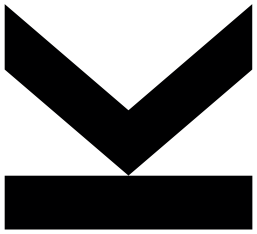
UNIT 3



Support Vector Machines

JOHANNES KEPLER
UNIVERSITY LINZ

INTRODUCTION

- Putting it simply, *Support Vector Machines (SVMs)* are based on the idea of finding a linear classification border that maximizes the margin between positive and negative samples.
- It will turn out that margin maximization is related to simultaneous minimization of model complexity.

OVERVIEW/AGENDA

1. We will start with the linear case and consider margin maximization, its computational formulation, and issues related to complexity in depth.
2. Then the generalization to the non-linear case is rather straightforward.
3. Then we can highlight different variants including support vector regression.

LINEAR SEPARABILITY: DEFINITION

Assume we are given a data set $\mathbf{Z}$ consisting of labeled samples $(\mathbf{x}^i, y^i)_{i=1,\ldots,l}$, where $\mathbf{x}^i \in X = \mathbb{R}^d$ and $y^i \in \{-1, 1\}$ and further assume that positive and negative samples are *linearly separable*, i.e. there exist a vector $\mathbf{w} \in \mathbb{R}^d$ and a constant $b \in \mathbb{R}$ such that, for all $i = 1, \ldots, l$,

$$\text{sign}(\mathbf{w} \cdot \mathbf{x}^i + b) = y^i.$$

Obviously, the hyperplane separating positive and negative samples is given as $\mathbf{w} \cdot \mathbf{x} + b = 0$.

Lemma. Two sets of points are linearly separable if and only if their convex hulls are disjoint.

LINEAR SEPARABILITY: CANONICAL FORM

Lemma. Given a linearly separable data set (in the sense of above) and a separating hyperplane $\mathbf{w} \cdot \mathbf{x} + b = 0$, there exists another separating hyperplane $\mathbf{w}' \cdot \mathbf{x} + b' = 0$ such that

$$\min_{i=1,\ldots,l} |\mathbf{w}' \cdot \mathbf{x}^i + b'| = 1.$$

Definition. If a separating hyperplane $\mathbf{w} \cdot \mathbf{x} + b = 0$ already fulfills

$$\min_{i=1,\ldots,l} |\mathbf{w} \cdot \mathbf{x}^i + b| = 1,$$

we say that $\mathbf{w} \cdot \mathbf{x} + b = 0$ is in *canonical form* (with respect to $\mathbf{Z}$).

LINEAR SEPARABILITY: CANONICAL FORM (cont'd)

Lemma. A separating hyperplane in canonical form fulfills the following set of inequalities:

$$\mathbf{w} \cdot \mathbf{x}^i + b \geq +1 \qquad \text{for } y^i = +1$$

$$\mathbf{w} \cdot \mathbf{x}^i + b \leq -1 \qquad \text{for } y^i = -1$$

These inequalities are equivalent to the following set of inequalities (for all $i = 1, \ldots, l$):

$$y^i(\mathbf{w} \cdot \mathbf{x}^i + b) - 1 \geq 0$$

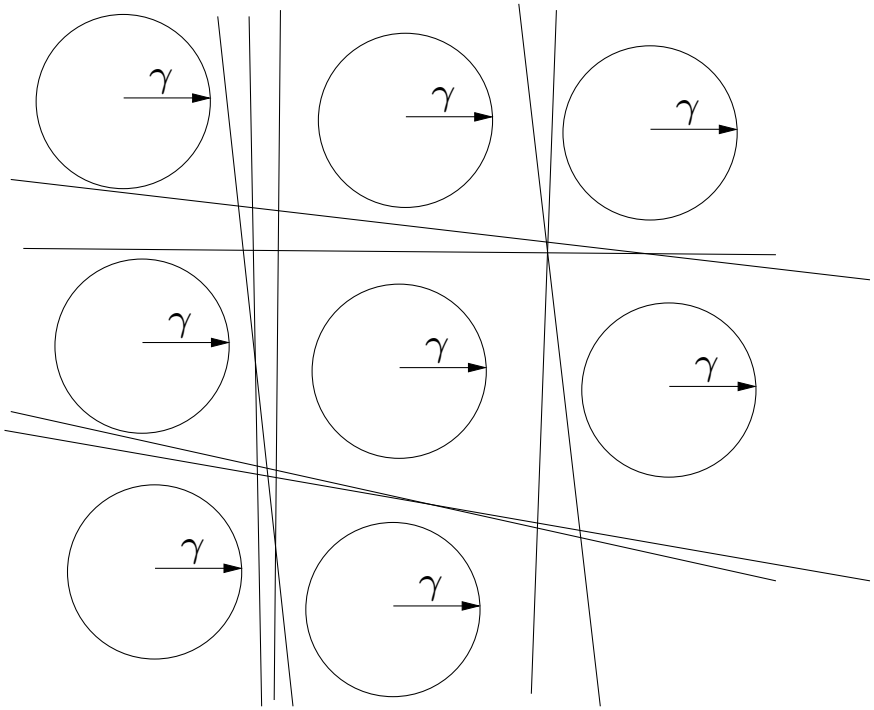
SEPARATING HYPERPLANES WITH BOUNDED MINIMAL DISTANCE

