Support vector machines

April 23, 2018

Table of Contents

Table of Contents

Support vector machines

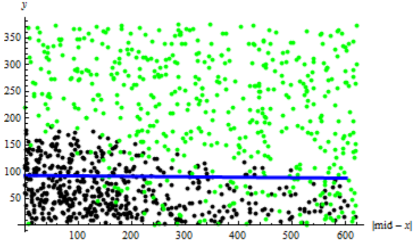
- Classifier derived from statistical learning theory by Vapnik et al in 1992
- widely used in object detection and recognition
- can handle large learning sets in high dimensional feature spaces

Discriminant function

Feature vector $\mathbf{x}$

Two-category case

- Categories ω_1, ω_2
- Decide ω_1 if $g(\mathbf{x}) > 0$, otherwise decide ω_2



Multi-category case

- Categories $\omega_1, \omega_2, \ldots, \omega_n$
- Decide ω_i if $g_i(\mathbf{x}) > g_j(\mathbf{x})$, for all $j \neq i$

Discriminant functions

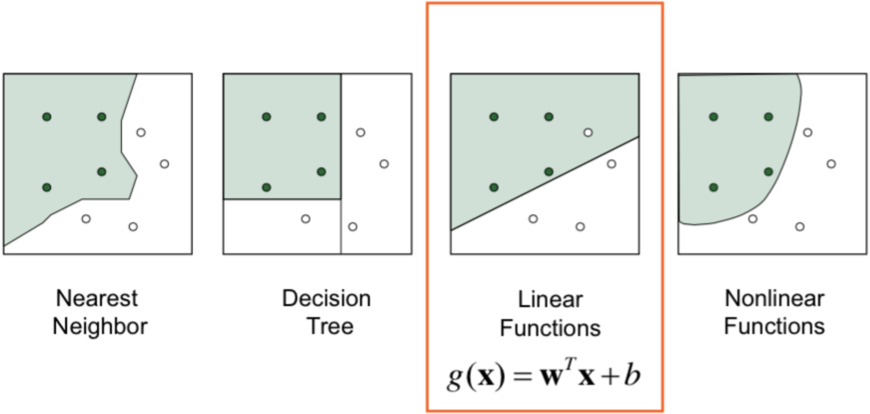


Table of Contents

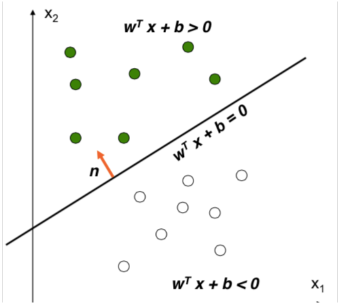
Linear discriminant functions

- $g(\mathbf{x})$ an affine function:

$$g(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$$

- $g(\mathbf{x}) = 0$ defines hyperplane
- b is called the *intercept*
- Unit-length normal of hyperplane:

$$\mathbf{n} = \frac{\mathbf{w}}{||\mathbf{w}||}$$



Linear discriminant functions

- There are infinitely many ways to choose discriminant function
- Which one is best?

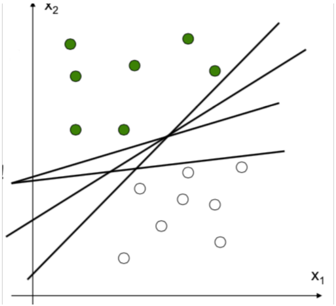
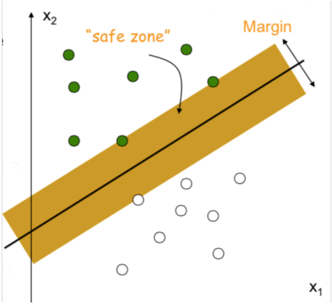


Table of Contents

Maximum margin linear classifier

- Linear discriminant function with maximal margin
- Why this criterion? Robust to outliers and good generalization properties
- Good generalization: works (almost) as well for test set as for training set

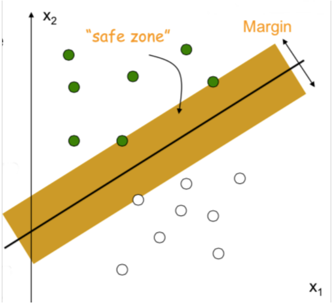


Finding the weights

- **Input**: a set of labeled data points $(\mathbf{x}_i, y_i)$, $i = 1, \ldots, n$, with $y_i \in \{+1, -1\}$
- Find weight vector $\mathbf{w}$ and scalar b such that

$$\mathbf{w}^T \mathbf{x}_i + b > 0, \text{ for } y_i = +1$$

$$\mathbf{w}^T \mathbf{x}_i + b < 0, \text{ for } y_i = -1$$



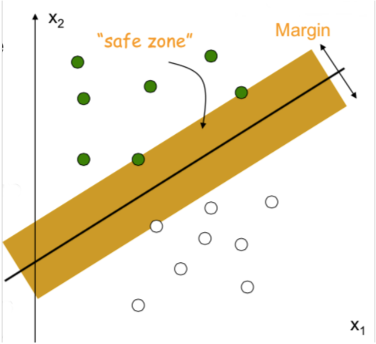
Finding the weights

- After a scale transformation this is equivalent to

$$\mathbf{w}^T\mathbf{x}_i + b \geq 1, \text{ for } y_i = +1$$

$$\mathbf{w}^T\mathbf{x}_i + b \leq 1, \text{ for } y_i = -1$$

where we introduced a margin $[-1, 1]$



Finding the weights

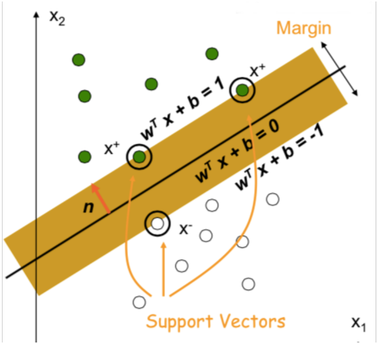
- To obtain a maximum margin there must be vectors $\mathbf{x}^+, \mathbf{x}^-$:

