Support Vector Machines

Lecturer:
Jiří Matas

Authors:
Ondřej Drbohlav, Jiří Matas

Centre for Machine Perception
Czech Technical University, Prague
http://cmp.felk.cvut.cz

Slide credits:
Alexander Apartsin

12.11.2018
A Linear Classifier

Classification according to signum of an affine function of $x$:

$$q(x) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b)$$

(1)

A solution for $\{\mathbf{w}, b\}$ correctly classifying the training set:
Maximum Margin Linear Classifier

- Let $d(x)$ denote the distance of a point $x \in \mathcal{T}$ from the training set $\mathcal{T}$ to the decision boundary of a linear classifier given by parameters $(\mathbf{w}, b)$.
- The margin $m$ of a linear classifier $(\mathbf{w}, b)$ is defined as follows:
  
  (i) If the classifier classifies all data correctly then $m = 2 \min_{x \in \mathcal{T}} d(x)$.
  
  Points $x \in \mathcal{T}$ satisfying $m = 2d(x)$ are called support vectors.
  
  (ii) If the classifier has non-zero error on $\mathcal{T}$ then $m = 0$.
- **Goal**: Find the classifier $(\mathbf{w}^*, b^*)$ maximizing the margin. Vapnik justifies the use of maximum margin from the viewpoint of Structural Risk Minimization.
Maximizing Margin, Formulation

Let us define signed distance $d(x, y)$ of a point $x$ belonging to class $y \in \{1, -1\}$ to the decision boundary of classifier $(w, b)$:

$$d(x, y) = \frac{y(w \cdot x + b)}{||w||}$$

We search for $(w, b)$ such that $d(x, y) > 0$ for all training data (all training points are in their class' half-space). This is equivalent to $y(w \cdot x + b) > 0$.

Optimization task:

$$(w^*, b^*) = \operatorname{argmax}_{w, b} \min_{(x, y) \in T} 2d(x, y)$$

subject to:

$$y(w \cdot x + b) > 0, \forall (x, y) \in T$$  \hspace{1cm} (C)
Maximizing Margin, Scale Ambiguity

There is a scale ambiguity in the parameters \((w, b)\). Any feasible \((w, b)\) (that is, satisfying Eq. (C)) can be multiplied by a positive constant \((w, b) \rightarrow (\sigma w, \sigma b)\), and:

(i) feasibility does not change, as

\[
y(\sigma w \cdot x + \sigma b) = \sigma y(w \cdot x + b) > 0 \iff y(w \cdot x + b) > 0,
\]

and

(ii) signed distances do not change, as

\[
d(x, y) = \frac{y(\sigma w \cdot x + \sigma b)}{\|\sigma w\|} = \frac{y(w \cdot x + b)}{\|w\|}.
\]

Optimization task:

\[
(w^*, b^*) = \arg\max_{w, b} \min_{(x, y) \in \mathcal{T}} 2d(x, y)
\]

subject to:

\[
y(w \cdot x + b) > 0, \forall (x, y) \in \mathcal{T}
\]
Maximizing Margin, Fixing Scale

- Constraints \( y(w \cdot x + b) > 0 \) are equivalent to \( y(w \cdot x + b) \geq \epsilon \) (with \( \epsilon > 0 \))
- Break the scale ambiguity by setting \( \epsilon = 1 \):

\[
(w^*, b^*) = \arg\max_{w, b} \min_{(x, y) \in \mathcal{T}} 2d(x, y)
\]

subject to: \( y(w \cdot x + b) \geq 1, \forall (x, y) \in \mathcal{T} \)  \( (5) \)

Optimization task (original):

\[
(w^*, b^*) = \arg\max_{w, b} \min_{(x, y) \in \mathcal{T}} 2d(x, y)
\]

subject to:

\( y(w \cdot x + b) > 0, \forall (x, y) \in \mathcal{T} \)  \( (C) \)

\[
d(x, y) = \frac{y(w \cdot x + b)}{\|w\|}
\]
Maximizing Margin, Final Optimization Formulation (1)

