(i)}(w^T x + b) - 1 \right)$$

- Dual problem is

$$\max_{\alpha} \quad \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} y^{(i)} y^{(j)} \alpha_i \alpha_j < x^{(i)}, x^{(j)} >$$

$$\text{s.t.} \quad \alpha_i \geq 0 \,, i = 1, \ldots, N$$

$$\sum_{i=1}^{N} \alpha_i y^{(i)} = 0$$

- Can solve dual instead of primal problem

Dual solution $\iff$ primal solution

- In the derivation of the dual problem, we obtained

$$w = \sum_{i=1}^{N} \alpha_i y^{(i)} x^{(i)}$$

- If we solved the dual to obtain $\alpha_1^*, \ldots, \alpha_N^*$, we can find w^* with the equation above

- The optimal value b^* of the intercept term is

$$b^* = -\frac{\max_{i, y^{(i)}=-1} (w^*)^T x^{(i)} + \min_{i, y^{(i)}=1} (w^*)^T x^{(i)}}{2}$$

Prediction with SVMs

- Suppose found $w*$ via $\alpha*$
- Naive: Compute $(w*)^T x + b$ and assign $y = 1$ if positive and $y = -1$ otherwise
- Write $(w*)^T x + b$ in terms of $\alpha*$

$$(w*)^T x + b = \left(\sum_{i=1}^{N} \alpha_i^* y^{(i)} x^{(i)} \right)^T x + b$$

$$= \sum_{i=1}^{N} \alpha_i^* y^{(i)} < (x^{(i)})^T, x > + b$$

- What structure does this formulation reveal?

Prediction with SVMs (cont'd)

$$(w^*)^T x + b = \left(\sum_{i=1}^{N} \alpha_i^* y^{(i)} x^{(i)}\right)^T x + b$$

$$= \sum_{i=1}^{N} \alpha_i^* y^{(i)} < (x^{(i)})^T, x > + b$$

- Only the inner product with training data $x^{(i)}$ required
- Additionally
  - The α_i^* are all 0 except for the (typically, few) support vectors
  - Thus, need only the support vector to make prediction (storage)

Inner product and SVMs

Solve:

$$\max_{\alpha} \quad \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} y^{(i)} y^{(j)} \alpha_i \alpha_j < x^{(i)}, x^{(j)} >$$

s.t. $\quad \alpha_i \geq 0\,, i = 1, \ldots, N$

$$\sum_{i=1}^{N} \alpha_i y^{(i)} = 0$$

Predict:

$$\sum_{i=1}^{N} \alpha_i^* y^{(i)} < (x^{(i)})^T, x > + b$$

- Solve and predict “touch” training data only via inner products
- This is key for using SVMs with kernels