$$\mathbf{w}^T\mathbf{x}^+ + b = 1$$

$$\mathbf{w}^T\mathbf{x}^- + b = -1$$

- vectors $\mathbf{x}^+, \mathbf{x}^-$ are called support vectors
- the maximal margin is

$$\begin{aligned} M &= (\mathbf{x}^+ - \mathbf{x}^-)\cdot\mathbf{n} \\ &= (\mathbf{x}^+ - \mathbf{x}^-)\cdot\frac{\mathbf{w}}{||\mathbf{w}||} \\ &= \frac{2}{||\mathbf{w}||} \end{aligned}$$



Mathematical formulation of maximal margin problem

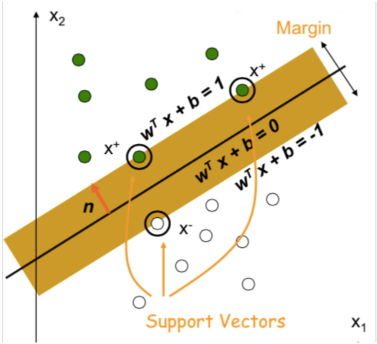
Maximize

$$\frac{2}{||\mathbf{w}||}$$

such that

$$\mathbf{w}^T\mathbf{x}_i + b \geq 1, \text{ for } y_i = +1$$

$$\mathbf{w}^T\mathbf{x}_i + b \leq 1, \text{ for } y_i = -1$$



Mathematical formulation of maximal margin problem

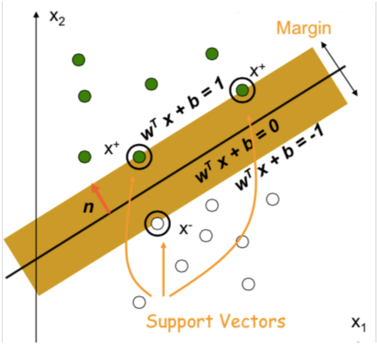
Or equivalently, minimize

$$\frac{1}{2}||\mathbf{w}||^2$$

such that

$$\mathbf{w}^T\mathbf{x}_i + b \geq 1, \text{ for } y_i = +1$$

$$\mathbf{w}^T\mathbf{x}_i + b \leq 1, \text{ for } y_i = -1$$



Mathematical formulation of maximal margin problem

Minimize

$$\frac{1}{2}||\mathbf{w}||^2$$

such that

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$$

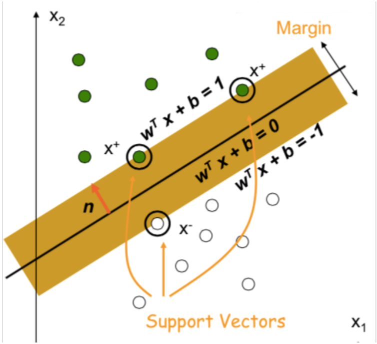


Table of Contents

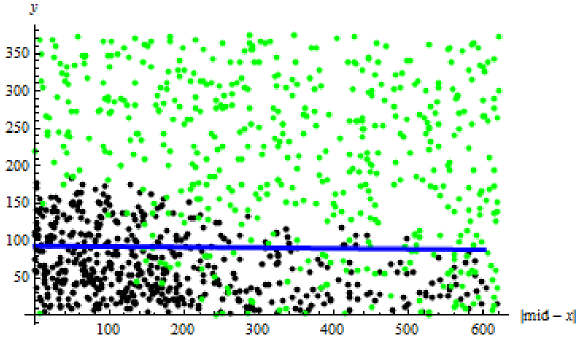
Classifiers

We use the position of the pixel as a feature vector. The position is encoded as

$$(|m - x|, y)$$

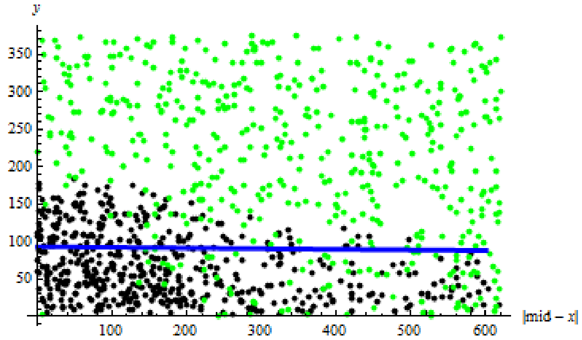
where m denotes the x-coordinate of the center of the image.