That is, all points must be outside the strip delineated by the two lines $w \cdot x + b = 1$ and $w \cdot x + b = -1$. The width of this strip is $\frac{2}{\|w\|}$. It follows that the maximum margin $m^*$ is

$$m^* = \max_{w,b} \min_{(x,y) \in T} 2d(x, y) = \max_{w,b} \frac{2}{\|w\|}$$

subject to: $y(w \cdot x + b) \geq 1, \forall (x, y) \in T$ (6)

Optimization task (original):

$$(w^*, b^*) = \arg\max_{w,b} \min_{(x,y) \in T} 2d(x, y)$$

subject to:
$$y(w \cdot x + b) > 0, \forall (x, y) \in T$$

(C)

$$d(x, y) = \frac{y(w \cdot x + b)}{\|w\|}$$
Maximizing Margin, Final Optimization Formulation (2)

- That is, all points must be outside the strip delineated by the two lines $\mathbf{w} \cdot \mathbf{x} + b = 1$ and $\mathbf{w} \cdot \mathbf{x} + b = -1$. The width of this strip is $\frac{2}{\|\mathbf{w}\|}$. It follows that the maximum margin $m^*$ is

$$m^* = \max_{\mathbf{w},b} \min_{(x,y) \in T} 2d(x,y) = \max_{\mathbf{w},b} \frac{2}{\|\mathbf{w}\|}$$

subject to: $y(\mathbf{w} \cdot \mathbf{x} + b) \geq 1, \forall (\mathbf{x},y) \in T$ (7)

- There holds: $\arg\max_{\mathbf{w}} \frac{2}{\|\mathbf{w}\|} = \arg\min_{\mathbf{w}} \|\mathbf{w}\| = \arg\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2$. Therefore, the $(\mathbf{w}^*, b^*)$ maximizing the margin are:

$$(\mathbf{w}^*, b^*) = \arg\min_{(\mathbf{w},b)} \frac{1}{2}\|\mathbf{w}\|^2$$

subject to: $y(\mathbf{w} \cdot \mathbf{x} + b) \geq 1, \forall (\mathbf{x},y) \in T$ (8)

- This is a Quadratic Programming (QP) problem (more generally, it is minimization of a convex function on a convex domain.)
SVM, Example (1D)

- Class -1: Points at x = -2, 0
- Class 1: Points at x = 2, 3

The diagram shows the decision boundary for a linear SVM in 1D space. The feasible region is where the inequalities $w \cdot x + b \leq -1$ and $w \cdot x + b \geq 1$ are satisfied, indicating the hyperplane and its margins.
SVM, Example (1D), Result

$wx + b = x - 1 = 0$

-2 0 2 3

\( w \cdot x + b \leq -1 \)

\( w \cdot 0 + b \leq -1 \)

\( (w^*, b^*) = \text{argmin}_{w,b} \frac{1}{2} w^2 \)

\( w^* = 1, b^* = -1 \)
SVM, Primal Problem

The derived optimization problem for $w$ and $b$ is

$$(w^*, b^*) = \arg\min_{(w, b)} \frac{1}{2} \|w\|^2$$

subject to: $y(w \cdot x + b) \geq 1, \forall (x, y) \in T \tag{9}$$

It is called *primal* problem. We will also soon derive the *dual* problem. For now, note that the above optimization task can be equivalently regarded as solving an unconstrained problem (this observation will become handy when deriving the dual problem):

$$(w^*, b^*) = \arg\min_{(w, b)} \left\{ \frac{1}{2} \|w\|^2 + \sum_{(x, y) \in T} f(x, y, w, b) \right\}, \text{ where}$$

$$f(x, y, w, b) = \begin{cases} 0 & \text{if } y(w \cdot x + b) \geq 1, \\ \infty & \text{otherwise} \end{cases} \tag{10}$$

Note that $f(x, y, w, b)$ for a given $(x, y)$ is a convex function of $w, b$. 
The Dual Formulation (1)