It follows easily (cf. Hesse normal form) that the closest distance of a point $\mathbf{x}$ to the separating hyperplane is

$$\frac{|\mathbf{w} \cdot \mathbf{x} + b|}{\|\mathbf{w}\|}.$$

Hence, if $\mathbf{w} \cdot \mathbf{x} + b$ is in canonical form, the distance of the separating hyperplane to the closest data point is $\frac{1}{\|\mathbf{w}\|}$. So if we want to restrict to those separating hyperplanes (in canonical form) that have a distance of at least γ to all data points, we have to introduce the constraint $\|\mathbf{w}\| \leq \frac{1}{\gamma}$.

SEPARATING HYPERPLANES WITH BOUNDED MINIMAL DIST. (cont'd)



Rationale: the farther a separating hyperplane is away from the data, the less likely it is to produce a misclassification.

SHATTERING COEFFICIENT

Let us assume from here on, that we are dealing with binary classification, i.e. $g(.;.) \in \{-1, +1\}$. For convenience, we will sometimes identify the model class $g(.;.)$ with the set of functions it contains, i.e. $g = \{g(.;\mathbf{w}) \mid \mathbf{w}\}$.

Definition. Given a model class $g(.;.)$ and a family of l sample inputs $(\mathbf{x}^1, \ldots, \mathbf{x}^l)$, the *shattering coefficient* of g for $(\mathbf{x}^1, \ldots, \mathbf{x}^l)$ is defined as

$$\mathcal{N}_g(\mathbf{x}^1, \ldots, \mathbf{x}^l) = \left|\left\{(g(\mathbf{x}^1;\mathbf{w}), \ldots, g(\mathbf{x}^l;\mathbf{w})) \mid \mathbf{w}\right\}\right|,$$

i.e. the number of possible labelings of $\{\mathbf{x}^1, \ldots, \mathbf{x}^l\}$ that the model class $g(.;.)$ is able to realize (for any parameter setting $\mathbf{w}$).

SHATTERING COEFFICIENT (cont'd)

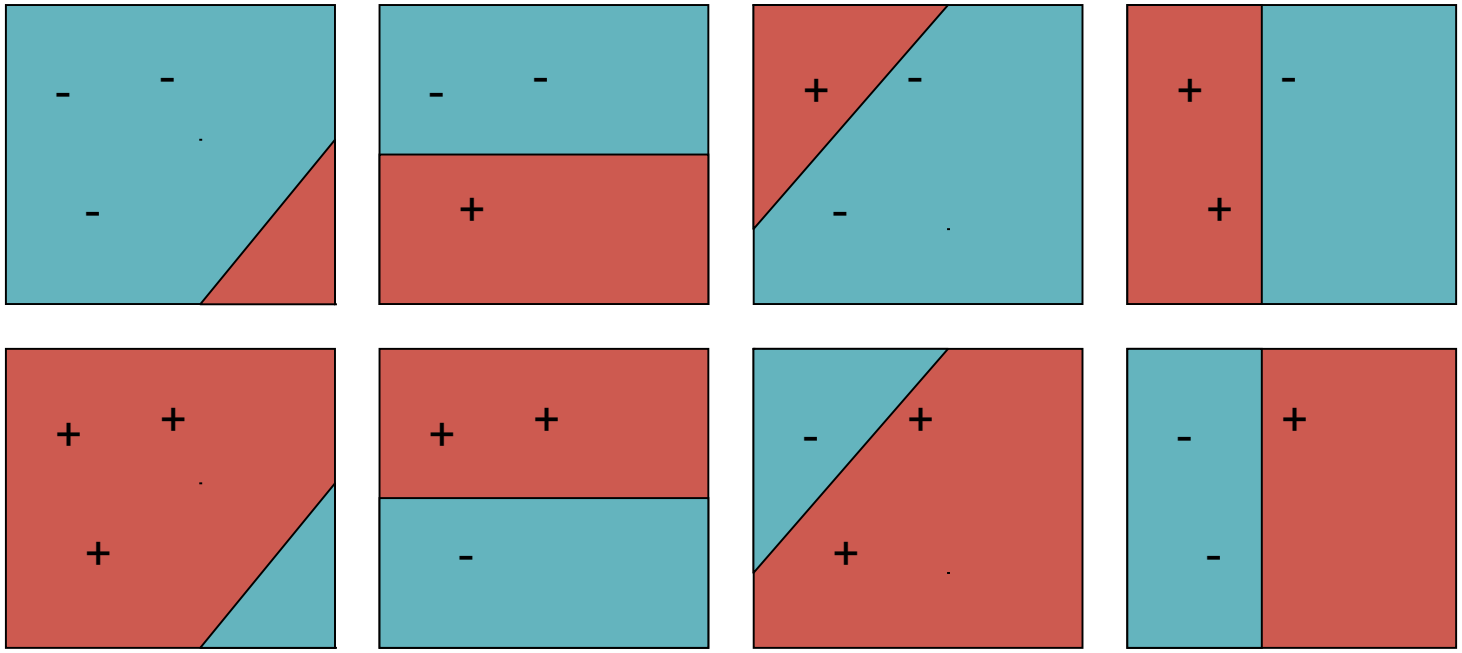
Obviously, if $\mathcal{N}_g(\mathbf{x}^1, \ldots, \mathbf{x}^l) = 2^l$, $g(.;.)$ can model any labeling of the inputs $\{\mathbf{x}^1, \ldots, \mathbf{x}^l\}$. In this case, we say that $g(.;.)$ *shatters* $\{\mathbf{x}^1, \ldots, \mathbf{x}^l\}$.