Segmentation based on position in image



- feature vector of pixel (x, y) is $(|m - x|, y)$
- black points: feature vectors of road pixels
- green points: feature vectors of environment pixels
- 500 points selected randomly from 15 images as training set

Segmentation based on position in image

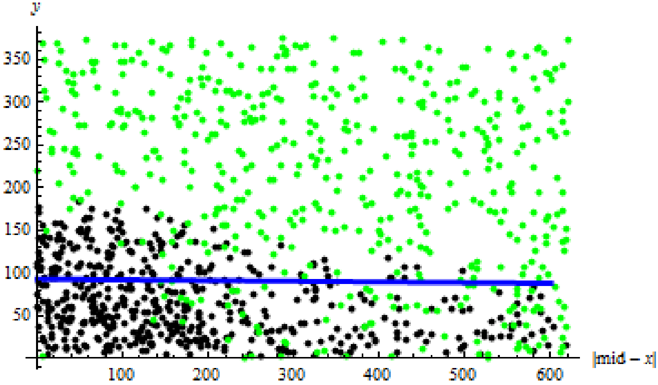


Scores on the training set

- precision = 0.73
- recall = 0.84
- F_1 score = 0.78

These scores measure how well the classifier has learned the training data, not how it will perform in general

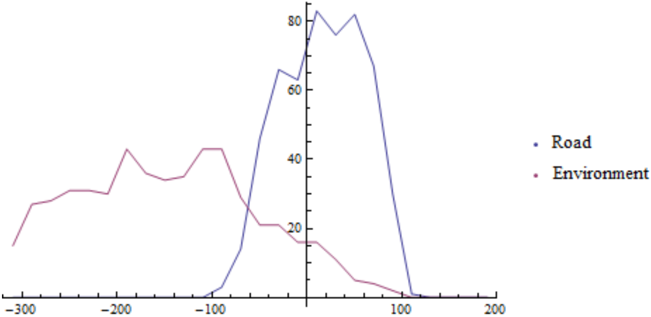
Segmentation based on position in image



The decision surface is not where we expect it:

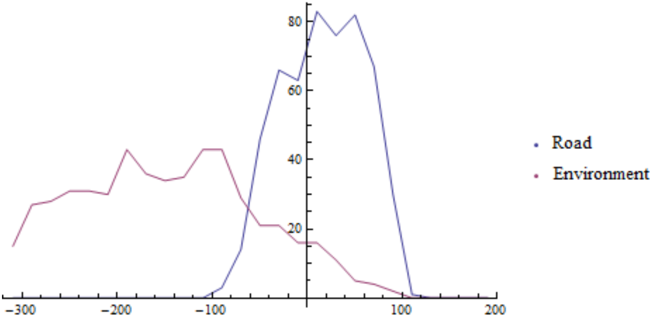
- some support vectors may be unreliable (outliers)
- distribution of misclassified feature vectors is not taken into account

Segmentation based on position



- Histogram of distances from decision surface for both classes
- Decision surface corresponds to vertical line through origin
- Decision surface is at the middle of the max and min arguments of the histograms

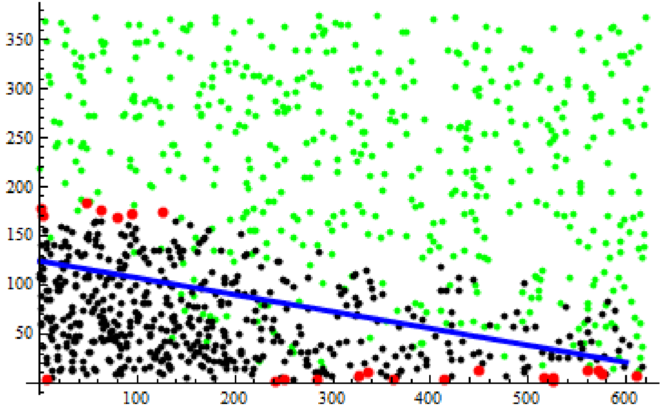
Segmentation based on position



We will handle both effects separately:

- some support vectors may be unreliable (outliers)
- distribution of misclassified feature vectors is not taken into account

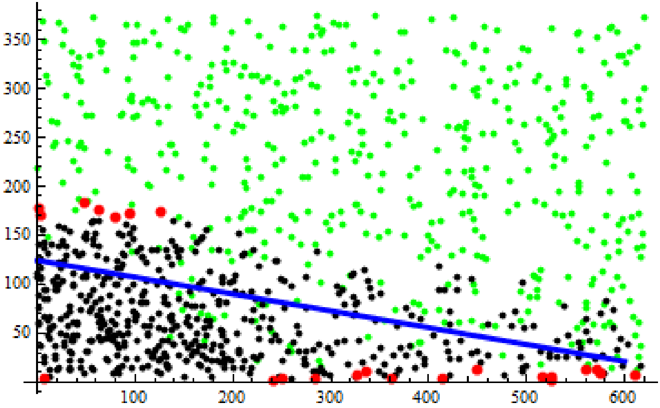
Outlier removal



To obtain a more reliable decision surface:

- remove feature points that are furthest away from the decision surface
- reapply the SVM algorithm

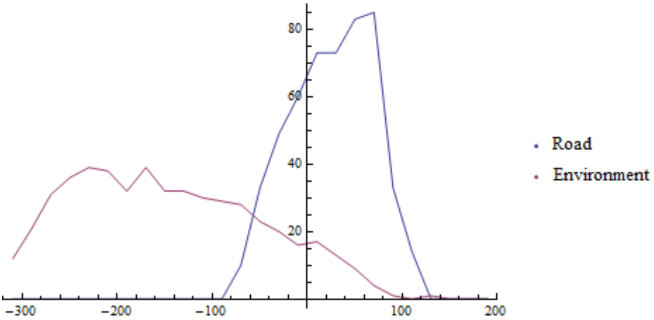
Outlier removal



Scores on the training set

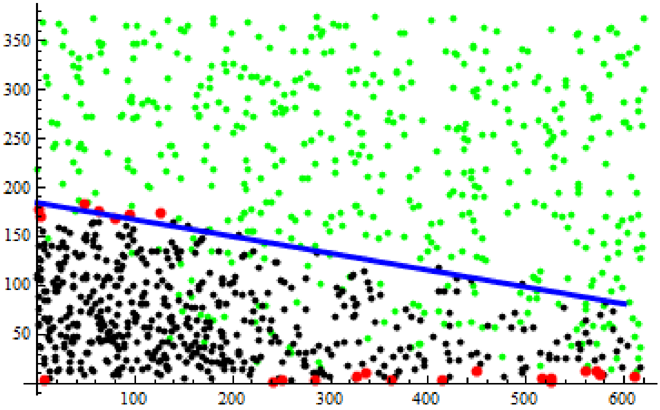
- precision = 0.75
- recall = 0.91
- F_1 score = 0.83

