Lecture 13: Support Vector Machines

CS 475: Machine Learning

Raman Arora

March 13, 2017

Review: SVM (dual)

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

s.t. $\quad y_i(w_0 + \mathbf{w}^T\mathbf{x}_i) \;\geq\; 1 - \xi_i \quad \forall\, i = 1, \ldots, N$

- We can solve this using Lagrange multipliers for all constraints
- The resulting dual problem:

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

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

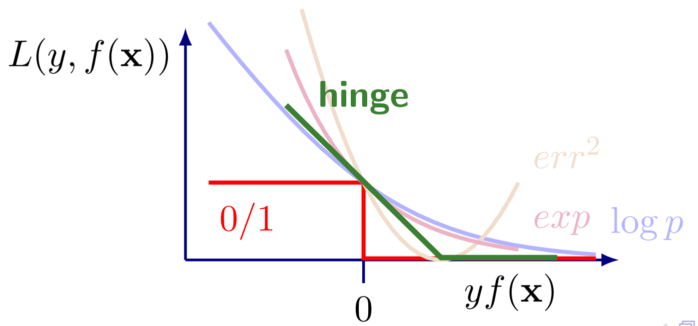
Loss in SVM

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

- 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

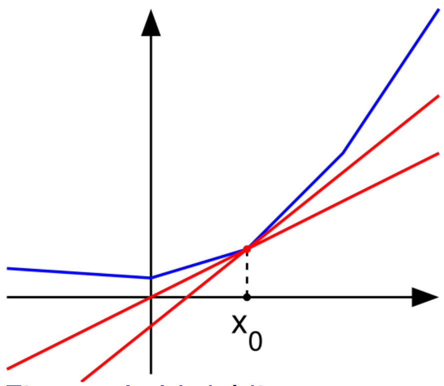


Figure: A. Vedaldi

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

SVM via subgradient descent

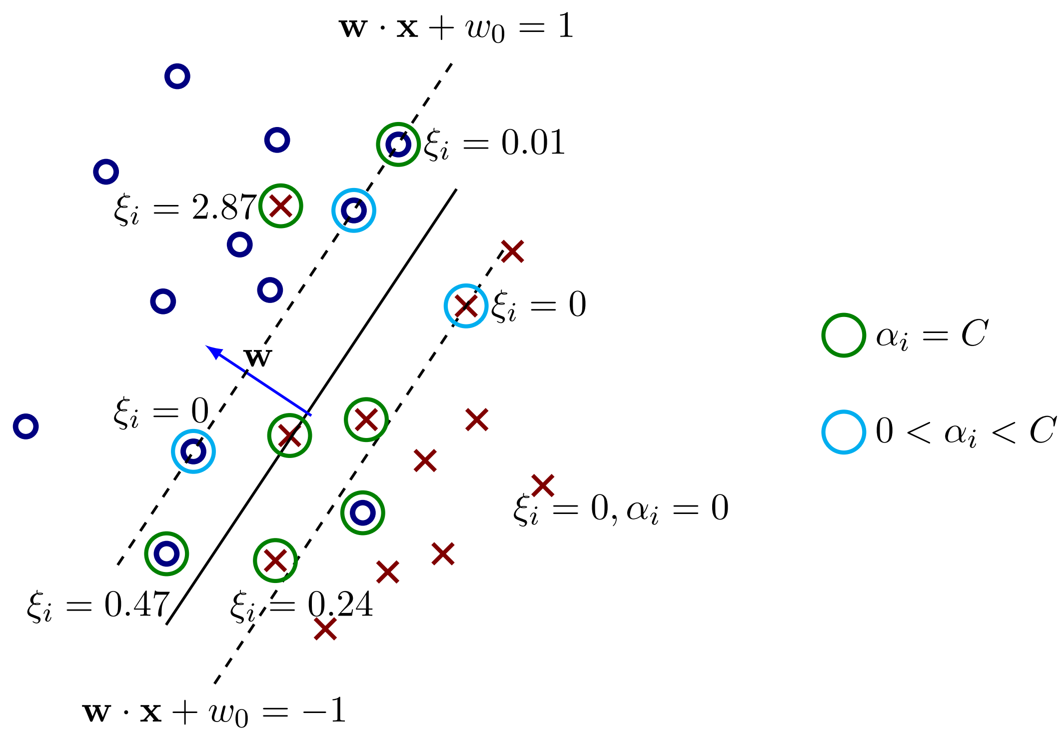
primal: $$\min_{\mathbf{w}} \frac{\lambda}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{N} \underbrace{\max\left\{0, 1 - y_i \mathbf{w} \cdot \mathbf{x}_i\right\}}_{L_i(\mathbf{w}, w_0)}$$

