EN.600.475 Machine Learning

Support Vector Machines

Raman Arora

Lecture 11

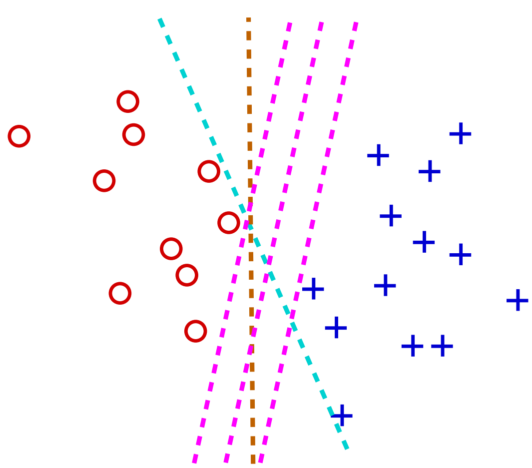
March 1, 2017

- Large margin classifiers
- SVMs with slack, kernels

Slides credit: Greg Shakhnarovich

Optimal linear classifier

- Which decision boundary is better?

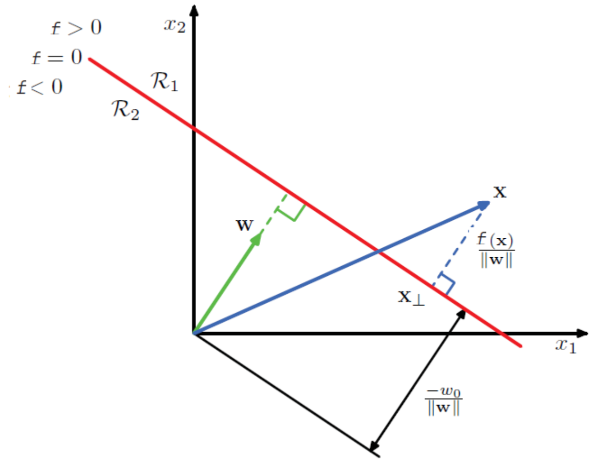


- Regularization alone does not capture this intuition

Classification margin

- Recall the geometry of linear classification:
- Discriminant function:

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + w_0$$



- Distance from a *correctly* classified $(\mathbf{x}, y)$ to the boundary:

$$\frac{1}{\|\mathbf{w}\|} y\left(\mathbf{w} \cdot \mathbf{x} + w_0\right)$$

- Important: the distance does not change if we scale $\mathbf{w} \to a\mathbf{w}$, $w_0 \to aw_0$

Large margin classifier

- Distance from a *correctly* classified $(\mathbf{x}, y)$ to the boundary:

$$\frac{1}{\|\mathbf{w}\|} y\left(\mathbf{w} \cdot \mathbf{x} + w_0\right)$$

- Margin of the classifier on $X = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, assuming it achieves 100% accuracy: the distance to the closest point

$$\min_i \frac{1}{\|\mathbf{w}\|} y_i\left(\mathbf{w} \cdot \mathbf{x}_i + w_0\right)$$

- We are interested in a large margin classifier:

$$\operatorname*{argmax}_{\mathbf{w}, w_0} \left\{ \frac{1}{\|\mathbf{w}\|} \min_i y_i\left(\mathbf{w} \cdot \mathbf{x}_i + w_0\right) \right\}$$

Optimal separating hyperplane

- So, we seek $\operatorname{argmax}_{\mathbf{w},w_0} \left\{ \frac{1}{\|\mathbf{w}\|} \min_i y_i \left(\mathbf{w} \cdot \mathbf{x}_i + w_0\right) \right\}$
- Hard optimization problem. . . but: we can set

$$\min_i y_i \left(\mathbf{w} \cdot \mathbf{x}_i + w_0\right) = 1,$$

since can rescale $\|\mathbf{w}\|$, w_0 appropriately.

- Then, the optimization becomes:

$$\begin{aligned} &\operatorname*{argmax}_{\mathbf{w},w_0} \quad \frac{1}{\|\mathbf{w}\|} \quad \text{s.t. } y_i \left(\mathbf{w} \cdot \mathbf{x}_i + w_0\right) \geq 1,\ \forall i = 1, \dots, N. \\ \Rightarrow &\operatorname*{argmin}_{\mathbf{w}} \quad \|\mathbf{w}\|^2 \quad \text{s.t. } y_i \left(\mathbf{w} \cdot \mathbf{x}_i + w_0\right) \geq 1,\ \forall i = 1, \dots, N. \end{aligned}$$

Margin and regularization

- In general d-dimensional case, we solve the regularization problem:

$$\text{minimize} \qquad \|\mathbf{w}\|^2 = \sum_{j=1}^{d} w_j^2,$$

subject to the margin constraint

$$\forall i, \qquad y_i(w_0 + \mathbf{w} \cdot \mathbf{x}_i) - 1 \geq 0.$$

Lagrange multipliers

$$\min_{\mathbf{w}} \ \frac{1}{2}\|\mathbf{w}\|^2 \ = \ \frac{1}{2}\sum_{j=1}^{d} w_j^2,$$

subject to $y_i(w_0 + \mathbf{w}\cdot\mathbf{x}_i) - 1 \ \geq \ 0, \quad i = 1, \ldots, N.$

