Machine learning: lecture 8

Tommi S. Jaakkola
MIT CSAIL
tommi@csail.mit.edu

Topics

- Support vector machine
  - definition, solution, interpretation
  - kernel function, examples, cross-validation
- Kernels and logistic regression

Review: support vector machine

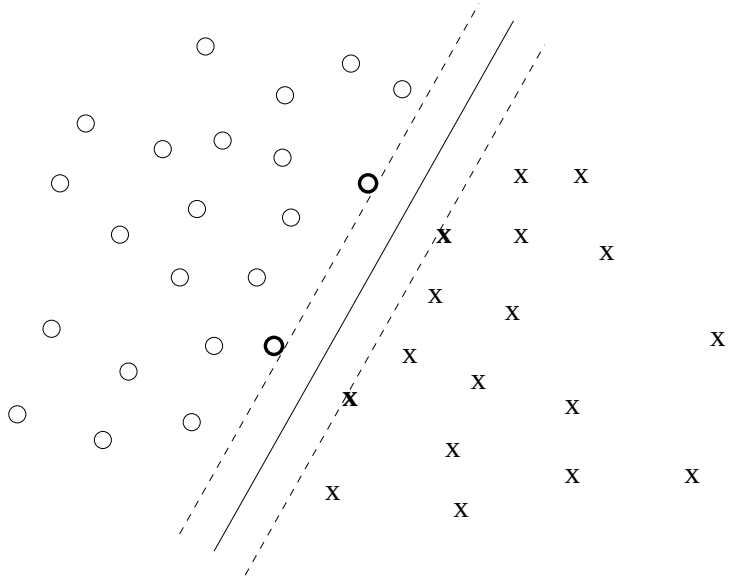
- We minimize the regularization penalty

$$\|\mathbf{w}\|^2/2 = \mathbf{w}^T\mathbf{w}/2 = \sum_{j=1}^{d} w_i^2/2$$

  subject to (hard) classification constraints

$$y_i\left[w_0 + \mathbf{w}^T\mathbf{x}_i\right] - 1 \geq 0, \quad i = 1, \ldots, n$$

- Assuming the training examples are linearly separable, the resulting margin is related to the solution via $\|\hat{\mathbf{w}}\| = 1/\text{margin}$

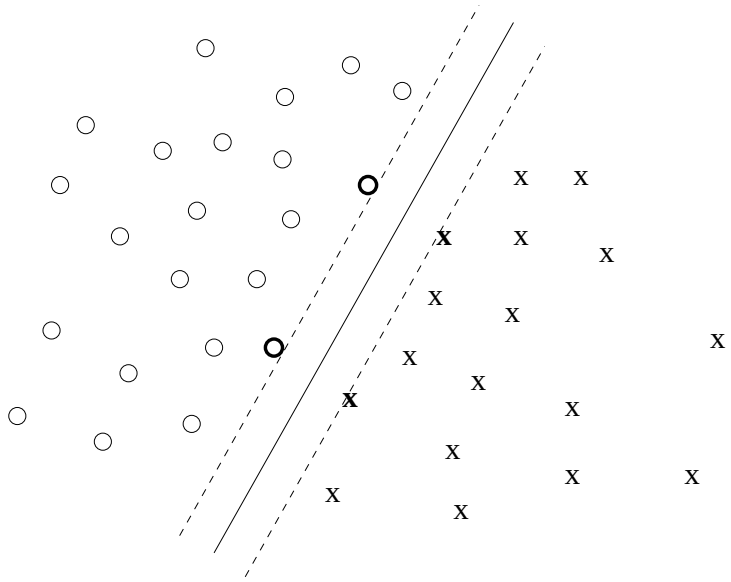


Review: support vector machine

- Only a subset of the classification constraints are relevant

$$y_i \left[w_0 + \mathbf{w}^T \mathbf{x}_i \right] - 1 \geq 0,$$

In other words, we need to enforce only some of the constraints $i \in S$; the others will be satisfied automatically.