Example: Consider $X = \mathbb{R}^2$ and

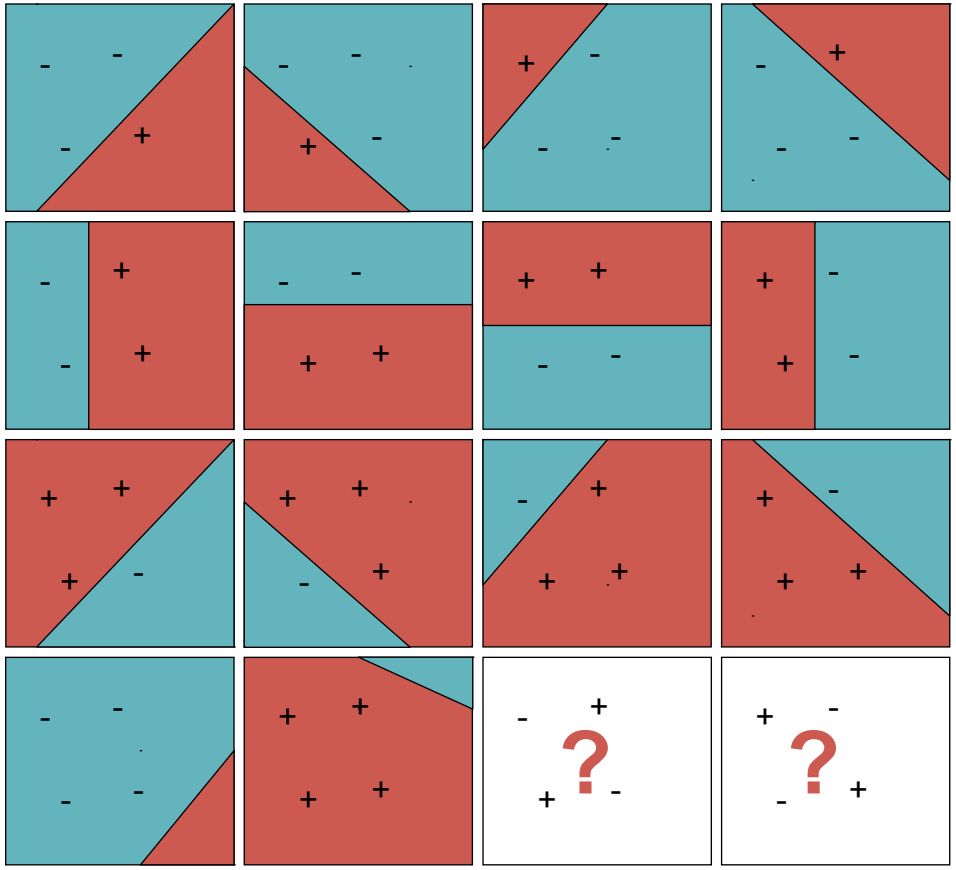
$$g_{\text{lin}}((x_1, x_2); (w_1, w_2, b)) = \begin{cases} 1 & \text{if } w_1 x_1 + w_2 x_2 \geq b, \\ -1 & \text{otherwise,} \end{cases}$$

i.e. linear separation. Then, for any three points $\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3$ that are not collinear, we have $\mathcal{N}_{g_{\text{lin}}}(\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3) = 8$. For four points $\mathbf{x}^1, \ldots, \mathbf{x}^4$ arranged as a general tetragon, we obtain $\mathcal{N}_{g_{\text{lin}}}(\mathbf{x}^1, \ldots, \mathbf{x}^4) = 14$.

SHATTERING COEFFICIENT
EXAMPLE #1



SHATTERING COEFFICIENT
EXAMPLE #2



THE VAPNIK-CHERVONENKIS (VC) DIMENSION

Definition. The *Vapnik-Chervonenkis dimension (VC dimension)* of a model class $g(.;.)$ is defined as

$$d_{\text{vc}}(g) = \sup\{l \in \mathbb{N} \mid \exists(\mathbf{x}^1, \ldots, \mathbf{x}^l)\, \mathcal{N}_g(\mathbf{x}^1, \ldots, \mathbf{x}^l) = 2^l\},$$

i.e. the VC dimension is the largest number l for which a configuration of l samples can be found that can be shattered by a model from the model class g. If this works for all l, the VC dimension is ∞.