- We will associate with each constraint the loss

$$\max_{\alpha_i \geq 0} \alpha_i \left[1 - y_i(w_0 + \mathbf{w}\cdot\mathbf{x}_i)\right] = \begin{cases} 0, & \text{if } y_i\left(w_0 + \mathbf{w}\cdot\mathbf{x}_i\right) - 1 \geq 0, \\ \infty & \text{otherwise (constraint violated).} \end{cases}$$

- We can reformulate our problem now:

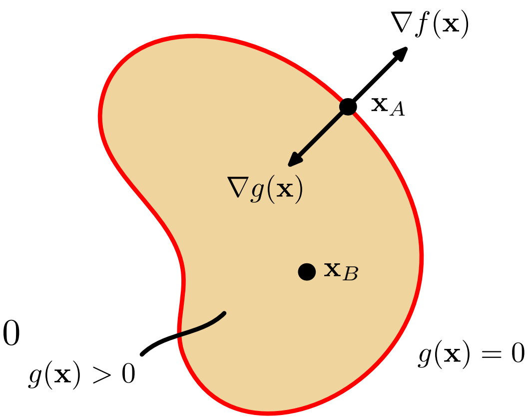
$$\min_{\mathbf{w}} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{N} \max_{\alpha_i \geq 0} \alpha_i \left[1 - y_i(w_0 + \mathbf{w}\cdot\mathbf{x}_i)\right] \right\}$$

Lagrange multipliers

- Constrained optimization problem:

$$\mathbf{x}^* \ = \ \operatorname*{argmax}_{\mathbf{x}} f(\mathbf{x})$$

s.t. $g(\mathbf{x}) \ \geq \ 0$



- $\mathbf{x}_B$: inactive constraint, $g(\mathbf{x}_B) \ > \ 0$
- $\mathbf{x}_A$: active constraint, $g(\mathbf{x}_A) \ = \ 0$
- We must have

$$\nabla f(\mathbf{x}_A) \ = \ -\lambda \nabla g(\mathbf{x}_A) \qquad \text{for some } \lambda \ > \ 0$$

KKT conditions

- Karush-Kuhn-Tucker conditions: solution to

$$\max_{\mathbf{x}} f(\mathbf{x}) \quad \text{s.t. } g(\mathbf{x}) \geq 0$$

is equivalent to solution of

$$\min_{\lambda} \max_{\mathbf{x}} \{f(\mathbf{x}) + \lambda g(\mathbf{x})\}$$

subject to

$$\begin{aligned} g(\mathbf{x}) &\geq 0, \\ \lambda &\geq 0, \\ \lambda g(\mathbf{x}) &= 0 \end{aligned}$$

Max-margin optimization

- We want all the constraint terms to be zero:

$$\begin{aligned} & \min_{\mathbf{w}} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{N} \max_{\alpha_i \geq 0} \alpha_i \left[1 - y_i(w_0 + \mathbf{w}\cdot\mathbf{x}_i)\right] \right\} \\ = & \min_{\mathbf{w}} \max_{\boldsymbol{\alpha}\geq 0} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{N} \alpha_i \left[1 - y_i(w_0 + \mathbf{w}\cdot\mathbf{x}_i)\right] \right\} \\ = & \max_{\boldsymbol{\alpha}\geq 0} \min_{\mathbf{w}} \underbrace{\left\{ \frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{N} \alpha_i \left[1 - y_i(w_0 + \mathbf{w}\cdot\mathbf{x}_i)\right] \right\}}_{J(\mathbf{w},w_0;\boldsymbol{\alpha})}. \end{aligned}$$

- Why could we switch $\min$ and $\max$? convexity!

Strategy for optimization

- We need to find

$$\max_{\boldsymbol{\alpha}\geq 0}\min_{\mathbf{w}} \underbrace{\left\{\frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{N}\alpha_i\left[1 - y_i(w_0 + \mathbf{w}\cdot\mathbf{x}_i)\right]\right\}}_{J(\mathbf{w},w_0;\boldsymbol{\alpha})}.$$

- We will first fix $\boldsymbol{\alpha}$ and treat $J(\mathbf{w}, w_0; \boldsymbol{\alpha})$ as a function of $\mathbf{w}$,w_0.
  - Find *functions* $\mathbf{w}(\boldsymbol{\alpha}), w_0(\boldsymbol{\alpha})$ that attain the minimum $\forall\,\boldsymbol{\alpha}$.
- Next, maximize $J(\mathbf{w}(\boldsymbol{\alpha}), w_0(\boldsymbol{\alpha}); \boldsymbol{\alpha})$ as a function of $\boldsymbol{\alpha}$.
- In the end, the solution is given by $\boldsymbol{\alpha}^*$;
  find $\mathbf{w}(\boldsymbol{\alpha}^*)$ and $w_0(\boldsymbol{\alpha}^*)$ by substitution.