Start with just discussed primal formulation. Let \( \mathcal{T} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\} \) be the training set. We want to solve

\[
(w^*, b^*) = \underset{(w,b)}{\text{argmin}} \left\{ \frac{1}{2} \|w\|^2 + \sum_{i=1}^{N} f(x_i, y_i, w, b) \right\},
\]

where

\[
f(x_i, y_i, w, b) = \begin{cases} 
0 & \text{if } y_i (w \cdot x_i + b) \geq 1, \\
\infty & \text{otherwise} 
\end{cases}
\]

This is the same as (\(\alpha_i\)'s are non-negative multipliers):

\[
(w^*, b^*) = \underset{w, b}{\text{argmin}} \left\{ \frac{1}{2} \|w\|^2 + \max_{\{\alpha_i\}} \left( -\sum_{i=1}^{N} \alpha_i [y_i (w \cdot x_i + b) - 1] \right) \right\}.
\]

because

\[
y_i (w \cdot x_i + b) > 1 \Rightarrow \max_{\alpha_i} (-\alpha_i [y_i (w \cdot x_i + b) - 1]) = 0 \quad \text{for } \alpha_i = 0,
\]

\[
y_i (w \cdot x_i + b) < 1 \Rightarrow \max_{\alpha_i} (-\alpha_i [y_i (w \cdot x_i + b) - 1]) = \infty \quad \text{for } \alpha_i = \infty,
\]

\[
y_i (w \cdot x_i + b) = 1 \Rightarrow \max_{\alpha_i} (-\alpha_i [y_i (w \cdot x_i + b) - 1]) = 0 \quad \text{for any } \alpha_i \geq 0.
\]
The Dual Formulation (2)

This is in turn the same as

$$(w^*, b^*) = \arg\min_{w,b} \max_{\{\alpha_i\}_{\alpha_i \geq 0}^{i \in \{1, \ldots, N\}}} \left\{ \frac{1}{2} \|w\|^2 - \sum_{i=1}^{N} \alpha_i [y_i(w \cdot x_i + b) - 1] \right\}. \quad (17)$$

There holds, in full generality, that $\max_p \min_q f(p, q) \leq \min_q \max_p f(p, q)$. For our case,

$$\min_{w,b} \max_{\{\alpha_i\}_{\alpha_i \geq 0}^{i \in \{1, \ldots, N\}}} \left\{ \frac{1}{2} \|w\|^2 - \sum_{i=1}^{N} \alpha_i [y_i(w \cdot x_i + b) - 1] \right\} \geq \max_{\{\alpha_i\}_{\alpha_i \geq 0}^{i \in \{1, \ldots, N\}}} \min_{w,b} \left\{ \frac{1}{2} \|w\|^2 - \sum_{i=1}^{N} \alpha_i [y_i(w \cdot x_i + b) - 1] \right\} \quad (18)$$

This is the essence of converting the primal problem to the dual one. And, our case is even better: strong duality holds, and the two terms are equal (duality gap is zero). Denote the inner term by $L(w, b, \alpha)$ (corresponds to what's commonly known as the Lagrangian):

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^{N} \alpha_i [y_i(w \cdot x_i + b) - 1] \quad (19)$$
The Dual Formulation (3)

\[
L(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \alpha_i [y_i(w \cdot x_i + b) - 1]
\] (20)

We want to find \(\arg\max_{\alpha \geq 0} \min_{w,b} L(w, b, \alpha)\). First, for fixed \(\alpha\), find \(\min_{w,b} L(w, b, \alpha)\):

\[
\frac{\partial L}{\partial w} = w - \sum_{i=1}^{N} \alpha_i y_i x_i = 0 \Rightarrow w = \sum_{i=1}^{N} \alpha_i y_i x_i
\] (21)

\[
\frac{\partial L}{\partial b} = \sum_{i=1}^{N} \alpha_i y_i = 0
\] (22)

Put this to Lagrangian:

\[
L(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \alpha_i [y_i(w \cdot x_i + b) - 1] =
\] (23)

\[
= \frac{1}{2}\|w\|^2 - \left(\sum_{i=1}^{N} \alpha_i y_i x_i\right) \cdot w - \sum_{i=1}^{N} \alpha_i y_i b + \sum_{i=1}^{N} \alpha_i
\] (24)

\[
= -\frac{1}{2}\|w\|^2 + \sum_{i=1}^{N} \alpha_i = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j x_i \cdot x_j
\] (25)
The Dual Formulation, Result and Insights