VC DIMENSION: EXAMPLES

- For $X = \mathbb{R}^2$, we have $d_{\text{VC}}(g_{\text{lin}}) = 3$.
- For $X = \mathbb{R}^d$, we have $d_{\text{VC}}(g_{\text{lin}}) = d + 1$ (where g_{lin} is generalized to the p-dimensional case in the obvious way).
- For $X = \mathbb{R}$ and any model class that contains only non-decreasing functions, we have $d_{\text{VC}}(g) = 1$, regardless of how many parameters are necessary to parametrize g.
- For $X = \mathbb{R}$ and $g_{\sin}(x, w) = \text{sign}(\sin(wx))$, we obtain $d_{\text{VC}}(g_{\sin}) = \infty$, although $g_{\sin}$ only depends on one parameter.

We conclude that there is not necessarily a dependency between the VC dimension and the number of parameters which describe a model class.

BOUNDED MINIMAL DISTANCE: A BOUND ON COMPLEXITY

Theorem. Consider input data from a sphere with radius R. The maximal number of points that linear hyperplanes can shatter without getting closer to any data point than γ is bounded above by

$$\min\left(\left\lfloor \frac{R^2}{\gamma^2} \right\rfloor, d\right) + 1.$$

BOUNDED MINIMAL DISTANCE: A BOUND ON THE ERROR RATE

Theorem. Consider a training set with l elements from a sphere with radius R again (drawn according to some distribution) and a linear separating hyperplane that has a distance of at least γ from each training sample. For a given $\rho > 0$, we define ν as the proportion of samples for which

$$y(\mathbf{w} \cdot \mathbf{x} + b) \leq \rho$$

holds (i.e. margin error of at least $\frac{\rho}{\|\mathbf{w}\|}$). Then, with probability $1 - \delta$, the probability to misclassify a new sample is bounded above by

$$\nu + \sqrt{\frac{c}{l}\left(\frac{R^2}{\rho^2\gamma^2}(\ln l)^2 + \ln(1/\delta)\right)},$$

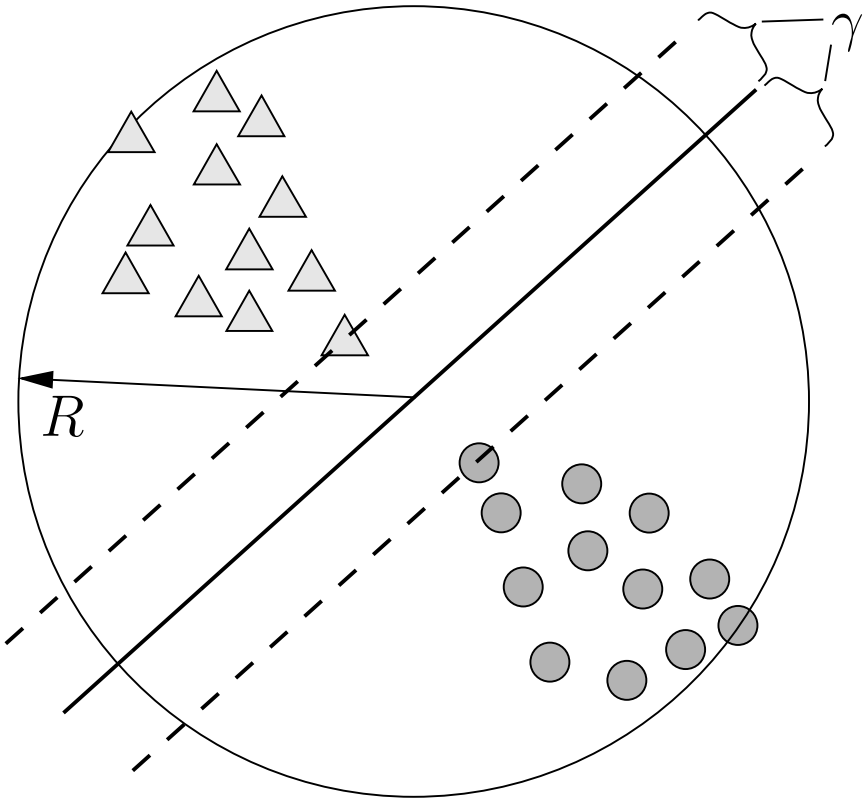
where $c > 0$ is a constant.

THE BIGGER γ, THE BETTER

The previous two theorems indicate that we should look for the largest γ such that there still exists a separating hyperplane that has a distance of at least γ to all training samples (bounding complexity, minimizing the test error à la SRM).

Hence, we are looking for that separating hyperplane whose minimal distance to all training samples is maximal. Assuming that the separating hyperplane is in canonical form, this is equivalent to maximizing the distance $\frac{1}{\|\mathbf{w}\|}$.