Distance histogram after outlier removal



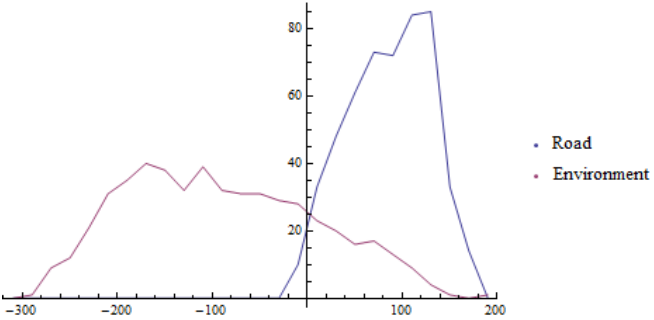
The histograms have changed, because the decision surface has changed

Adapting intercept for optimal F_1 score



- We shift the decision surface until the f_1-score is maximal
- best F_1 score = 0.87

Feature based on position in image

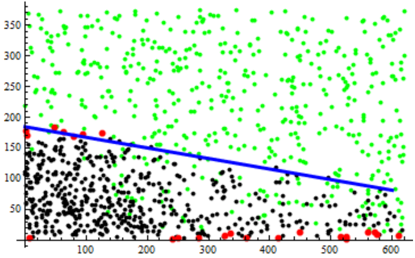


The decision surface is now where we expect it.

What does it mean for images?

- classifier returns the same result for all images
- classifier has learned best position and angle of lines that indicate road boundaries

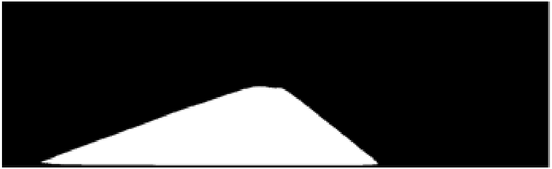
What does it mean for images?



$\rightarrow$

What does it mean for images?

SVM classifier



naive classifier that uses ground truth probabilities

Distribution of road vs environment pixels

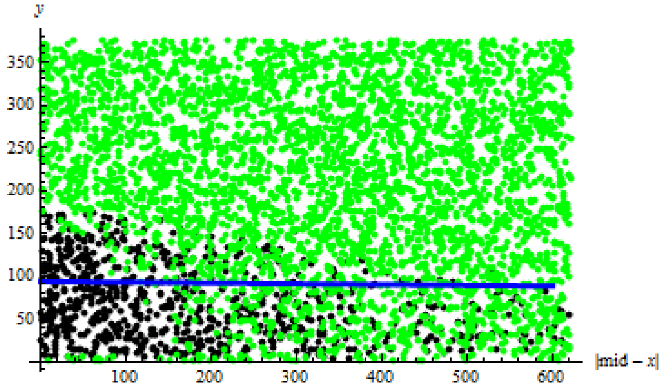
Up to now all SVMs where trained for data sets where number of road pixels = number of environment pixels

Hence, the F_1-score for the training data does not reflect the real F_1 score

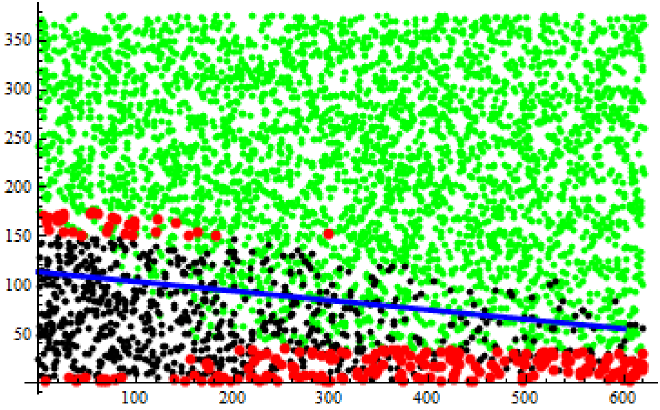
How can we take into account the fact that environment pixels are much more frequent?

Distribution of road vs environment pixels

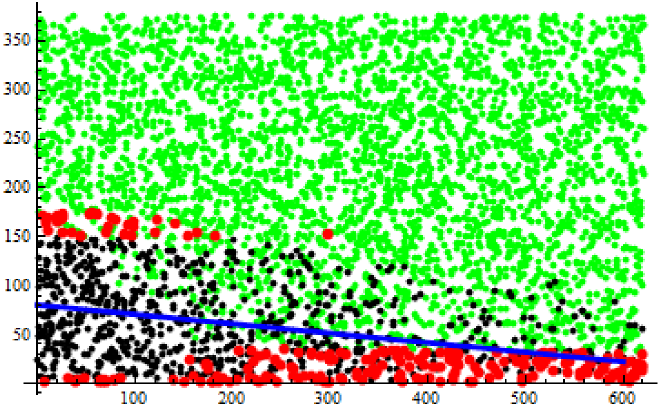
A simple solution: distribute the pixels of the training set uniformly over the entire image



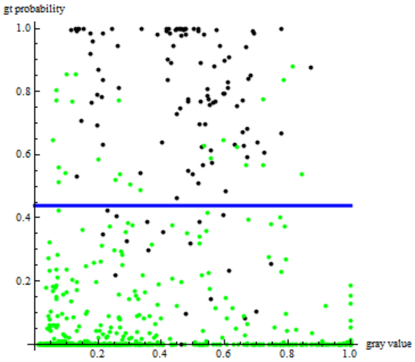
Removal of outliers



Optimization of the f_1-score



Another example



- Feature = (gray value, ground truth probability)
- ground truth probability is dominant, gray value is ignored
- F_1 score = 0.81 on training data
- F_1 score = 0.86 on validation data (!)