The dual optimization problem:

$$\alpha = \arg\max_{\alpha} \left( \min_{\mathbf{w}, b} L(\mathbf{w}, b, \alpha) \right) = \arg\max_{\alpha} \left\{ \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \right\} \quad (26)$$

subject to: $\sum_{i} \alpha_i y_i = 0; \; \alpha_i \geq 0, \; \forall i \in \{1, 2, ..., N\}$ \hspace{1cm} (27)

- Number of optimization variables $\alpha_i$’s is $N$ (the number of training data). But at the solution, all $\alpha_i$’s but those of support vectors are zero.
- Once the solution is obtained, the primal variables can be computed as

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i \quad \text{only support vectors (}\alpha_i > 0\text{) contribute} \quad (28)$$

$$y^S[\mathbf{w} \cdot \mathbf{x}^S + b] = 1 \text{ for any support vector } (\mathbf{x}^S, y^S) \Rightarrow b = y^S - \mathbf{w} \cdot \mathbf{x}^S \quad (29)$$

- The discriminant function $\mathbf{w} \cdot \mathbf{x} + b$ thus takes the form ($P$ are indices of all support vectors):

$$\mathbf{w} \cdot \mathbf{x} + b = \sum_{i \in P} \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{x}) + y^S - \sum_{i \in P} \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{x}^S) \quad (30)$$

constant, independent of $\mathbf{x}$

- Both the dual classification problem and the discriminant function involve data points only in the form of dot products.
The Dual Problem, Example (1)

Consider the 3 points as below

Objective: maximize

\[
\alpha_1 + \alpha_2 + \alpha_3 - \frac{1}{2} \begin{bmatrix}
\alpha_1 \\
\alpha_2 \\
\alpha_3
\end{bmatrix}^T \begin{bmatrix}
y_1 y_1 x_1 \cdot x_1 & y_1 y_2 x_1 \cdot x_2 & y_1 y_3 x_1 \cdot x_3 \\
y_2 y_1 x_2 \cdot x_1 & y_2 y_2 x_2 \cdot x_2 & y_2 y_3 x_2 \cdot x_3 \\
y_3 y_1 x_3 \cdot x_1 & y_3 y_2 x_3 \cdot x_2 & y_3 y_3 x_3 \cdot x_3
\end{bmatrix} \begin{bmatrix}
\alpha_1 \\
\alpha_2 \\
\alpha_3
\end{bmatrix}
\]

subject to: \( \alpha_1, \alpha_2, \alpha_3 \geq 0; \) \( \alpha_1 + \alpha_2 - \alpha_3 = 0 \)

\[
\begin{array}{c}
\text{class 1} \\
\bullet x_2 = (1, 2) \\
\bullet x_1 = (0, 1) \\
\text{class -1} \\
\bullet x_3 = (0, -1)
\end{array}
\]
The Dual Problem, Example (2)

Consider the 3 points as below

Objective: maximize

\[
\alpha_1 + \alpha_2 + \alpha_3 - \frac{1}{2} \begin{bmatrix}
\alpha_1 \\
\alpha_2 \\
\alpha_3 
\end{bmatrix}^T \begin{bmatrix}
1 & 2 & 1 \\
2 & 5 & 2 \\
1 & 2 & 1
\end{bmatrix} \begin{bmatrix}
\alpha_1 \\
\alpha_2 \\
\alpha_3
\end{bmatrix}
\]

subject to: \( \alpha_1, \alpha_2, \alpha_3 \geq 0; \ \alpha_1 + \alpha_2 - \alpha_3 = 0 \)

\[\text{class 1} \quad \bullet \ \mathbf{x}_2 = (1, 2) \]

\[\text{class } -1 \quad \bullet \ \mathbf{x}_1 = (0, 1) \]

\[\bullet \ \mathbf{x}_3 = (0, -1) \]
The Dual Problem, Example (3)