$\Rightarrow$ the solution is *sparse* in the sense that it can be defined on the basis of only a subset of the training examples, the *support vectors*

Solution

- We find the optimal setting of $\{w_0, \mathbf{w}\}$ by introducing *Lagrange multipliers* $\alpha_i \geq 0$ for the inequality constraints
- We *minimize*

$$J(\mathbf{w}, w_0, \alpha) = \|\mathbf{w}\|^2/2 - \sum_{i=1}^{n} \alpha_i \left(y_i \left[w_0 + \mathbf{w}^T \mathbf{x}_i \right] - 1 \right)$$

with respect to $\mathbf{w}, w_0$. $\alpha_i \geq 0$ ensure that the classification constraints are indeed satisfied.

For fixed $\{\alpha_i\}$

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

$$\frac{\partial}{\partial w_0} J(\mathbf{w}, w_0, \alpha) = - \sum_{i=1}^{n} \alpha_i y_i = 0$$

Solution cont'd

- Let's use the optimality conditions to write the objective solely in terms of Lagrange multipliers

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i, \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$J(\mathbf{w}, w_0, \alpha) = \|\mathbf{w}\|^2/2 - \sum_{i=1}^{n} \alpha_i \left(y_i \left[w_0 + \mathbf{w}^T \mathbf{x}_i \right] - 1 \right)$$

Solution cont'd

- Let's use the optimality conditions to write the objective solely in terms of Lagrange multipliers

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i, \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$J(\mathbf{w}, w_0, \alpha) = \|\mathbf{w}\|^2/2 - \overbrace{\sum_{i=1}^{n} \alpha_i \left(y_i \left[w_0 + \mathbf{w}^T \mathbf{x}_i \right] - 1 \right)}^{=0}$$

Solution cont'd

- Let's use the optimality conditions to write the objective solely in terms of Lagrange multipliers

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i, \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$J(\mathbf{w}, w_0, \alpha) = \|\mathbf{w}\|^2/2 - \overbrace{\sum_{i=1}^{n} \alpha_i \left(y_i \left[w_0 + \mathbf{w}^T \mathbf{x}_i \right] - 1 \right)}^{=0}$$

$$= \mathbf{w}^T \mathbf{w}/2 - \sum_{i=1}^{n} \alpha_i \left(y_i \left[\mathbf{w}^T \mathbf{x}_i \right] - 1 \right)$$

Solution cont'd

- Let's use the optimality conditions to write the objective solely in terms of Lagrange multipliers

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i, \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\begin{aligned} J(\mathbf{w}, w_0, \alpha) &= \|\mathbf{w}\|^2/2 - \overbrace{\sum_{i=1}^{n} \alpha_i \left(y_i \left[w_0 + \mathbf{w}^T \mathbf{x}_i \right] - 1 \right)}^{=0} \\ &= \mathbf{w}^T \mathbf{w}/2 - \sum_{i=1}^{n} \alpha_i \left(y_i \left[\mathbf{w}^T \mathbf{x}_i \right] - 1 \right) \\ &= \sum_{i=1}^{n} \alpha_i + \mathbf{w}^T \mathbf{w}/2 - \sum_{i=1}^{n} \alpha_i y_i \mathbf{w}^T \mathbf{x}_i \end{aligned}$$

Solution cont'd

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i, \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$J(\mathbf{w}, w_0, \alpha) = \sum_{i=1}^{n} \alpha_i + \mathbf{w}^T \mathbf{w}/2 - \sum_{i=1}^{n} \alpha_i y_i \mathbf{w}^T \mathbf{x}_i$$

Solution cont'd

$$\begin{aligned}
\mathbf{w} &= \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i, \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \\
J(\mathbf{w}, w_0, \alpha) &= \sum_{i=1}^{n} \alpha_i + \mathbf{w}^T \mathbf{w}/2 - \sum_{i=1}^{n} \alpha_i y_i \mathbf{w}^T \mathbf{x}_i \\
&= \sum_{i=1}^{n} \alpha_i + \mathbf{w}^T \mathbf{w}/2 - \mathbf{w}^T \left(\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \right)
\end{aligned}$$