Limitations of hard linear discriminant functions

Limitations

- hyperplane often too simple to separate data
- some support vectors may be unreliable, in particular when the data sets cannot be separated

How to cope with these limitations?

- introduction of kernels to obtain more powerful decision surfaces
- use of soft margins

Table of Contents

Mathematical formulation of maximal margin problem

Quadratic programming problem with linear constraints

Minimize

$$\frac{1}{2}||\mathbf{w}||^2$$

such that $y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$

is equivalent to

Quadratic programming problem with linear constraints

Minimize

$$L_p(\mathbf{w}, b, \alpha_i) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{n} \alpha_i(y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1)$$

such that $\alpha_i \geq 0$

Mathematical formulation of maximal margin problem

Quadratic programming problem with linear constraints

Minimize

$$L_p(\mathbf{w}, b, \alpha_i) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{n} \alpha_i(y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1)$$

such that $\alpha_i \geq 0$

$L_p(\mathbf{w}, b, \alpha_i)$ is called a Lagrangian function
α_i are called Lagrange multipliers

Mathematical formulation of maximal margin problem

Quadratic programming problem with linear constraints

Minimize

$$L_p(\mathbf{w}, b, \alpha_i) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{n} \alpha_i(y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1)$$

such that $\alpha_i \geq 0$

Minus sign in front of Lagrange multipliers since we

- minimize with respect to **w** and b
- maximize with respect to α_i
- often solved in dual space

Lagrangian dual problem

Quadratic programming problem with linear constraints

Minimize

$$L_p(\mathbf{w}, b, \alpha_i) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{n} \alpha_i (y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1)$$

such that $\alpha_i \geq 0$

$$\frac{\partial L_p}{\partial \mathbf{w}} = 0 \quad \rightarrow \quad \mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$$

$$\frac{\partial L_p}{\partial b} = 0 \quad \rightarrow \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

Lagrangian dual problem

Since

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

we have

$$\begin{array}{lcl} \frac{1}{2}||\mathbf{w}||^2 & = & \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \\ \sum_{i=1}^{n} \alpha_i y_i b & = & 0 \\ \sum_{i=1}^{n} \alpha_i y_i \mathbf{w}^T \mathbf{x}_i & = & \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \end{array}$$

Lagrangian dual problem

Since

$$\begin{aligned}
\frac{1}{2}||\mathbf{w}||^2 &= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j \\
\sum_{i=1}^{n}\alpha_i y_i b &= 0 \\
\sum_{i=1}^{n}\alpha_i y_i \mathbf{w}^T\mathbf{x}_i &= \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j
\end{aligned}$$

the Lagrangian

$$L_p(\mathbf{w}, b, \alpha_i) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{n}\alpha_i(y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1)$$

can be rewritten as

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

Lagrangian dual problem

Maximal margin problem

Minimize

$$L_p(\mathbf{w}, b, \alpha_i) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{n} \alpha_i (y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1)$$

such that $\alpha_i \geq 0$

is equivalent to

Lagrangian dual problem

Maximize

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

such that $\alpha_i \geq 0$ and $\sum_{i=1}^{n} \alpha_i y_i = 0$

Lagrangian dual problem

Lagrangian dual problem

Maximize

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

such that $\alpha_i \geq 0$ and $\sum_{i=1}^{n} \alpha_i y_i = 0$

Why is dual problem interesting?

- input data always appear in the form $\mathbf{x}_i^T \mathbf{x}_j$.
- $\mathbf{x}_i^T \mathbf{x}_j$ can be replaced by $K(\mathbf{x}_i, \mathbf{x}_j)$
- $K(\mathbf{x}_i, \mathbf{x}_j)$ is called a kernel function

Lagrangian dual problem

Lagrangian dual problem

Maximize

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

such that $\alpha_i \geq 0$ and $\sum_{i=1}^{n} \alpha_i y_i = 0$

Valid kernel functions

- Linear kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$
- Polynomial kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^T \mathbf{x}_j)^p$
- Radial basis function: $K(\mathbf{x}_i, \mathbf{x}_j) = exp(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma^2})$

A kernel function has to satisfy Mercer's condition

Kernels in dual space vs lifting in primal space

Example: Let $\mathbf{x} = (x_1, x_2)^T$, and let $K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^T \mathbf{x}_j)^2$.