Substitute $\alpha_3 = \alpha_1 + \alpha_2$ and search for solution as a problem in $\alpha_1, \alpha_2$. After some straightforward computation, the original problem turns to:

$$\text{maximize } 2(\alpha_1 + \alpha_2) - \frac{1}{2} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix}^T \begin{bmatrix} 4 & 6 \\ 6 & 10 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix}$$

subject to: $\alpha_1, \alpha_2 \geq 0$. **Solution:** $(\alpha_1, \alpha_2) = \left(\frac{1}{2}, 0\right)$, $\alpha_3 = \frac{1}{2} + 0 = \frac{1}{2}$. 

![Diagram showing the feasible region, solution, and points for class 1 and class -1.](image-url)
The Dual Problem, Example, Result

Result: \((\alpha_1, \alpha_2, \alpha_3) = (\frac{1}{2}, 0, \frac{1}{2})\). The support vectors are \(x_1\) and \(x_3\) because their \(\alpha_i > 0\).

Vector \(w = \sum_{i=1,3} \alpha_i y_i x_i = \frac{1}{2}(0, 1) - \frac{1}{2}(0, -1) = (0, 1)\).

Offset \(b = y^S - wx^S = 1 - wx_1 = -1 - wx_3 = 0\).

Decision boundary \((0, 1)^T \cdot x = 0\).
Soft Margin SVM

If the data are not linearly separable, *slack variables* $\xi_i$ need to be introduced.

- Position and size of margin is implied by $w$ and $b$, as before.
- If a point $(x, y)$ fulfills the condition $y(w \cdot x + b) \geq 1$ then no penalty is paid.
- Otherwise, the condition is relaxed to $y(w \cdot x + b) \geq 1 - \xi$ and penalty $C \cdot \xi$ is paid.

$$ (w^*, b^*) = \arg\min_{(w, b)} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{N} \xi_i \quad (31) $$

subject to:

$$ y_i (w \cdot x_i + b) \geq 1 - \xi_i, \quad (32) $$

$$ \xi_i \geq 0, \quad (33) $$

$$ \forall i = 1, \ldots, N $$
Soft Margin SVM

The primal problem

\[(w^*, b^*) = \arg\min_{(w, b)} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{N} \xi_i\]

subject to: \(y_i(w \cdot x_i + b) \geq 1 - \xi_i, \ \forall i = 1, ..., N\) \hspace{1cm} (34)

\(\xi_i \geq 0, \ \forall i = 1, ..., N\) \hspace{1cm} (35)

The dual problem:

\[\alpha = \arg\max_{\alpha} \left\{ \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \right\}\]

subject to: \(\sum_i \alpha_i y_i = 0\) \hspace{1cm} (37)

\(0 \leq \alpha_i \leq C, \ \forall i \in \{1, 2, ..., N\}\) \hspace{1cm} (38)
Linear SVMs: Overview

- The classifier is a *separating hyperplane*.

- Most “important” training points are support vectors; they define the hyperplane.

- Quadratic optimization algorithms can identify which training points \( \mathbf{x}_i \) are support vectors with non-zero Lagrangian multipliers \( \lambda_i \).

- Both in the dual formulation of the problem and in the solution training points appear only inside inner-products.
Who really need linear classifiers

- Datasets that are linearly separable with some noise, linear SVM work well:

- But if the dataset is non-linearly separable?

- How about… mapping data to a higher-dimensional space:
Non-linear SVMs: Feature spaces

- General idea: the original space can always be mapped to some higher-dimensional feature space where the training set becomes separable:

\[ \Phi: \mathbf{x} \rightarrow \phi(\mathbf{x}) \]
The “Kernel Trick”

- The SVM only relies on the inner-product between vectors $\mathbf{x}_i \cdot \mathbf{x}_j$.

- If every datapoint is mapped into high-dimensional space via some transformation $\Phi$: $\mathbf{x} \rightarrow \varphi(\mathbf{x})$, the inner-product becomes:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}_j)$$

- $K(\mathbf{x}_i, \mathbf{x}_j)$ is called the kernel function.