Solution cont'd

$$
\begin{aligned}
\mathbf{w} &= \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i, \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \\
J(\mathbf{w}, w_0, \alpha) &= \sum_{i=1}^{n} \alpha_i + \mathbf{w}^T\mathbf{w}/2 - \sum_{i=1}^{n} \alpha_i y_i \mathbf{w}^T \mathbf{x}_i \\
&= \sum_{i=1}^{n} \alpha_i + \mathbf{w}^T\mathbf{w}/2 - \mathbf{w}^T \left(\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \right) \\
&= \sum_{i=1}^{n} \alpha_i + \mathbf{w}^T\mathbf{w}/2 - \mathbf{w}^T\mathbf{w}
\end{aligned}
$$

Solution cont'd

$$
\begin{aligned}
\mathbf{w} &= \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i, \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \\
J(\mathbf{w}, w_0, \alpha) &= \sum_{i=1}^{n} \alpha_i + \mathbf{w}^T\mathbf{w}/2 - \sum_{i=1}^{n} \alpha_i y_i \mathbf{w}^T \mathbf{x}_i \\
&= \sum_{i=1}^{n} \alpha_i + \mathbf{w}^T\mathbf{w}/2 - \mathbf{w}^T \left(\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \right) \\
&= \sum_{i=1}^{n} \alpha_i + \mathbf{w}^T\mathbf{w}/2 - \mathbf{w}^T\mathbf{w} \\
&= \sum_{i=1}^{n} \alpha_i - \mathbf{w}^T\mathbf{w}/2
\end{aligned}
$$

Solution cont'd

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i, \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$J(\mathbf{w}, w_0, \alpha) = \sum_{i=1}^{n} \alpha_i - \mathbf{w}^T \mathbf{w}/2$$

Solution cont'd

$$\begin{aligned}
\mathbf{w} &= \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i, \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \\
J(\mathbf{w}, w_0, \alpha) &= \sum_{i=1}^{n} \alpha_i - \mathbf{w}^T \mathbf{w}/2 \\
&= \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \left(\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \right)^T \left(\sum_{j=1}^{n} \alpha_j y_j \mathbf{x}_j \right)
\end{aligned}$$

Solution cont'd

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i, \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\begin{aligned} J(\mathbf{w}, w_0, \alpha) &= \sum_{i=1}^{n} \alpha_i - \mathbf{w}^T \mathbf{w}/2 \\ &= \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \left(\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \right)^T \left(\sum_{j=1}^{n} \alpha_j y_j \mathbf{x}_j \right) \\ &= \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i y_i \mathbf{x}_i^T \mathbf{x}_j \alpha_j y_j \end{aligned}$$

Solution cont'd

$$
\begin{aligned}
\mathbf{w} &= \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i, \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \\
J(\mathbf{w}, w_0, \alpha) &= \sum_{i=1}^{n} \alpha_i - \mathbf{w}^T \mathbf{w}/2 \\
&= \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \left(\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \right)^T \left(\sum_{j=1}^{n} \alpha_j y_j \mathbf{x}_j \right) \\
&= \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i y_i \mathbf{x}_i^T \mathbf{x}_j \alpha_j y_j \\
&= \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i^T \mathbf{x}_j)
\end{aligned}
$$

Solution cont'd

- By substituting the solution $\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$ back into the objective we get the following (dual) optimization problem over the Lagrange multipliers:

  We *maximize*

  $$J(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i^T \mathbf{x}_j)$$

  subject to the constraints

  $$\alpha_i \geq 0, \quad i = 1, \ldots, n, \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

  (For non-separable problems we have to limit $\alpha_i \leq C$)

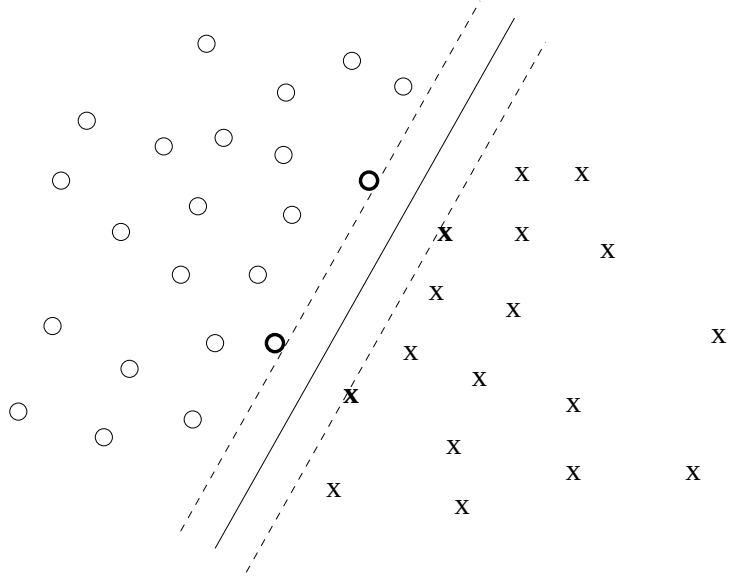
- This is a *quadratic programming problem*