Minimizing J with respect to $\mathbf{w}, w_0$

- For fixed α we can minimize

$$J(\mathbf{w}, w_0; \boldsymbol{\alpha}) \;=\; \frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{N}\alpha_i\left[1 - y_i(w_0 + \mathbf{w}\cdot\mathbf{x}_i)\right]$$

  by setting derivatives w.r.t. $w_0, \mathbf{w}$ to zero:

$$\frac{\partial}{\partial \mathbf{w}}J(\mathbf{w}, w_0; \boldsymbol{\alpha}) \;=\; \mathbf{w} - \sum_{i=1}^{N}\alpha_i y_i \mathbf{x}_i \qquad = 0,$$

$$\frac{\partial}{\partial w_0}J(\mathbf{w}, w_0; \boldsymbol{\alpha}) \;=\; -\sum_{i=1}^{N}\alpha_i y_i \qquad = 0.$$

- Note that the bias term w_0 dropped out but has produced a “global” constraint on $\boldsymbol{\alpha}$.

Solving for α

$$\underbrace{\mathbf{w}(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i}_{\text{later: Representer theorem!}}, \qquad \sum_{i=1}^{N} \alpha_i y_i = 0.$$

- Now can (with a bit of algebra) substitute this solution into

$$\max_{\boldsymbol{\alpha} \geq 0, \sum_i \alpha_i y_i = 0} \left\{ \frac{1}{2} \|\mathbf{w}(\boldsymbol{\alpha})\|^2 + \sum_{i=1}^{N} \alpha_i \left[1 - y_i(w_0(\boldsymbol{\alpha}) + \mathbf{w}(\boldsymbol{\alpha}) \cdot \mathbf{x}_i)\right] \right\}$$
$$= \max_{\boldsymbol{\alpha} \geq 0, \sum_i \alpha_i y_i = 0} \left\{ \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \right\}.$$

Max-margin and quadratic programming

- We started by writing down the max-margin problem and arrived at the *dual problem* in α:

$$\max \left\{ \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \right\}$$

subject to $\sum_{i=1}^{N} \alpha_i y_i = 0,\ \alpha_i \geq 0$ for all $i = 1, \ldots, N$.

- Solving this *quadratic program* with linear constraints yields $\boldsymbol{\alpha}^*$.
- We substitute $\boldsymbol{\alpha}^*$ back to get $\mathbf{w}$:

$$\hat{\mathbf{w}} = \mathbf{w}(\boldsymbol{\alpha}^*) = \sum_{i=1}^{N} \alpha_i^* y_i \mathbf{x}_i$$

Maximum margin decision boundary

$$\hat{\mathbf{w}} \;=\; \mathbf{w}(\alpha^*) \;=\; \sum_{i=1}^{N} \alpha_i^* y_i \mathbf{x}_i$$

- Suppose that, under the optimal solution, the margin (distance to the boundary) of a particular $\mathbf{x}_i$ is

$$y_i \left(w_0 + \hat{\mathbf{w}} \cdot \mathbf{x}_i\right) > 1.$$

- Then, necessarily, $\alpha_i^* = 0 \Rightarrow$ not a support vector.
- The direction of the max-margin decision boundary is

$$\hat{\mathbf{w}} \;=\; \sum_{\alpha_i^* > 0} \alpha_i^* y_i \mathbf{x}_i.$$

- w_0 is set by making the margin equidistant to two classes.

Support vectors

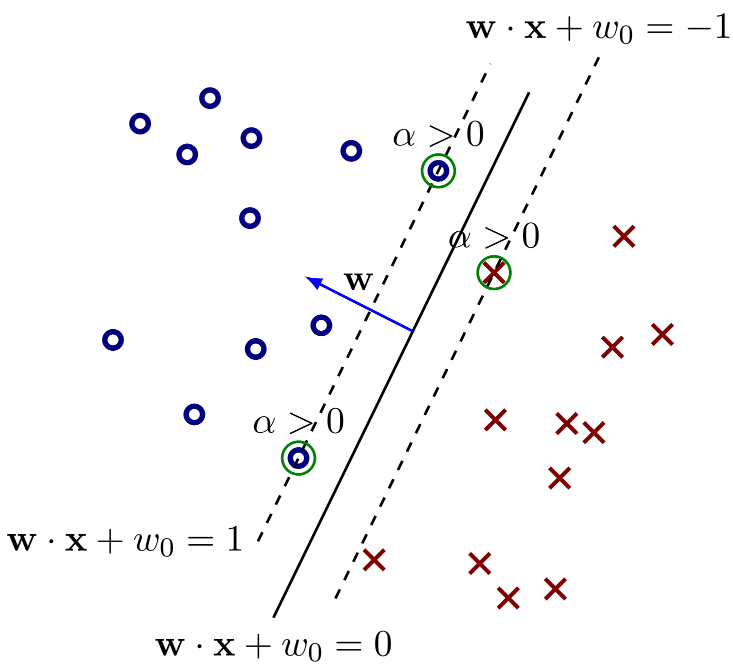
$$\hat{\mathbf{w}} \;=\; \sum_{\alpha_i > 0} \alpha_i y_i \mathbf{x}_i.$$

- Given a test example $\mathbf{x}$, it is classified by

$$\begin{aligned}
\hat{y} \;&=\; \mathrm{sign}\left(\hat{w_0} + \hat{\mathbf{w}} \cdot \mathbf{x}\right) \\
&=\; \mathrm{sign}\left(\hat{w_0} + (\sum_{\alpha_i > 0} \alpha_i y_i \mathbf{x}_i) \cdot \mathbf{x}\right) \\
&=\; \mathrm{sign}\left(\hat{w_0} + \sum_{\alpha_i > 0} \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x}\right)
\end{aligned}$$

- The classifier is based on the expansion in terms of dot products of $\mathbf{x}$ with support vectors.