Then

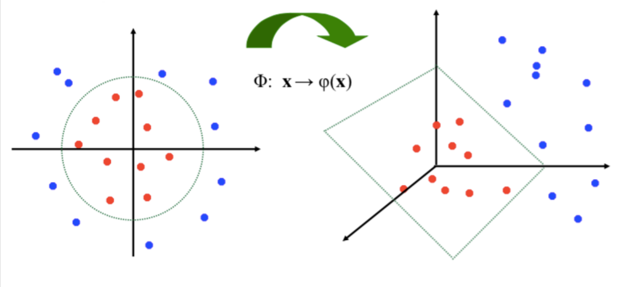
$$
\begin{aligned}
&K(\mathbf{x}_i, \mathbf{x}_j) \\
&= (1 + \mathbf{x}_i^T \mathbf{x}_j)^2 \\
&= (1 + x_{i1} x_{j1} + x_{i2} x_{j2})^2 \\
&= 1 + x_{i1}^2 x_{j1}^2 + x_{i1} x_{j1} x_{i2} x_{j2} + x_{i2}^2 x_{j2}^2 + 2x_{i1} x_{j2} + 2x_{i2} x_{j2} \\
&= (1, x_{i1}^2, \sqrt{2} x_{i1} x_{i2}, x_{i2}^2, \sqrt{2} x_{i1}, \sqrt{2} x_{i2}) \cdot (1, x_{j1}^2, \sqrt{2} x_{j1} x_{j2}, x_{j2}^2, \sqrt{2} x_{j1}, \sqrt{2} x_{j2}) \\
&= \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_i)
\end{aligned}
$$

Using $(1 + \mathbf{x}_i^T \mathbf{x}_j)^2$ in the dual problem, is equivalent to a linear separation of *lifted vectors* in a certain primal space

$$(x_1, x_2) \to (1, x_{j1}^2, \sqrt{2} x_{j1} x_{j2}, x_{j2}^2, \sqrt{2} x_{j1}, \sqrt{2} x_{j2})$$

But we do not have to compute the lifted vectors, only $(1 + \mathbf{x}_i^T \mathbf{x}_j)^2$

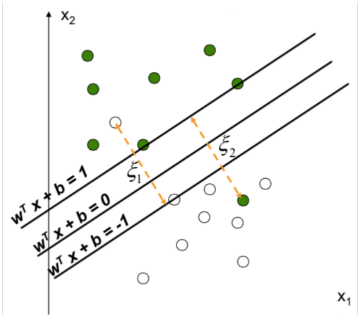
Kernels vs lifting



- Use of polynomial kernel in dual space = lifting in primal space
- $(x_1, x_2) \to (1, x_{j1}^2, \sqrt{2}x_{j1}x_{j2}, x_{j2}^2, \sqrt{2}x_{j1}, \sqrt{2}x_{j2}) = (z_0, z_1, z_2, z_3, z_4, z_5)$
- Separating hyperplane in $(z_0, z_1, z_2, z_3, z_4, z_5)$ space corresponds to separating conic in the (x_1, x_2) plane.
- In dual space this comes with a very small computational cost.

Soft margins

- Data that is not linearly separable
- Introduce slack variables ξ_i to allow misclassification



Mathematical formulation soft margin

Soft margins in primal space

Minimize

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

such that

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

- C is parameter used to control over-fitting
- $C \to \infty$ corresponds to hard margins

Lagrangian dual problem

Soft margins in the dual space

Maximize

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

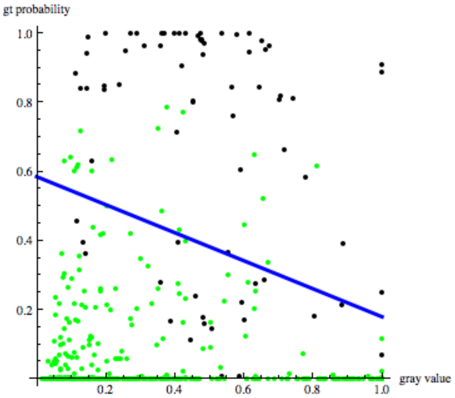
such that $0 \leq \alpha_i \leq C$ and $\sum_{i=1}^{n} \alpha_i y_i = 0$

- only constraints $\alpha_i \leq C$ have been added

Table of Contents

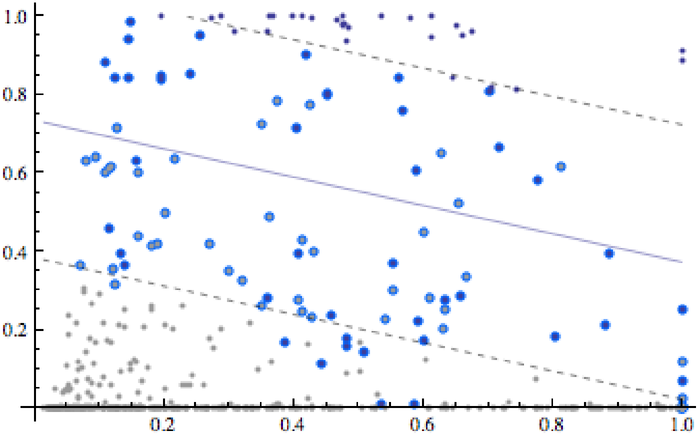
Training set

- Pixel features = (gray value, road probability)
- training set = 20 random pixels taken from 25 images



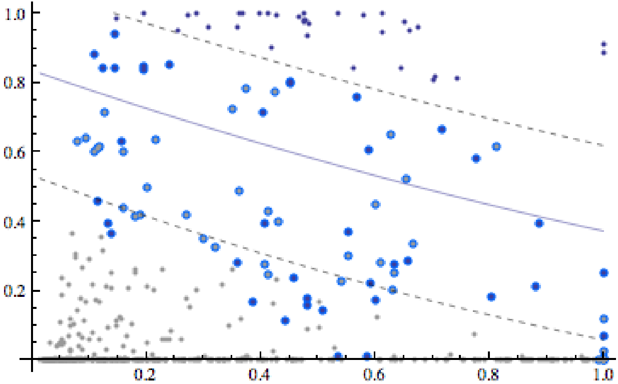
Linear separation with minimal distance criterion