Support vector machines

- Once we have the Lagrange multipliers $\{\hat{\alpha}_i\}$, we can reconstruct the parameter vector $\hat{\mathbf{w}}$ as a weighted combination of the training examples:

$$\hat{\mathbf{w}} = \sum_{i=1}^{n} \hat{\alpha}_i y_i \mathbf{x}_i$$

where the "weight" $\hat{\alpha}_i = 0$ for all but the *support vectors* (SV)



- The discriminant function has an interpretable form

$$\hat{\mathbf{w}}^T \mathbf{x} + \hat{w}_0 = \sum_{i \in SV} \hat{\alpha}_i \, y_i \, (\mathbf{x}_i^T \mathbf{x}) + \hat{w}_0 = f(\mathbf{x}; \hat{\alpha}, \hat{w}_0)$$

Interpretation of support vector machines

- To use support vector machines we only have to specify the inner products (or *kernel*) between the examples $(\mathbf{x}_i^T \mathbf{x})$
- The weights $\{\alpha_i\}$ associated with the training examples are solved by enforcing the classification constraints.

  $\Rightarrow$ sparse solution
- We make decisions by comparing each new example $\mathbf{x}$ with **only** the support vectors $\{\mathbf{x}_i\}_{i \in SV}$:

$$\hat{y} = \text{sign}\left(\sum_{i \in SV} \hat{\alpha}_i \, y_i \, (\mathbf{x}_i^T \mathbf{x}) + \hat{w}_0 \right)$$

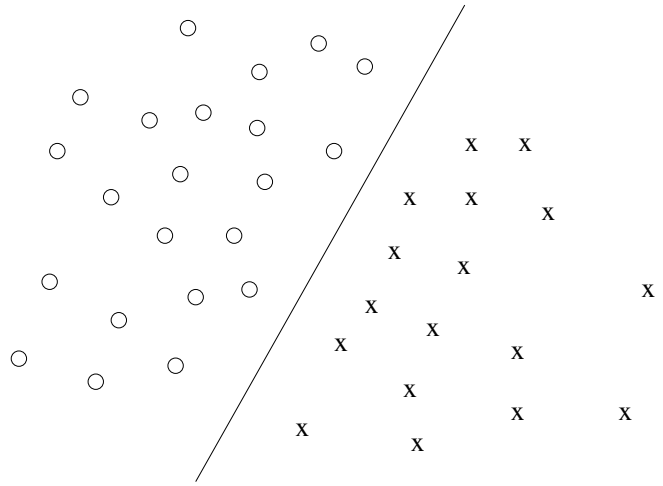
Non-linear classifier

- So far our classifier can make only linear separations
- We can easily obtain a non-linear classifier by mapping our examples $\mathbf{x} = [x_1\ x_2]$ into longer feature vectors $\phi(\mathbf{x})$

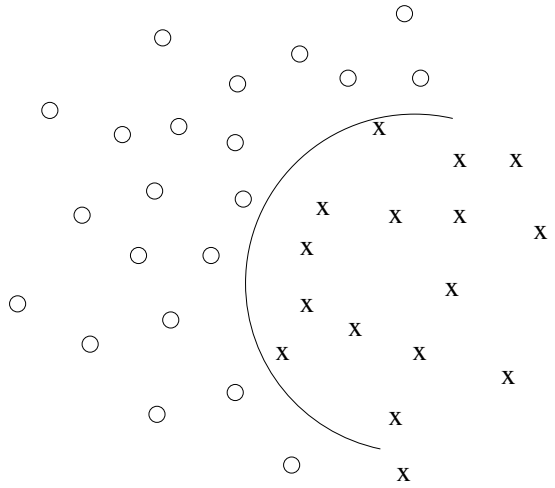
$$\phi(\mathbf{x}) = [x_1^2\ \ x_2^2\ \ \sqrt{2}x_1x_2\ \ \sqrt{2}x_1\ \ \sqrt{2}x_2\ \ 1]$$

  and applying the linear classifier to the new feature vectors $\phi(\mathbf{x})$ instead