SVM geometry



- Support vectors:

$$\alpha_i > 0$$

$$y_i(\mathbf{w}\cdot\mathbf{x}_i + w_0) \;=\; 1$$

- Other examples:

$$\alpha_i = 0$$

$$y_i(\mathbf{w}\cdot\mathbf{x}_i + w_0) \;>\; 1$$

Non-separable case

- Not linearly separable data: we can no longer satisfy $y_i\left(\mathbf{w}\cdot\mathbf{x}_i + w_0\right) \geq 1$ for all i.
- Recall the constraint-based terms in separable case:

$$\max_{\boldsymbol{\alpha}\geq 0} \sum_i \alpha_i \left[1 - y_i(w_0 + \mathbf{w}\cdot\mathbf{x}_i)\right]$$

- We can no longer have $\boldsymbol{\alpha} \geq 0$ if constraint violation is unavoidable; would yield $J = \infty$
- We will set maximum penalty on constraint violation:

$$\max_{\color{red}{0\leq\boldsymbol{\alpha}\leq C}} \sum_i \alpha_i \left[1 - y_i(w_0 + \mathbf{w}\cdot\mathbf{x}_i)\right]$$

Slack variables

- We introduce *slack variables* to satisfy margin constraints

$$y_i\left(w_0 + \mathbf{w}\cdot\mathbf{x}_i\right) - 1 + \xi_i \;\geq\; 0, \qquad \xi_i \;\geq\; 0.$$

- We want ξ_i to capture the *minimum* amount we need to fix:

$$\xi_i \;=\; \max\left\{0,\, 1 - y_i\left(w_0 + \mathbf{w}\cdot\mathbf{x}_i\right)\right\}$$

  note: ξ_i is really a function of $\mathbf{w}$
- Our objective now:

$$\min_{\mathbf{w}}\left\{\frac{1}{2}\|\mathbf{w}\|^2 \;+\; C\sum_{i=1}^{N}\xi_i\right\}.$$

Non-separable case: solution

$$\min_{\mathbf{w}}\left\{\frac{1}{2}\|\mathbf{w}\|^2 \;+\; C\sum_{i=1}^{N}\xi_i\right\}.$$

- We can solve this using Lagrange multipliers
  - Introduce additional multipliers for the $\xi \;\geq\; 0$.
- The resulting dual problem:

$$\max\left\{\sum_{i=1}^{N}\alpha_i \;-\frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j \mathbf{x}_i\cdot\mathbf{x}_j\right\}$$

subject to $\sum_{i=1}^{N}\alpha_i y_i = 0,\; 0 \leq \boldsymbol{\alpha} \leq C.$

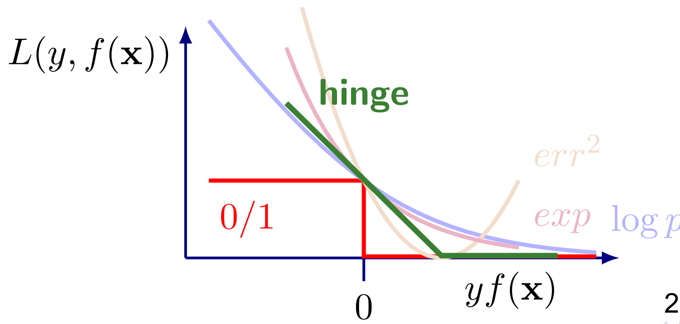
Loss in SVM

$$\min_{\mathbf{w}} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 \;+\; C\sum_{i=1}^{N}\xi_i \right\}$$

- L_2-regularized loss, measured as

$$\sum_{i=1}^{N}\xi_i \;=\sum_{i=1}^{N}\max\left\{0, 1 - y_i(w_0 + \mathbf{w}\cdot\mathbf{x}_i)\right\}$$

- This surrogate loss is known as *hinge loss*



Solving SVM in the primal

- Setting $\lambda = 2/C$ we get

primal: $$\min_{\mathbf{w}} \frac{\lambda}{2}\|\mathbf{w}\|^2 \;+\; \sum_{i=1}^{N}\max\left\{0, 1 - y_i\mathbf{w}\cdot\mathbf{x}_i\right\}$$

- Traditional tactic: write the dual, solve using QP
- Alternative: optimize the primal directly using gradient descent
- Problem: hinge loss is not differentiable at $y\mathbf{w}\cdot\mathbf{x} = 1$
- Solution: *sub*gradient descent