Linear separation with soft margins



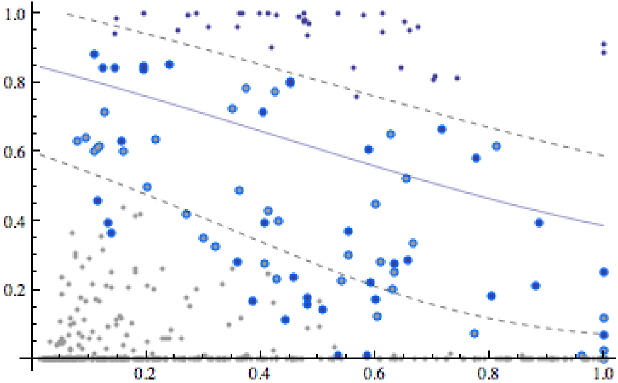
Identity kernel $\mathbf{x} \cdot \mathbf{y}$, $C = 0.80$, smaller C = softer boundaries

Polynomial separation with soft margins



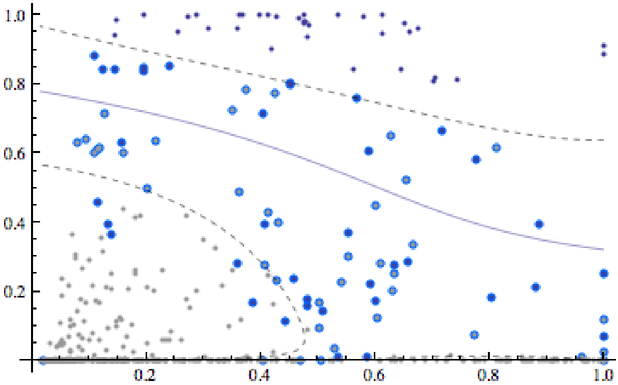
Polynomial kernel $(1 + \mathbf{x} \cdot \mathbf{y})^2$, $C = 0.80$
Discriminant function:
$-0.0971x_1^2 + 1.343x_1x_2 + 1.0138x_1 + 1.548x_2^2 + 1.178x_2 - 2.063$

Polynomial separation with soft margins



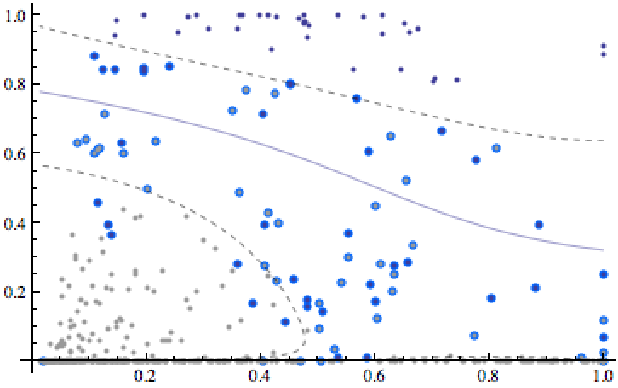
Polynomial kernel $(1 + \mathbf{x} \cdot \mathbf{y})^3$, $C = 0.80$
Discriminant function:
$-1.64 + 0.72x_1 - 0.013x_1^2 - 0.21x_1^3 - 0.397x_2 + 1.316x_1x_2 + 1.033x_1^2x2 + 1.79x_2^2 + 0.48x_1x2^2 + 1.11x_2^3$

RBF with soft margins



- Kernel is Radial Basis Function, $C = 1.5$, $\sigma = 0.25$
- Support vectors lie on two surfaces
- First surface corresponds to large recall, second surface to large precision.

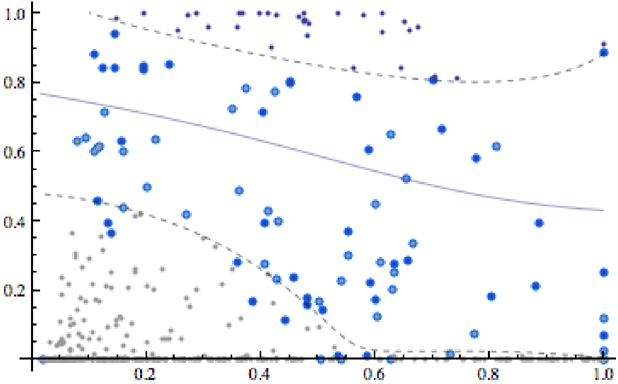
RBF with soft margins



Discriminant function:

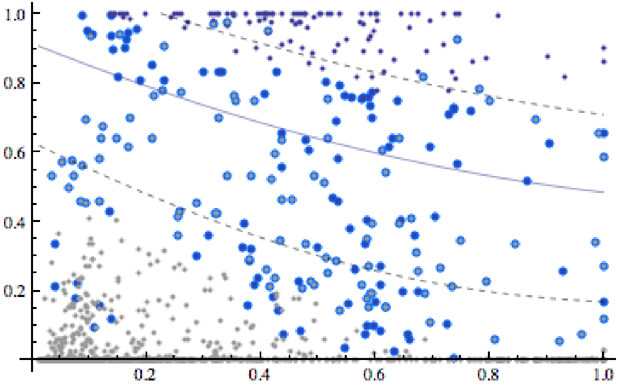
$0.0564 + 1.5e^{-2((-0.111+x_1)^2+(-0.8815+x_2)^2)} +$
$1.5e^{(-2((-0.2405+x1)^2+(-0.850+x2)^2))} + \cdots$