Non-linear classifier



Linear separator in the feature space



Non-linear separator in the original space

Feature mapping and kernels

- Let's look at the previous example in a bit more detail

$$\mathbf{x} \rightarrow \phi(\mathbf{x}) = [x_1^2 \;\; x_2^2 \;\; \sqrt{2}x_1x_2 \;\; \sqrt{2}x_1 \;\; \sqrt{2}x_2 \;\; 1]$$

- The SVM classifier deals only with inner products of examples (or feature vectors). In this example,

$$\begin{aligned}\phi(\mathbf{x})^T\phi(\mathbf{x}') &= x_1^2x_1'^2 + x_2^2x_2'^2 + 2x_1x_2x_1'x_2' + 2x_1x_1' + 2x_2x_2' + 1 \\ &= (1 + x_1x_1' + x_2x_2')^2 \\ &= \left(1 + (\mathbf{x}^T\mathbf{x}')\right)^2\end{aligned}$$

  so the inner products can be evaluated without ever explicitly constructing the feature vectors $\phi(\mathbf{x})$!

- $K(\mathbf{x}, \mathbf{x}') = \left(1 + (\mathbf{x}^T\mathbf{x}')\right)^2$ is a *kernel function* (inner product in the feature space)

Examples of kernel functions

- **Linear kernel**

$$K(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T \mathbf{x}')$$

- **Polynomial kernel**

$$K(\mathbf{x}, \mathbf{x}') = \left(1 + (\mathbf{x}^T \mathbf{x}')\right)^p$$

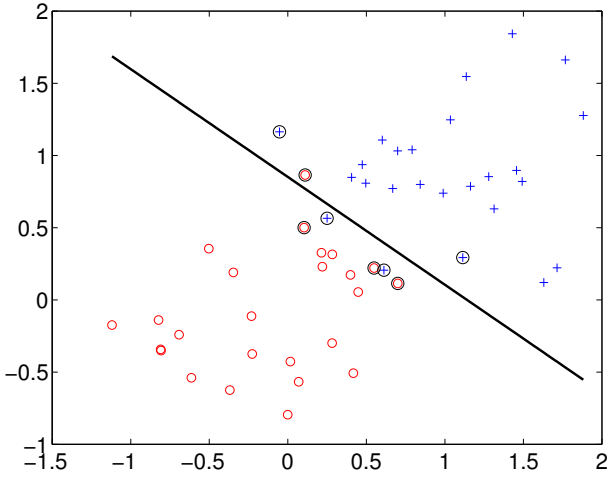
  where $p = 2, 3, \ldots$. To get the feature vectors we concatenate all up to p^{th} order polynomial terms of the components of $\mathbf{x}$ (weighted appropriately)

- **Radial basis kernel**

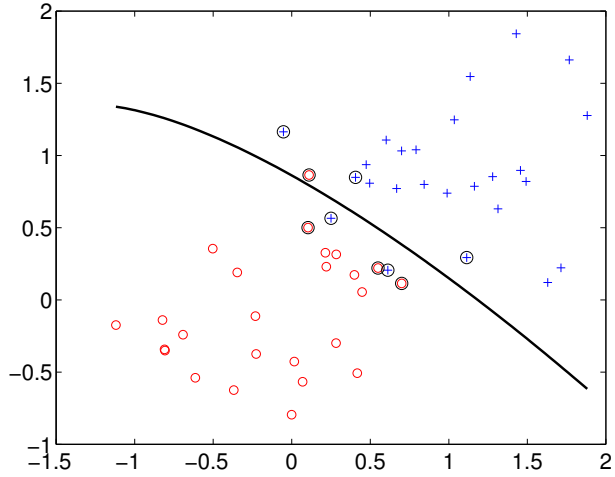
$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{1}{2}\|\mathbf{x} - \mathbf{x}'\|^2\right)$$

  In this case the feature space consists of functions and results in a *non-parametric* classifier.