THE BIGGER γ, THE BETTER (cont'd)

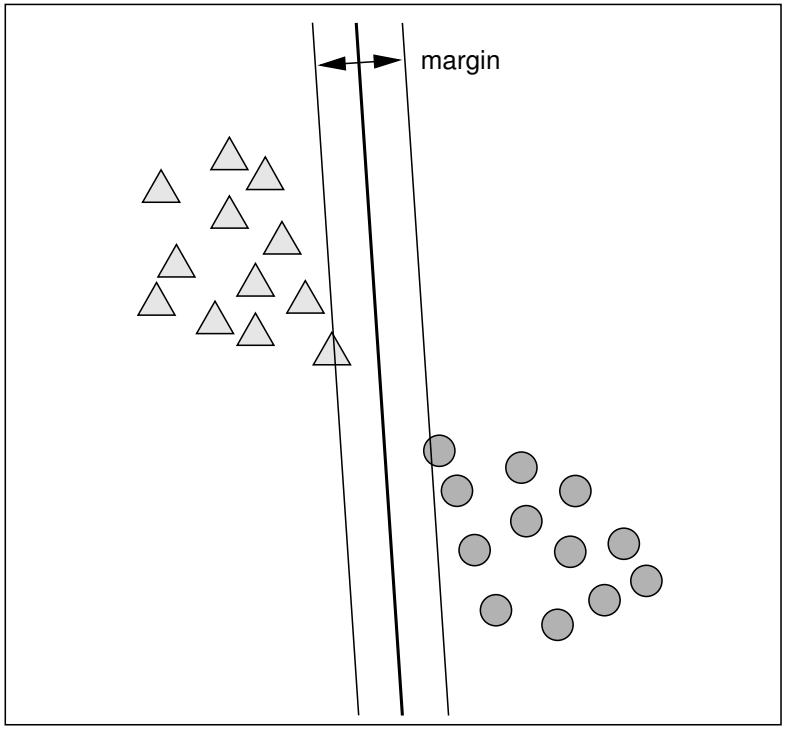


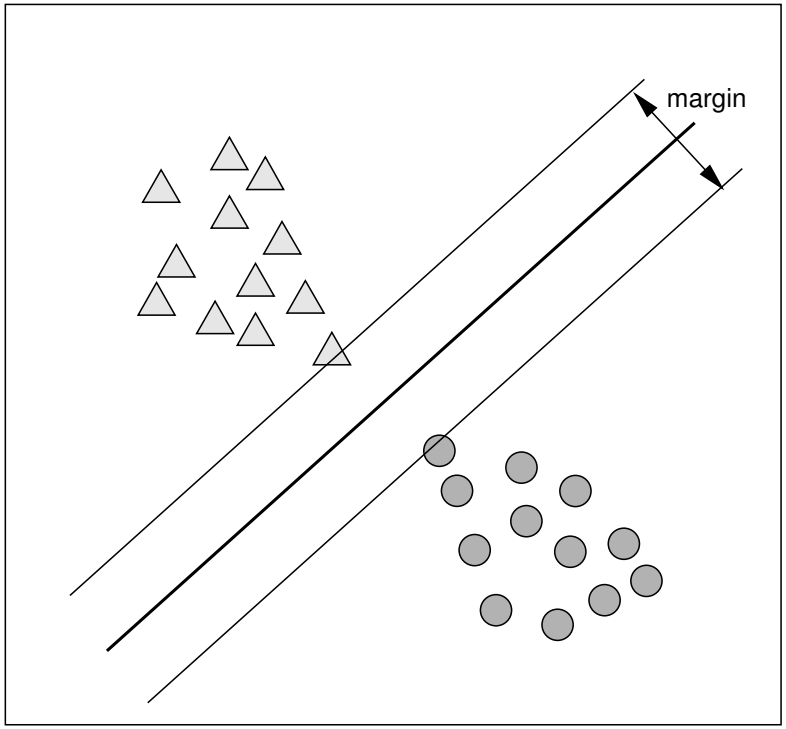
MARGIN MAXIMIZATION

Obviously, for such an optimal hyperplane, the smallest distance to the closest negative sample d_- and the smallest distance to the closest positive sample d_+ are the same, and the distance of positive and negative samples perpendicular to $\mathbf{w}$ is $d_+ + d_- = \frac{2}{\|\mathbf{w}\|}$. This distance is commonly called *margin*. Hence, maximizing the minimal distance to all data points (by maximizing $\frac{1}{\|\mathbf{w}\|}$) is nothing else but margin maximization.

Lemma. The separating hyperplane that maximizes the margin between positive and negative samples is uniquely given as the hyperplane that orthogonally bisects the shortest distance between the convex hulls of positive and negative samples.