RBF with soft margins



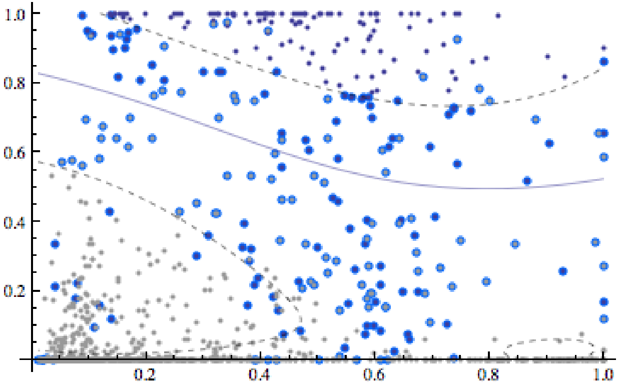
- Kernel is Radial Basis Function, $C = 0.7$, $\sigma = 0.25$
- When C becomes smaller, margins become softer

Larger training set



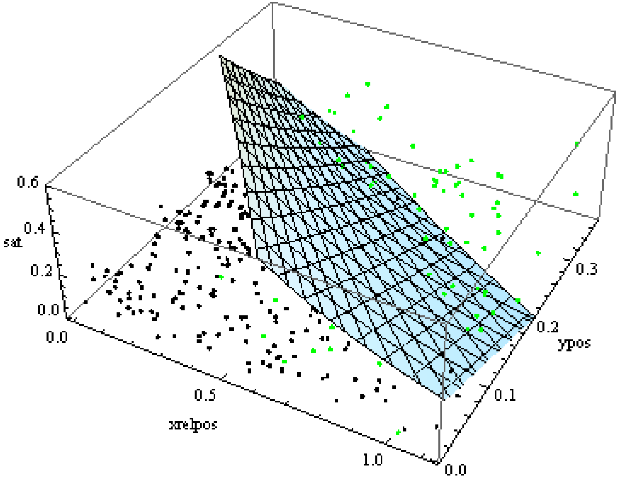
- Larger training set (50 pixels, from 25 images)
- Polynomial kernel $(1 + \mathbf{x} \cdot \mathbf{y})^2$, $C = 0.80$

Radial Basis Function with soft margins



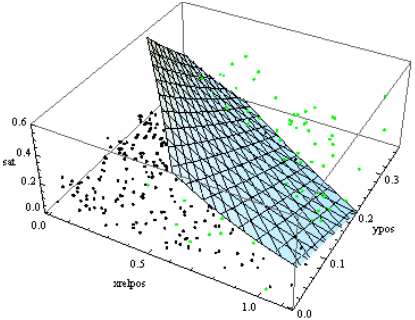
- Larger training set (50 pixels, from 25 images)
- RBF kernel, $C = 0.70$, $\sigma = 0.25$

Example with a 3D feature space



- feature vector = $(|x - m|, y, saturation)$
- polynomial kernel $(1 + \mathbf{x} \cdot \mathbf{y})^3$, $C = 1.0$

Polynomial separation with soft margins



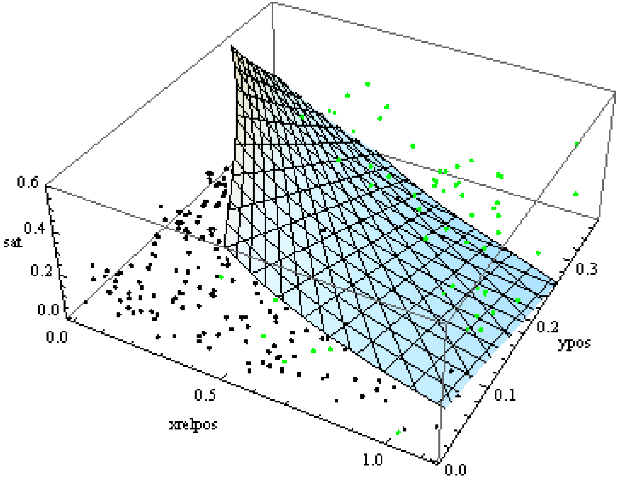
Discriminant function:

$$g(x_1, x_2, x_3) = 3.255 - 0.9306x_1 - 0.2237x_1^2 + 0.04301x_1^3 - 5.230x_2 - 2.472x_1x_2 - 0.3586x_1^2x_2 - 4.093x_2^2 - 1.009x_1x_2^2 - 0.7133x_2^3 - 2.254x_3 - 3.120x_1x_3 - 1.120x_1^2x_3 - 2.537x_2x_3 - 1.213x_1x_2x_3 - 0.5974x_2^2x_3 + 0.3817x_3^2 + 0.05558x_1x_3^2 - 0.06647x_2x_3^2 + 0.1124x_3^3$$

What does it mean for images?

Classification with $g(x_1, x_2, x_3)$
as discriminant function for $(|x - m|, y, \mathit{saturation})$

3D feature space



- feature vector = $(|x - m|, y, saturation)$
- RBF kernel, $C = 1.0$, $\sigma = 0.25$

LibSVM

- SVM in OpenCV is based on LibSVM
- LibSVM uses Platt's Sequential Minimization (SMO) algorithm
- In this presentation we used the Keerthi-Gilbert algorithm

How to choose the best SVM?

- first normalize data so that ranges are similar
- first try linear SVM, then RBF, then polynomial
- grid based method tries many parameter settings, e.g., for RBF we experiment with $\sigma \in [2^{-20}, 2^{-19}, \ldots, 2^{20}]$ $(\sigma = 1/\gamma)$, and $C \in [2^{-7}, 2^{-6}, \ldots, 2^{7}]$.

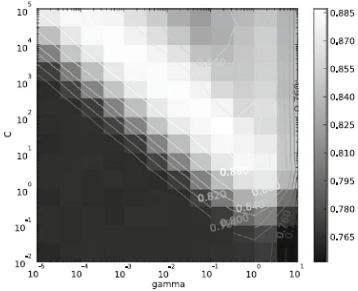


Fig. 13.6. SVM accuracy on a grid of parameter values.