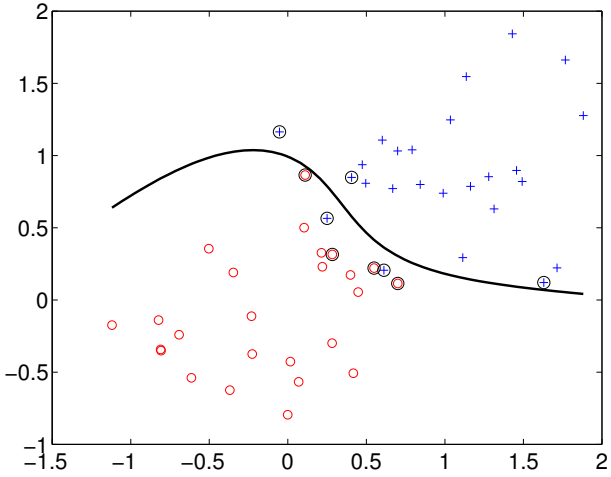
SVM examples



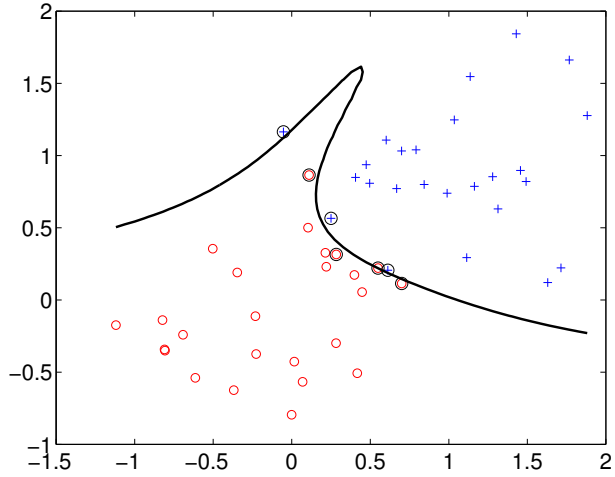
linear



2^{nd} order polynomial

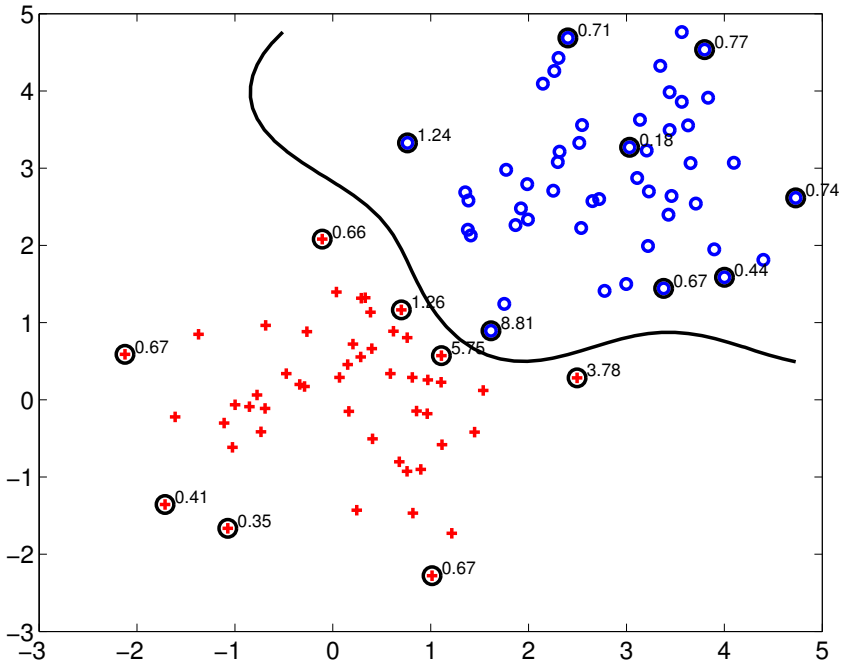


4^{th} order polynomial



8^{th} order polynomial

SVM examples cont'd



Radial basis kernel

- The support vectors need not appear close to the boundary in the input space, only in the feature space

Dimensionality and complexity

- Example: even for small values of p the polynomial kernel

$$K(\mathbf{x}, \mathbf{x}') = \left(1 + (\mathbf{x}^T \mathbf{x}')\right)^p$$

corresponds to long feature vectors $\phi(\mathbf{x})$.