MARGIN MAXIMIZATION (cont'd)





MARGIN MAXIMIZATION: OPTIMIZATION PROBLEM

Original Problem: For a given linearly separable data set $\mathbf{Z}$, maximize $\frac{2}{\|\mathbf{w}\|}$ with respect to $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$ subject to the following constraints ($i = 1, \ldots, l$):

$$y^i(\mathbf{w} \cdot \mathbf{x}^i + b) - 1 \geq 0 \tag{1}$$

This is equivalent to the following optimization problem:

Primal Problem: For a given linearly separable data set $\mathbf{Z}$, minimize $\frac{1}{2}\|\mathbf{w}\|^2 = \frac{1}{2}\sum_{i=1}^{d} w_i^2$ with respect to $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$ subject to the constraints (1).

Obviously, the latter is a convex quadratic optimization problem with linear constraints.

EXCURSION: CONVEX OPTIMIZATION (1/3)

Suppose that the functions f and g_i ($i = 1, \ldots, n$) are all *convex*. A function h is convex if $h(\lambda \mathbf{x} + (1 - \lambda)\mathbf{y}) \leq \lambda h(\mathbf{x}) + (1 - \lambda)h(\mathbf{y})$ for all $\mathbf{x}, \mathbf{y}$ and all $\lambda \in [0, 1]$. For convenience, assume that f and all g_i are continuously differentiable. Further assume that the *Slater condition* holds, i.e. there exists an $\mathbf{x}'$ such that $g_i(\mathbf{x}') < 0$ for all $i = 1, \ldots, n$.

Primal Problem: minimize $f(\mathbf{x})$ with respect to $\mathbf{x}$ subject to the constraints $g_i(\mathbf{x}) \leq 0$, where $i = 1, \ldots, n$ (note that, for simplicity, we do not deal with equality constraints here).

EXCURSION: CONVEX OPTIMIZATION (2/3)

Lagrange function:

$$L(\mathbf{x}; \alpha_1, \ldots, \alpha_n) = f(\mathbf{x}) + \sum_{i=1}^{n} \alpha_i g_i(\mathbf{x})$$

The auxiliary variables $\alpha_1, \ldots, \alpha_l$ are called *Lagrange multipliers*.

Dual Problem: maximize

$$\mathcal{L}(\alpha_1, \ldots, \alpha_n) = \inf_{\mathbf{x}} L(\mathbf{x}; \alpha_1, \ldots, \alpha_n)$$

subject to the constraints $\alpha_i \geq 0$ $(i = 1, \ldots, n)$.

EXCURSION: CONVEX OPTIMIZATION (3/3)

From the *Karush-Kuhn-Tucker Theorem*, we can infer the following: under the assumptions made above, for a solution $\mathbf{x}^*$ of the primal problem, there exist non-negative Lagrange multipliers such that

$$\mathcal{L}(\alpha_1, \ldots, \alpha_n) = L(\mathbf{x}^*; \alpha_1, \ldots, \alpha_n)$$

and such that $\alpha_i g_i(\mathbf{x}^*) = 0$ holds for all $i = 1, \ldots, n$. These conditions are not only necessary, but also sufficient for $\mathbf{x}^*$ to be a solution of the primal problem.

LAGRANGE FUNCTION OF MARGIN MAXIMIZATION

We introduce l Lagrange multipliers $\alpha_1, \ldots, \alpha_l$. Then the Lagrange function is given as

$$L(\mathbf{w}, b; \alpha_1, \ldots, \alpha_l) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{l} \alpha_i \big(y^i(\mathbf{w} \cdot \mathbf{x}^i + b) - 1\big)$$

$$= \frac{1}{2}\|\mathbf{w}\|^2 - \mathbf{w} \cdot \sum_{i=1}^{l} \alpha_i y^i \mathbf{x}^i - b \sum_{i=1}^{l} \alpha_i y^i + \sum_{i=1}^{l} \alpha_i$$

MARGIN MAXIMIZATION: DUAL FORMULATION (1/4)

Solving the dual problem includes minimizing L with respect to $\mathbf{w}$ and b (for fixed Lagrange multipliers). This enforces the conditions

$$\frac{\partial L}{\partial \mathbf{w}}(\mathbf{w}, b; \alpha_1, \ldots, \alpha_l) = 0 \qquad \frac{\partial L}{\partial b}(\mathbf{w}, b; \alpha_1, \ldots, \alpha_l) = 0,$$

which imply the following:

$$\mathbf{w} = \sum_{i=1}^{l} \alpha_i y^i \mathbf{x}^i \qquad \sum_{i=1}^{l} \alpha_i y^i = 0$$

MARGIN MAXIMIZATION: DUAL FORMULATION (2/4)

By using the previous two equalities, we obtain

$$\mathcal{L}(\alpha_1, \ldots, \alpha_l) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y^i y^j \mathbf{x}^i \cdot \mathbf{x}^j.$$

The final solution can be found by maximizing $\mathcal{L}$ with respect to the Lagrange multipliers α_i subject to the constraints $\alpha_i \geq 0$ (for all $i = 1, \ldots, l$) and $\sum_{i=1}^{l} \alpha_i y^i = 0$.