- For SVM, we only need specify the kernel $K(\mathbf{x}_i, \mathbf{x}_j)$, without need to know the corresponding non-linear mapping, $\varphi(\mathbf{x})$. 
Non-linear SVMs

- The dual problem:
  
  Maximizing: \( L(h) = \sum_{i=1}^{N} h_i - \frac{1}{2} h \cdot D \cdot h \)

  Subject to: \( h \cdot y = 0 \)
  \( 0 \leq h \leq C \)

  where \( D_{ij} = y_i y_j K(x_i, x_j) \)

- Optimization techniques for finding \( h_i \)'s remain the same!
- The solution is:

  \[
  w^* = \sum_{i \in SV} h_i y_i \varphi(x_i) \\
  f(x) = w^* \cdot \varphi(x) + b^* \\
  = \sum_{i \in SV} h_i y_i K(x_i, x) + b^* 
  \]
Examples of Kernel Trick (1)

- For the example in the previous figure:
  - The non-linear mapping
    \[ x \rightarrow \varphi(x) = (x, x^2) \]
  - The kernel
    \[ \varphi(x_i) = (x_i, x_i^2), \quad \varphi(x_j) = (x_j, x_j^2) \]
    \[ K(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j) \]
    \[ = x_i x_j (1 + x_i x_j) \]

- Where is the benefit?
Examples of Kernel Trick (2)

- Polynomial kernel of degree 2 in 2 variables
  - The non-linear mapping:
    \[ \mathbf{x} = (x_1, x_2) \]
    \[ \varphi(\mathbf{x}) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2) \]
  - The kernel
    \[ \varphi(\mathbf{x}) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2) \]
    \[ \varphi(\mathbf{y}) = (1, \sqrt{2}y_1, \sqrt{2}y_2, y_1^2, y_2^2, \sqrt{2}y_1y_2) \]
    \[ K(\mathbf{x}, \mathbf{y}) = \varphi(\mathbf{x}) \cdot \varphi(\mathbf{y}) \]
    \[ = (1 + \mathbf{x} \cdot \mathbf{y})^2 \]
Examples of kernel trick (3)

- Gaussian kernel:
  \[ K(x_i, x_j) = e^{-\frac{(x_i - x_j)^2}{2\sigma^2}} \]

- The mapping is of infinite dimension:
  \[ \varphi(x) = (\ldots, \varphi_\omega(x), \ldots), \text{ for } \omega \in \mathbb{R}^d \]
  \[ \varphi_\omega(x) = Ae^{-B\omega^2} e^{-iwx} \]
  \[ K(x, y) = \int \varphi_\omega(x) \varphi_\omega^*(y) d\omega \]

- The moral: very high-dimensional and complicated non-linear mapping can be achieved by using a simple kernel!
What Functions are Kernels?

- For some functions $K(x_i, x_j)$ checking that $K(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j)$ can be cumbersome.
- Mercer’s theorem:

Every semi-positive definite symmetric function is a kernel
Examples of Kernel Functions

- **Linear kernel:**
  \[
  K(x_i, x_j) = x_i \cdot x_j
  \]

- **Polynomial kernel of power \( p \):**
  \[
  K(x_i, x_j) = (1 + x_i \cdot x_j)^p
  \]

- **Gaussian kernel:**
  \[
  K(x_i, x_j) = e^{-\frac{||x_i - x_j||^2}{2\sigma^2}}
  \]
  In the form, equivalent to RBFNN, but has the advantage of that the center of basis functions, i.e., support vectors, are optimized in a supervised.

- **Two-layer perceptron:**
  \[
  K(x_i, x_j) = \tanh(\alpha x_i \cdot x_j + \beta)
  \]
Let \( d \in \mathbb{N} \) and \( x = [x_1, x_2, \ldots, x_D]^\top \in \mathbb{R}^D \).