In two dimensions:

degree p	# of features
2	6
3	10
4	15
5	21

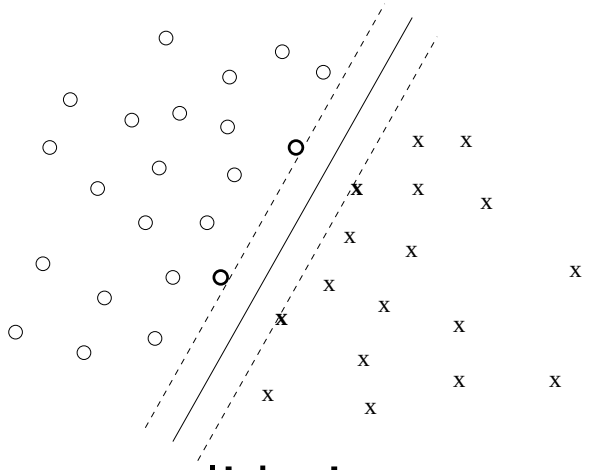
In three dimensions

degree p	# of features
2	10
3	20
4	35
5	56

(it gets much worse in higher dimensions)

- The dimensionality of the feature space does not tell the whole story

Cross-validation error



- The leave-one-out cross-validation error does not depend on the dimensionality of the feature space but only on the # of support vectors!

$$\text{Leave-one-out CV error} \leq \frac{\#\text{ support vectors}}{\#\text{ of training examples}}$$

SVM examples

- Digit recognition example (16x16 grayscale pixel images)

Method	error %
SVM (4^{th} order polynomial)	1.1
LeNet 1 (neural network)	1.7
LeNet 4 (neural network)	1.1
Tangent distance (template matching)	0.7

- Document classification etc.

Logistic regression and kernels

- Kernels are not specific to support vector machines. Indeed, we can kernels with logistic regression models.
- Consider the regularized logistic regression model

$$l_c(D; w_0, \mathbf{w}) = \sum_{i=1}^{n} \log P(\tilde{y}_i | \mathbf{x}_i, w_0, \mathbf{w}) - \frac{\lambda}{2} \|\mathbf{w}\|^2$$

  where $\tilde{y} \in \{0, 1\}$ (back to 0/1 labels) and

$$P(\tilde{y} = 1 | \mathbf{x}, w_0, \mathbf{w}) = g(\, w_0 + \mathbf{w}^T \mathbf{x}\,)$$

- To use kernels we'd have to show that both the objective and the predictions can be expressed solely in terms of inner products (for the optimal $\mathbf{w}$).

Logistic regression and kernels cont'd

- The optimimality condition for $\mathbf{w}$ gives

$$\frac{\partial}{\partial \mathbf{w}} l_c(D; w_0, \mathbf{w}) = \sum_{i=1}^{n} \overbrace{(\tilde{y}_i - P(\tilde{y}_i|\mathbf{x}_i, w_0, \mathbf{w}))}^{\epsilon_i} \mathbf{x}_i - \lambda \mathbf{w} = 0$$

  so that $\mathbf{w} = \sum_i (\epsilon_i/\lambda)\mathbf{x}_i$.

- Substituting this back into the model we get

$$\begin{aligned} P(\tilde{y} = 1|\mathbf{x}, w_0, \mathbf{w}) &= g(\, w_0 + \mathbf{w}^T \mathbf{x}\,) \\ &= g\left(w_0 + \sum_i (\epsilon_i/\lambda)(\mathbf{x}_i^T \mathbf{x}) \right) \end{aligned}$$

  which is expressed solely in terms of inner products and may be replaced with kernels.

- The solution is not sparse, however.