- 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)



Review: linear SVM classifier

- The form of the trained linear SVM classifier:

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

- The support vectors' contributions are summarized in

$$\widehat{\mathbf{w}} = \sum_i \alpha_i y_i \mathbf{x}_i$$

Dot product similarity

- First, consider two unit vectors $\mathbf{u}$ and $\mathbf{v}$, $\|\mathbf{u}\| = \|\mathbf{v}\| = 1$.
- Dot product measures angle between them

$$\mathbf{u} \cdot \mathbf{v} \; = \; \cos(\measuredangle \mathbf{u}, \mathbf{v})$$

  If we consider $\mathbf{u}, \mathbf{v}$ to represent directions in feature space, this is a measure of similarity
- $\mathbf{u} \cdot \mathbf{v}$ ranges from -1 when $\mathbf{u} = -\mathbf{v}$ to 1 when $\mathbf{u} = \mathbf{v}$
- When the vectors are not unit length:

$$\mathbf{u} \cdot \mathbf{v} \; = \; \sqrt{\|\mathbf{u}\|\|\mathbf{v}\| \cos(\measuredangle \mathbf{u}, \mathbf{v})}$$

Dot products and SVM

$$\hat{y} \; = \; \text{sign}\left(\hat{w}_0 + \sum_{\alpha_i > 0} \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x}\right)$$

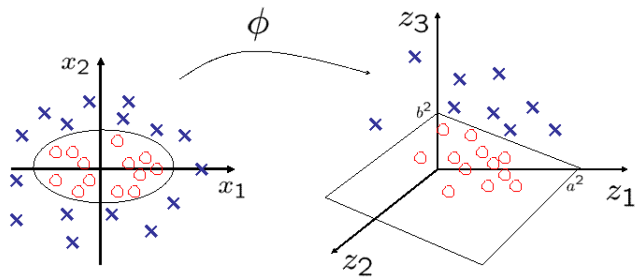
- Interpretation: each SV $\mathbf{x}_i$ "votes" for $\mathbf{x}$ to be assigned to class y_i
  The "trust" we place in its vote is determined by α_i
  It is modulated by similarity to $\mathbf{x}$, measured by $\mathbf{x}_i \cdot \mathbf{x}$
- If $\mathbf{x}_i \cdot \mathbf{x} = 0$ (orthogonal) no $\mathbf{x}_i$ has no opinion on $\mathbf{x}$;
  if $\mathbf{x}_i \cdot \mathbf{x} = 1$ it wants $\hat{y} = y_i$ (but may be overridden by other SVs);
  if $\mathbf{x}_i \cdot \mathbf{x} = -1$ (opposite) it wants $\hat{y} = -y_i$
- Often done in practice: normalize every example to unit length before training SVM

$$\mathbf{x}' \; = \; \frac{\mathbf{x}}{\|\mathbf{x}\|}$$

  especially for sparse, high-dim data

Nonlinear features

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



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

- 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:
  $\phi : \ [x_1, x_2] \ \to \ [1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2]$.
- 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:

$$\begin{aligned} \phi(\mathbf{x}) \cdot \phi(\mathbf{z}) &= 1 + 2x_1z_1 + 2x_2z_2 + x_1^2z_1^2 + x_2^2z_2^2 + 2x_1x_2z_1z_2 \\ &= (1 + \mathbf{x} \cdot \mathbf{z})^2 . \end{aligned}$$

Dot products and feature space

- We defined a non-linear mapping into feature space

$$\phi: \; [x_1, x_2] \; \rightarrow \; [1, \, \sqrt{2}x_1, \, \sqrt{2}x_2, \, x_1^2, \, x_2^2, \, \sqrt{2}x_1x_2]$$

  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

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

- 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

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

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

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

    must be positive definite.

Some popular kernels

- The linear kernel:

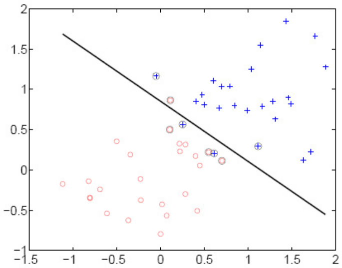
$$K(\mathbf{x}, \mathbf{z}) = \mathbf{x} \cdot \mathbf{z}.$$

  This leads to the original, linear SVM.
- The polynomial kernel:

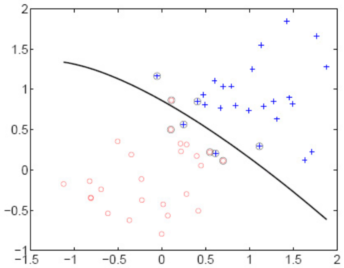
$$K(\mathbf{x}, \mathbf{z};\, b, p) = (b + \mathbf{x} \cdot \mathbf{z})^p.$$

  We can write the expansion explicitly, by concatenating powers up to d and multiplying by appropriate weights.
- How many dimensions are in $\boldsymbol{\phi}(\mathbf{x})$? If $\mathbf{x} \in \mathbb{R}^d$, and $d \gg p$, number of terms grows as d^p.