MARGIN MAXIMIZATION: DUAL FORMULATION (3/4)

With the notations

$$\mathbf{0} = (\overbrace{0, \ldots, 0}^{l \text{ times}})^T, \qquad \boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_l)^T,$$

$$\mathbf{1} = (\underbrace{1, \ldots, 1}_{l \text{ times}})^T, \qquad \mathbf{Q} = \left(y^i y^j \mathbf{x}^i \cdot \mathbf{x}^j\right)_{i=1,\ldots,l}^{j=1,\ldots,l},$$

we can write the dual problem as follows:

Minimize

$$\frac{1}{2}\boldsymbol{\alpha}^T \mathbf{Q} \boldsymbol{\alpha} - \mathbf{1}^T \boldsymbol{\alpha}$$

with respect to $\boldsymbol{\alpha}$ subject to the constraints $\boldsymbol{\alpha} \geq \mathbf{0}$ and $\boldsymbol{\alpha}^T \mathbf{y} = 0$.

MARGIN MAXIMIZATION: DUAL FORMULATION (4/4)

Note that $\mathbf{K} = (\mathbf{x}^i \cdot \mathbf{x}^j)_{i=1,\ldots,l}^{j=1,\ldots,l}$ is positive semi-definite, since $\mathbf{K} = \mathbf{X}\mathbf{X}^T$ holds.[a] From this fact, we can easily infer that $\mathbf{Q}$ is positive semi-definite.

Hence, not surprisingly, the dual problem, like the equivalent ones described above, is a *convex quadratic optimization problem with linear constraints*. For such problems, no local minima exist. The set of global minima (consisting of equally good solutions) is convex. If $\mathbf{Q}$ is positive definite, the minimum is even unique. For such problems, a host of solving algorithms are available.

[a] It is easy to prove that Gram matrices of scalar products are in general positive semi-definite.

SUPPORT VECTORS

Once we have solved the dual optimization problem, we have Lagrange multipliers $\alpha_1, \ldots, \alpha_l$ which, by the Karush-Kuhn-Tucker theorem, also solve the primal problem. By the Karush-Kuhn-Tucker conditions, we have

$$\alpha_i \big(y^i (\mathbf{w} \cdot \mathbf{x}^i + b) - 1 \big) = 0 \tag{2}$$

for all $i = 1, \ldots, l$. This means, for all $i = 1, \ldots, l$, we either have $\alpha_i = 0$ or $y^i(\mathbf{w} \cdot \mathbf{x}^i + b) - 1 = 0$ (or both). Samples for which $\alpha_i > 0$ holds (thus implying $y^i(\mathbf{w} \cdot \mathbf{x}^i + b) - 1 = 0$) are called *support vectors*. It is intuitively clear anyway, that the maximal margin only depends on those samples for which the constraints are tight.

CONSTRUCTING THE FINAL CLASSIFIER

Given Lagrange multipliers $\alpha_1, \ldots, \alpha_l$ solving the primal problem, we can construct $\mathbf{w}$ as noted above already:

$$\mathbf{w} = \sum_{i=1}^{l} \alpha_i y^i \mathbf{x}^i$$

Hence, the final classification function (the *linear Support Vector Machine (SVM)*) is given as

$$g(\mathbf{x}) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b) = \text{sign}\Big(\underbrace{\sum_{i=1}^{l} \alpha_i y^i \mathbf{x}^i \cdot \mathbf{x} + b}_{\text{discriminant function } \bar{g}(\mathbf{x})} \Big).$$

CONSTRUCTING THE FINAL CLASSIFIER (cont'd)

For an arbitrary support vector $\mathbf{x}^j$ (then $\alpha_j > 0$), the Karush-Kuhn-Tucker condition (2) implies $y^j(\mathbf{w} \cdot \mathbf{x}^j + b) = 1$, and we can compute b as follows:

$$b = y^j - \mathbf{w} \cdot \mathbf{x}^j = y^j - \sum_{i=1}^{l} \alpha_i y^i \mathbf{x}^i \cdot \mathbf{x}^j$$

It is recommended, however, not to base the computation of b on only one support vector (for reasons of numerical precision), but to compute a b value for each support vector and to use the average finally.

Under specific conditions (e.g. asymmetric misclassification costs), it may be useful to adjust b according to some other quality measure after training.

THE NON-SEPARABLE CASE: MOTIVATION

If positive and negative samples are not linearly separable, the constraints contradict each other; thus the method described above cannot be applied. This problem can be solved by introducing non-negative *slack variables* ξ_i $(i = 1, \ldots, l)$ that correspond to the extent to which the i-th sample violates its constraint:

$$y^i(\mathbf{w} \cdot \mathbf{x}^i + b) \geq 1 - \xi_i$$

Of course, we have to require the slack variables to be as small as possible. This is achieved by adding the sum of the slack variables to the objective function, scaled with a cost factor C. We refer to this idea as the *linear C-SVM* in the following.