Review: subgradient

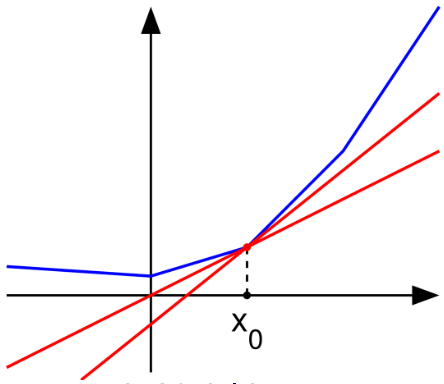


Figure: A. Vedaldi

- Subgradient of L at $\mathbf{w}$ is any $\mathbf{g}$ s.t.

  $$\forall \mathbf{w}' : \quad L(\mathbf{w}') \geq L(\mathbf{w}) + \mathbf{g} \cdot (\mathbf{w}' - \mathbf{w})$$

  i.e., $\mathbf{g}$ defines a tight linear lower bound on L at $\mathbf{w}$

- Subdifferential of L at $\mathbf{w}$:
  $\partial L(\mathbf{w}) = \{\mathbf{g} : \mathbf{g} \text{ is a subgradient of } L \text{ at } \mathbf{w}\}$
- If L is differentiable at $\mathbf{w}$ then $\partial L(\mathbf{w}) = \{\nabla L(\mathbf{w})\}$

SVM via subgradient descent

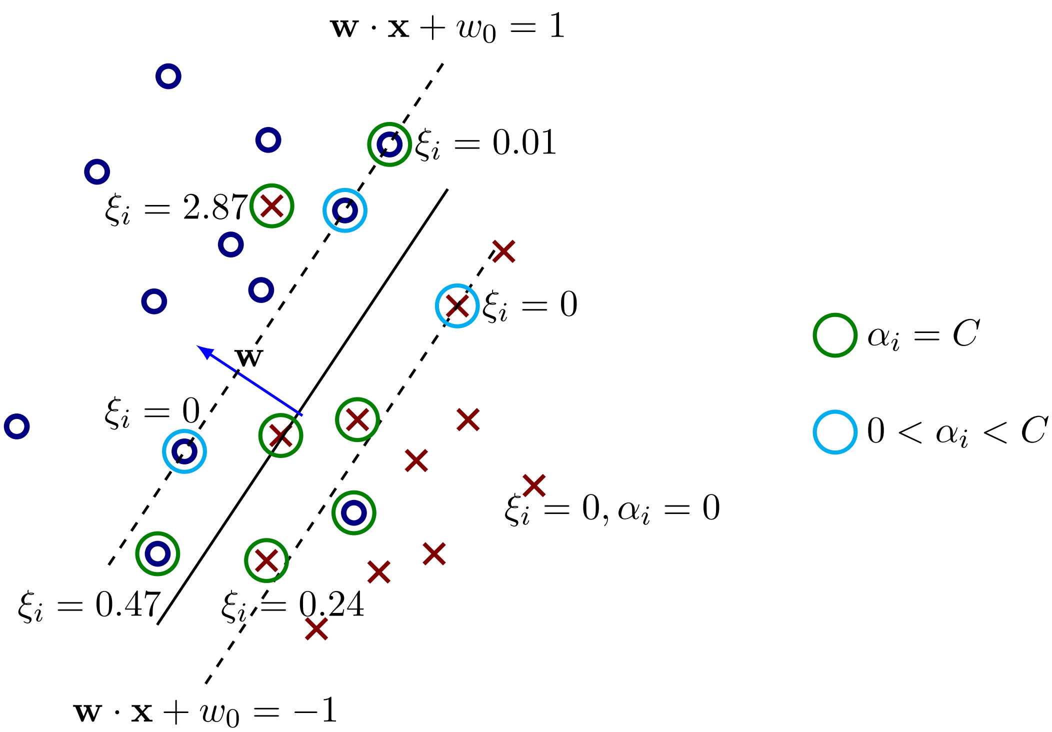
primal:
$$\min_{\mathbf{w}} \frac{\lambda}{2} \|\mathbf{w}\|^2 + \sum_{i=1}^{N} \underbrace{\max\{0, 1 - y_i \mathbf{w} \cdot \mathbf{x}_i\}}_{L_i(\mathbf{w}, w_0)}$$

- Subgradient of the hinge loss on $(\mathbf{x}_i, y_i)$:

$$\nabla_{\mathbf{w}} L_i(\mathbf{w}, w_0) = \begin{cases} \text{if } y_i(\mathbf{w} \cdot \mathbf{x}_i + w_0) < 1 : & -y_i \mathbf{x}_i \\ \text{if } y_i(\mathbf{w} \cdot \mathbf{x}_i + w_0) \geq 1 : & 0 \end{cases}$$

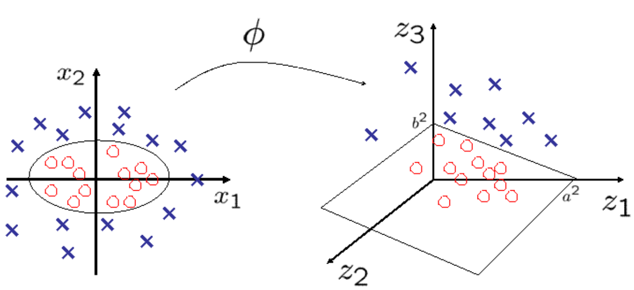
- Similarly compute for $\partial L_i / \partial w_0$
- Remember to add gradient of the regularizer!
- An interesting interpretation: if current $\mathbf{w}, w_0$ classify $(\mathbf{x}_i, y_i)$ correctly with large enough margin, that example contributes nothing to update (not a support vector)