Example: SVM with polynomial kernel

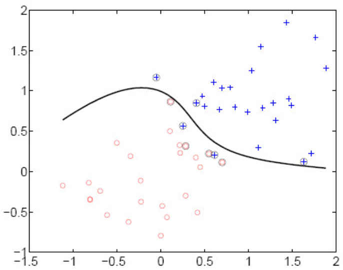


linear

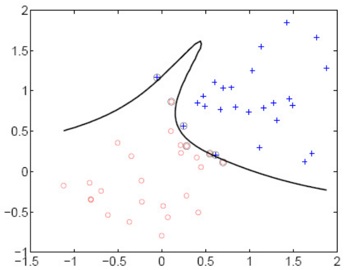


2^{nd} order polynomial

(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

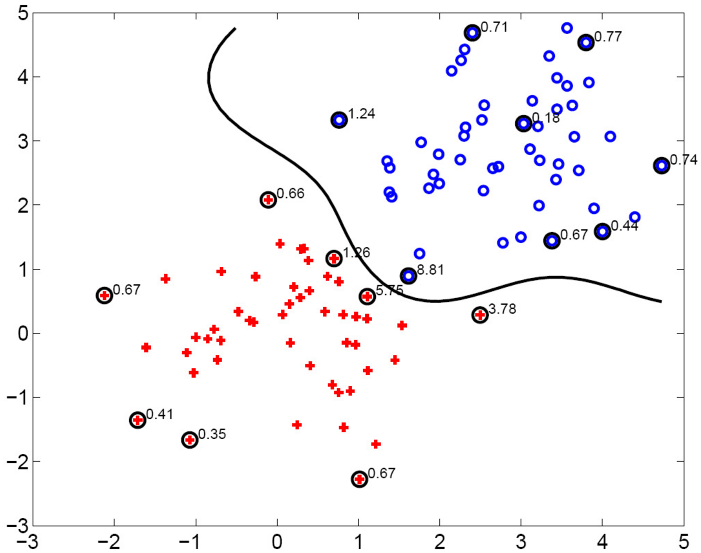
$$K(\mathbf{x}, \mathbf{z}; \sigma) \;=\; \exp\left(-\frac{1}{\sigma^2}\|\mathbf{x}-\mathbf{z}\|^2\right).$$

- 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 $\sigma \to 0$.

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

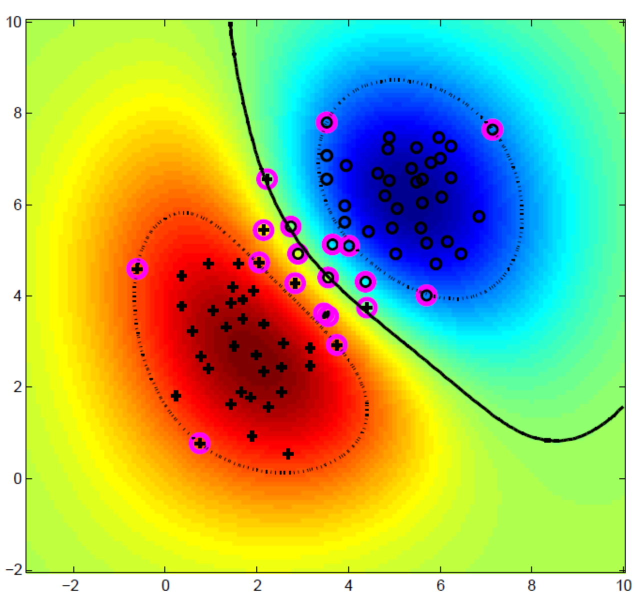
- All examples become SVs $\Rightarrow$ likely overfitting.