Let \( \phi_d(x) \) denote the mapping which lifts \( x \) to the space containing all monomials of degree \( d' \), \( 1 \leq d' \leq d \) in the components of \( x \):

For example, when \( x = [x_1, x_2]^\top \in \mathbb{R}^2 \),

\[
\begin{align*}
\phi_1(x) &= [x_1, x_2]^\top, \\
\phi_2(x) &= [x_1, x_2, x_1^2, x_1x_2, x_2^2]^\top, \\
\phi_3(x) &= [x_1, x_2, x_1^2, x_1x_2, x_2^2, x_1^3, x_1^2x_2, x_1x_2^2, x_2^3]^\top.
\end{align*}
\]

The number of monomials of degree \( d' \) of \( x \in \mathbb{R}^D \) is \( \binom{d' + D - 1}{d'} \). The dimensionality \( L \) of the output space of \( \phi_d(x) \) is thus

\[
L = \sum_{d'=1}^{d} \binom{d' + D - 1}{d'}. \tag{42}
\]
**Lifting Dimension by Polynomial Mapping of Degree $d$**

Feature space dimensionality $D$, lifting by $\phi_d(x)$

<table>
<thead>
<tr>
<th>$d$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>5</td>
<td>9</td>
<td>14</td>
<td>20</td>
<td>27</td>
<td>35</td>
<td>44</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>9</td>
<td>19</td>
<td>34</td>
<td>55</td>
<td>83</td>
<td>119</td>
<td>164</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>14</td>
<td>34</td>
<td>69</td>
<td>125</td>
<td>209</td>
<td>329</td>
<td>494</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>20</td>
<td>55</td>
<td>125</td>
<td>251</td>
<td>461</td>
<td>791</td>
<td>1286</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>27</td>
<td>83</td>
<td>209</td>
<td>461</td>
<td>923</td>
<td>1715</td>
<td>3002</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>35</td>
<td>119</td>
<td>329</td>
<td>791</td>
<td>1715</td>
<td>3431</td>
<td>6434</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>44</td>
<td>164</td>
<td>494</td>
<td>1286</td>
<td>3002</td>
<td>6434</td>
<td>12869</td>
</tr>
</tbody>
</table>
Lifting by Polynomial Mapping of Degree $d$, Example

\[ d = 1, \dim(\phi_d(\mathbf{x})) = 2 \]

support vectors : 3

\[ f(\mathbf{x}) = \mathbf{w} \cdot \phi_d(\mathbf{x}) + b \]

\[ d = 2, \dim(\phi_d(\mathbf{x})) = 5 \]

support vectors : 5

\[ f(\mathbf{x}) = \mathbf{w} \cdot \phi_d(\mathbf{x}) + b \]
Lifting by Polynomial Mapping of Degree $d$, Example

$d = 3$, $\dim(\phi_d(x)) = 9$

support vectors: 5

\[ f(x) = \mathbf{w} \cdot \phi_d(x) + b \]

$d = 4$, $\dim(\phi_d(x)) = 14$

support vectors: 6

\[ f(x) = \mathbf{w} \cdot \phi_d(x) + b \]
SVM Overviews

Main features:

- By using the kernel trick, data is mapped into a high-dimensional feature space, without introducing much computational effort;
- Maximizing the margin achieves better generation performance;
- Soft-margin accommodates noisy data;
- Not too many parameters need to be tuned.

Demos(https://svm.dcs.rhbnc.ac.uk/pagesnew/GPat.shtml)
SVM so far

- SVMs were originally proposed by Boser, Guyon and Vapnik in 1992 and gained increasing popularity in late 1990s.
- SVMs are currently among the best performers for many benchmark datasets.
- SVM techniques have been extended to a number of tasks such as regression [Vapnik et al. ’97].
- Most popular optimization algorithms for SVMs are SMO [Platt ’99] and SVMlight [Joachims’ 99], both use decomposition to handle large size datasets.
- It seems the kernel trick is the most attracting site of SVMs. This idea has now been applied to many other learning models where the inner-product is concerned, and they are called ‘kernel’ methods.
- Tuning SVMs remains to be the main research focus: how to an optimal kernel? Kernel should match the smooth structure of data.
Appendix

Online demo: http://cs.stanford.edu/people/karpathy/svmjs/demo/