SVM geometry (general case)



Nonlinear features

- As with logistic regression, we can move to nonlinear classifiers by mapping data into nonlinear *feature space*. Ecample:



$$\phi : [x_1, x_2] \rightarrow [x_1^2, \sqrt{2}x_1x_2, x_2^2]$$

- Elliptical decision boundary in the input space becomes linear in the feature space $\mathbf{z} = \phi(\mathbf{x})$:

$$\frac{x_1^2}{a^2} + \frac{x_2^2}{b^2} = c \Rightarrow \frac{z_1}{a^2} + \frac{z_3}{b^2} = c.$$

Example of nonlinear mapping

- Consider the mapping:
  $\phi : [x_1, x_2] \rightarrow [1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1 x_2]$.
- The (linear) SVM classifier in the feature space:

$$\hat{y} = \mathrm{sign}\left(\hat{w_0} + \sum_{\alpha_i > 0} \alpha_i y_i \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x})\right)$$

- The dot product in the feature space:

$$\begin{aligned}\phi(\mathbf{x}) \cdot \phi(\mathbf{z}) &= 1 + 2x_1 z_1 + 2x_2 z_2 + x_1^2 z_1^2 + x_2^2 z_2^2 + 2x_1 x_2 z_1 z_2 \\ &= (1 + \mathbf{x} \cdot \mathbf{z})^2 .\end{aligned}$$

Dot products and feature space

- We defined a non-linear mapping into feature space

$$\phi : [x_1, x_2] \rightarrow [1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1 x_2]$$

  and saw that $\phi(\mathbf{x}) \cdot \phi(\mathbf{z}) = K(\mathbf{x}, \mathbf{z})$ using the *kernel*

$$K(\mathbf{x}, \mathbf{z}) = (1 + \mathbf{x} \cdot \mathbf{z})^2 .$$

- I.e., we can calculate dot products in the feature space implicitly, without ever writing the feature expansion!

The kernel trick

- Replace dot products in the SVM formulation with kernel values.
- The optimization problem:

$$\max \left\{ \sum_{i=1}^{N} \alpha_i \; - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \right\}$$

  - Need to compute the *kernel matrix* for the training data
- The classifier:

$$\hat{y} = \operatorname{sign}\left(\hat{w}_0 + \sum_{\alpha_i > 0} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) \right)$$

  - Need to compute $K(\mathbf{x}_i, \mathbf{x})$ for all SVs $\mathbf{x}_i$.

Representer theorem

- Consider the optimization problem

$$\mathbf{w}^* \;=\; \operatorname*{argmin}_{\mathbf{w}} \|\mathbf{w}\|^2 \quad \text{s.t. } y_i(\mathbf{w} \cdot \mathbf{x}_i + w_0) \geq 1 \; \forall i$$

- Theorem: the solution can be represented as

$$\mathbf{w}^* \;=\; \sum_{i=1}^{N} \beta_i \mathbf{x}_i$$

- This is the “magic” behind Support Vector Machines!

Representer theorem - proof I

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \|\mathbf{w}\|^2 \quad \text{s.t. } y_i(\mathbf{w} \cdot \mathbf{x}_i + w_0) \geq 1 \; \forall i \quad \Rightarrow \quad \mathbf{w}^* = \sum_{i=1}^{N} \beta_i \mathbf{x}_i$$

- Let $\mathbf{w}^* = \mathbf{w}_X + \mathbf{w}_\perp$, where
  $\mathbf{w}_X = \sum_{i=1}^{N} \beta_i \mathbf{x}_i \in Span(\mathbf{x}_1, \ldots, \mathbf{x}_N)$,
  $\mathbf{w}_\perp \notin Span(\mathbf{x}_1, \ldots, \mathbf{x}_N)$, i.e., $\mathbf{w}_\perp \cdot \mathbf{x}_i = 0$ for all $i = 1, \ldots, N$
- For all $\mathbf{x}_i$ we have

$$\mathbf{w}^* \cdot \mathbf{x}_i = \mathbf{w}_X \cdot \mathbf{x}_i + \mathbf{w}_\perp \cdot \mathbf{x}_i = \mathbf{w}_X \cdot \mathbf{x}_i$$

therefore,

$$y_i(\mathbf{w}^* \cdot \mathbf{x}_i + w_0) \geq 1 \quad \Rightarrow \quad y_i(\mathbf{w}_X \cdot \mathbf{x}_i + w_0) \geq 1$$

Representer theorem - proof II

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \|\mathbf{w}\|^2 \quad \text{s.t. } y_i(\mathbf{w} \cdot \mathbf{x}_i + w_0) \geq 1 \; \forall i \quad \Rightarrow \quad \mathbf{w}^* = \sum_{i=1}^{N} \beta_i \mathbf{x}_i$$

- Now, we have

$$\|\mathbf{w}^*\|^2 = \mathbf{w}^* \cdot \mathbf{w}^* = (\mathbf{w}_X + \mathbf{w}_\perp) \cdot (\mathbf{w}_X + \mathbf{w}_\perp) = \underbrace{\mathbf{w}_X \cdot \mathbf{w}_X}_{\|\mathbf{w}_X\|^2} + \underbrace{\mathbf{w}_\perp \cdot \mathbf{w}_\perp}_{\|\mathbf{w}_\perp\|^2},$$

since $\mathbf{w}_X \cdot \mathbf{w}_\perp = 0$.