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

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

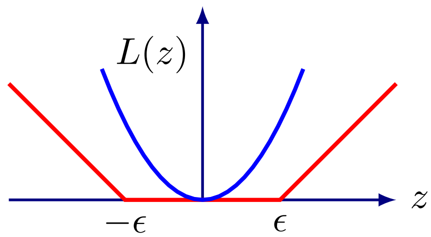
- negative margin: level set

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

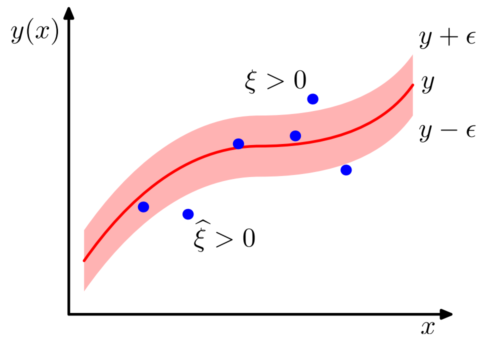
SVM regression

- The key ideas:

  ϵ-insensitive loss



  ϵ-tube



- Two sets of slack variables:

$$
\begin{aligned}
y_i &\leq f(\mathbf{x}_i) + \epsilon + \xi_i, \\
y_i &\geq f(\mathbf{x}_i) - \epsilon - \tilde{\xi}_i, \qquad\qquad \xi_i \geq 0,\ \tilde{\xi}_i \geq 0.
\end{aligned}
$$

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

SVM with more than two classes

- Some classifiers are "natively multiclass"
  e.g., decision trees
- With any natively binary classifier (AdaBoost; logistic regression; SVM), our options for $C > 2$ classes include:
- One-vs-all: build C classifiers
  need to reconcile; easy if have calibrated $p\,(y \,|\, \mathbf{x})$
- One-vs-one: build $\binom{C}{2}$ classifiers
  need to reconcile; more problematic since can have inconsistencies
- Build some sort of "tournament", or a class tree
  often the most efficient; how to build the tree?
- Extend to multi-class by modifying the machinery
  softmax is an extension of logistic regression;
  multi-class SVM extension is next

Multiclass SVM: setup

- Many attempts to generalize SVM to multi-class; we will follow the one due to Crammer and Singer (2000).
- Basic idea: for C classes, learn $\mathbf{w}_c$ for $c = 1, \ldots, C$,

$$\hat{y}(\mathbf{x}; \underbrace{\mathbf{w}_1, \ldots, \mathbf{w}_C}_{\mathbf{W}}) = \operatorname*{argmax}_c \mathbf{w}_c \cdot \mathbf{x}.$$

- Can stack $\mathbf{w}_c$s into rows of $\mathbf{W}$
- Empirical 0/1 loss on (x, y): $[\![\hat{y}(\mathbf{x}; \mathbf{W}) \neq y]\!]$
- Surrogate loss on $(\mathbf{x}, y)$:

$$\max_r \left\{ \mathbf{w}_r{}^T \mathbf{x} + 1 - \delta_{r,y_i} \right\} - \mathbf{w}_y{}^T \mathbf{x}$$

$\delta_{a,b} = 1$ iff $a = b$, otherwise 0.

Optimization

- Surrogate loss is a bound on 0/1 loss:

$$\frac{1}{N} \sum_i [\![\hat{y}(\mathbf{x}_i; \mathbf{W}) \neq y_i]\!] \leq \frac{1}{N} \sum_i \left[\max_r \left\{ \mathbf{w}_r^T \mathbf{x} + 1 - \delta_{y_i,r} \right\} - \mathbf{w}_y^T \mathbf{x}_i \right]$$

- Proceed as in (separable) SVM: want to find the lowest norm solution that achieves 1-margin

$$\min_{\mathbf{W}} \frac{\lambda}{2} \|\mathbf{W}\|_2^2$$
$$\text{s.t. } \forall\, i,\ c \neq y_i \quad \mathbf{w}_{y_i}^T \mathbf{x}_i - \mathbf{w}_c{}^T \mathbf{x}_i \geq 1.$$

where $\|\mathbf{W}\|_2^2$ is the Frobenius norm of $\mathbf{W}$.

Soft constraint version

- General (non-separable) case:

$$\min_{\mathbf{W}} \frac{\lambda}{2}\|\mathbf{W}\|_2^2 + \sum_i \xi_i$$

$$\text{s.t. } \forall\, i,\ c \neq y_i \quad \mathbf{w}_{y_i}^T \mathbf{x}_i - \mathbf{w}_c{}^T \mathbf{x}_i \ \geq\ 1 - \xi_i.$$

- Introducing Lagrange multipliers $\alpha_{i,r}$:

$$\min_{\mathbf{W},\boldsymbol{\xi}} \max_{\boldsymbol{\alpha}} \frac{\lambda}{2}\sum_c \|\mathbf{w}_c\|_2^2 + \sum_i \xi_i$$

$$+ \sum_i \sum_r \alpha_{i,r} \left[(\mathbf{w}_r^T - \mathbf{w}_{y_i}^T)\mathbf{x}_i \ - \delta_{y_i,r} \ + 1 \ - \xi_i \right]$$

$$\text{s.t.} \forall i,\ r, \quad \alpha_{i,r} \ \geq\ 0, \quad \xi_i \ \geq\ 0.$$