- Suppose $\mathbf{w}_\perp \neq \mathbf{0}$. Then, we have a solution $\mathbf{w}_X$ that satisfies all the constraints, and for which
  $\|\mathbf{w}_X\|^2 < \|\mathbf{w}_X\|^2 + \|\mathbf{w}_\perp\|^2 = \|\mathbf{w}^*\|^2$.
- This contradicts optimality of $\mathbf{w}^*$, hence $\mathbf{w}^* = \mathbf{w}_X$. QED

Kernel SVM in the primal

- Recall: $\hat{y} = \text{sign}\left(\hat{w_0} + \sum_{\alpha_i>0} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x})\right)$
- Can not write $\mathbf{w}$ explicitly; instead, optimize $\boldsymbol{\alpha}$
- How can we write the regularizer?

$$\|\mathbf{w}\|^2 = \mathbf{w} \cdot \mathbf{w} = \left[\sum_i \alpha_i y_i \boldsymbol{\phi}(\mathbf{x}_i)\right] \cdot \left[\sum_j \alpha_j y_j \boldsymbol{\phi}(\mathbf{x}_j)\right]$$
$$= \sum_{i=1,j=1}^{N} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

- The objective for learning is

$$\min_{\boldsymbol{\alpha}} \left\{ \frac{\lambda}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) + \sum_i \left[1 - y_i \sum_j \alpha_j y_j K(\mathbf{x}_i, \mathbf{x}_j)\right]_+ \right\}$$

Mercer's kernels

- What kind of function K is a valid kernel, i.e. such that there exists a feature space $\Phi(\mathbf{x})$ in which $K(\mathbf{x}, \mathbf{z}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{z})$?
- Theorem due to Mercer (1909): K must be
  - Continuous;
  - symmetric: $K(\mathbf{x}, \mathbf{z}) = K(\mathbf{z}, \mathbf{x})$;
  - positive definite: for any $\mathbf{x}_1, \ldots, \mathbf{x}_N$, the *kernel matrix*

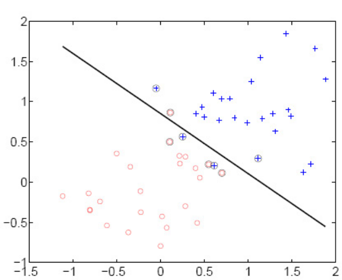
$$K = \begin{bmatrix} K(\mathbf{x}_1, \mathbf{x}_1) & K(\mathbf{x}_1, \mathbf{x}_2) & K(\mathbf{x}_1, \mathbf{x}_N) \\ \ldots & \ldots & \ldots \\ K(\mathbf{x}_N, \mathbf{x}_1) & K(\mathbf{x}_N, \mathbf{x}_2) & K(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix}$$

  must be positive definite.

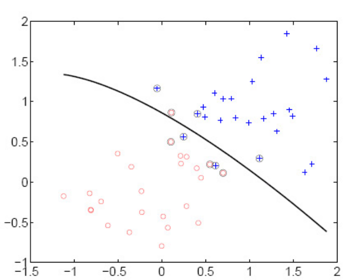
Some popular kernels

- The linear kernel:
$$K(\mathbf{x}, \mathbf{z}) \;=\; \mathbf{x} \cdot \mathbf{z}.$$
This leads to the original, linear SVM.
- The polynomial kernel:
$$K(\mathbf{x}, \mathbf{z};\, b, p) \;=\; (b + \mathbf{x} \cdot \mathbf{z})^p.$$
We can write the expansion explicitly, by concatenating powers up to d and multiplying by appropriate weights.
- How many dimensions are in $\phi(\mathbf{x})$? If $\mathbf{x} \in \mathbb{R}^d$, and $d \gg p$, number of terms grows as d^p.

Example: SVM with polynomial kernel

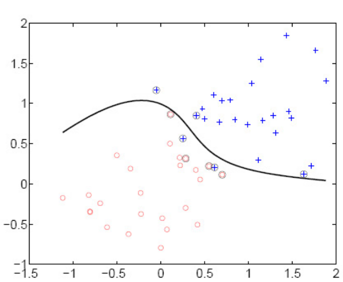


linear

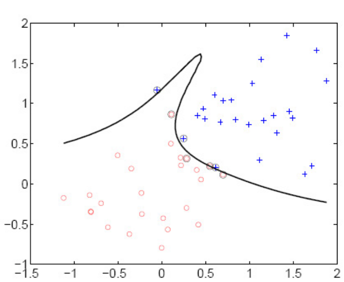


2^{nd} order polynomial

(using $C < \infty$)



4^{th} order polynomial



8^{th} order polynomial

Compare to the effect of model order in regression or logistic regression.

Radial basis function kernel

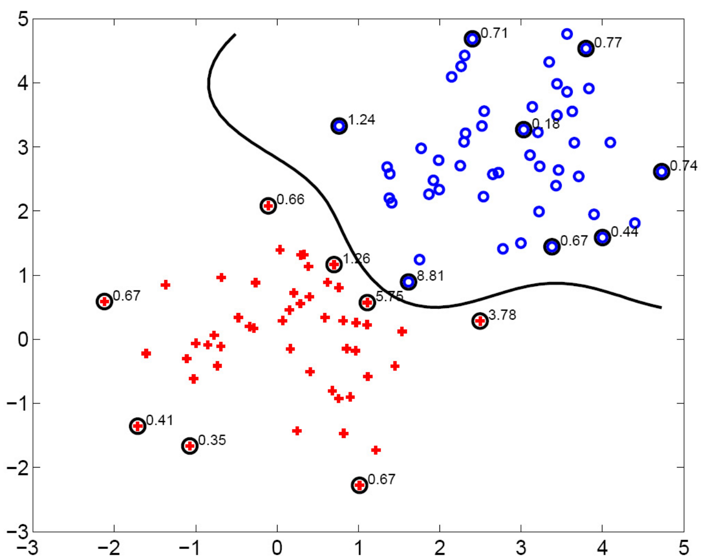
$$K(\mathbf{x}, \mathbf{z}; \sigma) = \exp\left(-\frac{1}{\sigma^2}\|\mathbf{x} - \mathbf{z}\|^2\right).$$

- The RBF kernel is a measure of similarity between two examples.
  - The feature space is infinite-dimensional!
- What is the role of parameter σ? Consider $\sigma \to 0$.

$$K(\mathbf{x}_i, \mathbf{x}; \sigma) \to \begin{cases} 1 & \text{if } \mathbf{x} = \mathbf{x}_i, \\ 0 & \text{if } \mathbf{x} \neq \mathbf{x}_i. \end{cases}$$

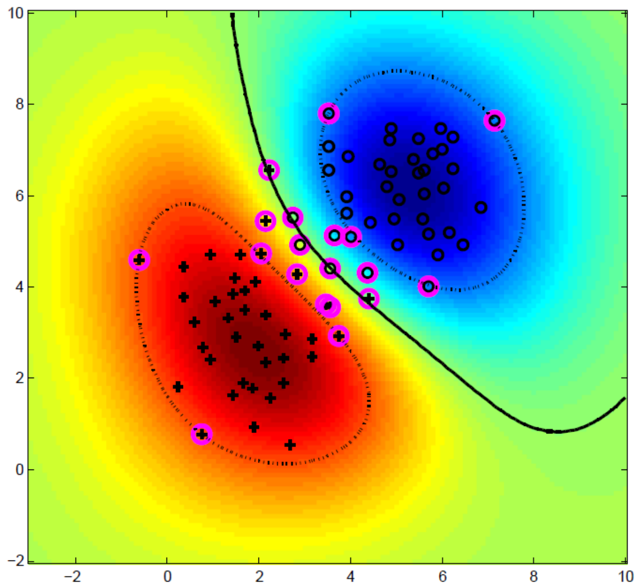
- All examples become SVs $\Rightarrow$ likely overfitting.

SVM with RBF (Gaussian) kernels



- Data are linearly separable in the (infinite-dimensional) feature space
- We don't need to explicitly compute dot products in that feature space – instead we simply evaluate the RBF kernel.

SVM with RBF kernels: geometry



- positive margin: level set

$$\{\mathbf{x} : \sum \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) = 1\}$$

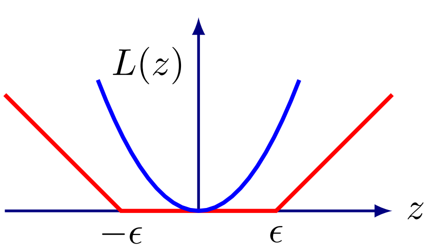
- negative margin: level set

$$\{\mathbf{x} : \sum \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) = -1\}$$

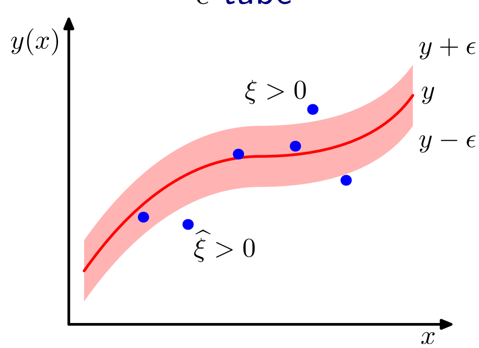
SVM regression

- The key ideas:

ϵ-insensitive loss



ϵ-tube



- Two sets of slack variables:

$$y_i \le f(\mathbf{x}_i) + \epsilon + \xi_i,$$
$$y_i \ge f(\mathbf{x}_i) - \epsilon - \tilde{\xi}_i, \qquad \xi_i \ge 0,\ \tilde{\xi}_i \ge 0.$$

- Optimization: $\min C \sum_i \left(\xi_i + \tilde{\xi}_i\right) + \frac{1}{2}\|\mathbf{w}\|^2$