LINEAR C-SVM: THE PRIMAL PROBLEM

For a given data set $\mathbf{Z}$, minimize

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

with respect to $\mathbf{w} \in \mathbb{R}^d$, $b \in \mathbb{R}$, and $(\xi_1, \ldots, \xi_l) \in \mathbb{R}^l$ subject to the following constraints:

$$\left.\begin{aligned} y^i(\mathbf{w}\cdot\mathbf{x}^i + b) - 1 + \xi_i &\geq 0 \\ \xi_i &\geq 0 \end{aligned}\right\} \quad \text{for all } i = 1, \ldots, l$$

LINEAR C-SVM: LAGRANGE FUNCTION

We introduce Lagrange multipliers $\alpha_1, \ldots, \alpha_l$ and $\lambda_1, \ldots, \lambda_l$. Then the Lagrange function is given as

$$
\begin{aligned}
&L(\mathbf{w}, b, \xi_1, \ldots, \xi_l; \alpha_1, \ldots, \alpha_l, \lambda_1, \ldots, \lambda_l) \\
&\quad = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l}\xi_i - \sum_{i=1}^{l}\alpha_i\left(y^i(\mathbf{w}\cdot\mathbf{x}^i + b) - 1 + \xi_i\right) - \sum_{i=1}^{l}\lambda_i\xi_i \\
&\quad = \frac{1}{2}\|\mathbf{w}\|^2 - \mathbf{w}\cdot\sum_{i=1}^{l}\alpha_i y^i\mathbf{x}^i - b\sum_{i=1}^{l}\alpha_i y^i + \sum_{i=1}^{l}\alpha_i + \sum_{i=1}^{l}(C - \alpha_i - \lambda_i)\xi_i
\end{aligned}
$$

LINEAR C-SVM: DUAL FORMULATION (1/3)

Solving the dual problem includes minimizing L with respect to $\mathbf{w}$, b and $\xi_1, \ldots, \xi_l$ (for fixed Lagrange multipliers). This enforces the conditions

$$\frac{\partial L}{\partial \mathbf{w}}(\mathbf{w}, b, \xi_1, \ldots, \xi_l; \alpha_1, \ldots, \alpha_l, \lambda_1, \ldots, \lambda_l) = 0,$$
$$\frac{\partial L}{\partial b}(\mathbf{w}, b, \xi_1, \ldots, \xi_l; \alpha_1, \ldots, \alpha_l, \lambda_1, \ldots, \lambda_l) = 0,$$
$$\frac{\partial L}{\partial \xi_j}(\mathbf{w}, b, \xi_1, \ldots, \xi_l; \alpha_1, \ldots, \alpha_l, \lambda_1, \ldots, \lambda_l) = 0, \qquad \text{for all } j = 1, \ldots, l$$

which again imply

$$\mathbf{w} = \sum_{i=1}^{l} \alpha_i y^i \mathbf{x}^i \qquad \sum_{i=1}^{l} \alpha_i y^i = 0$$

and, additionally, $C - \alpha_j - \lambda_j = 0$ for all $j = 1, \ldots, l$.

LINEAR C-SVM: DUAL FORMULATION (2/3)

The equalities $C - \alpha_j - \lambda_j = 0$ imply that we may substitute $\lambda_j = C - \alpha_j$. The constraints $\lambda_j \geq 0$ further imply that we must ensure $C - \alpha_j \geq 0$, hence $\alpha_j \leq C$ for all $j = 1, \ldots, l$.

Finally, we obtain the same objective function

$$\mathcal{L}(\alpha_1, \ldots, \alpha_l) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y^i y^j \mathbf{x}^i \cdot \mathbf{x}^j.$$

The final solution can be found by maximizing $\mathcal{L}$ with respect to the Lagrange multipliers α_i subject to the constraints $\alpha_i \geq 0$ (for all $i = 1, \ldots, l$), $\sum_{i=1}^{l} \alpha_i y^i = 0$, and the *additional constraints* $\alpha_i \leq C$ (for all $i = 1, \ldots, l$).

LINEAR C-SVM: DUAL FORMULATION (3/3)

With the same conventions as above, we can write the dual problem as follows:

Minimize

$$\frac{1}{2}\boldsymbol{\alpha}^T \mathbf{Q} \boldsymbol{\alpha} - \mathbf{1}^T \boldsymbol{\alpha}$$

with respect to $\boldsymbol{\alpha}$ subject to the constraints $\mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{1}$ and $\boldsymbol{\alpha}^T \mathbf{y} = 0$.

Again, this is a convex quadratic optimization problem with linear constraints, so we can efficiently determine a global minimum.

LINEAR C-SVM: CONSTRUCTING THE FINAL CLASSIFIER

Analogously to above, the final classification function is given as

$$g(\mathbf{x}) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b) = \text{sign}\left(\sum_{i=1}^{l} \alpha_i y^i \mathbf{x}^i \cdot \mathbf{x} + b\right).$$

The computation of b, however, requires a bit more caution. In the non-separable case, the Karush-Kuhn-Tucker conditions tell us that

$$\alpha_i \left(y^i(\mathbf{w} \cdot \mathbf{x}^i + b) - 1 + \xi_i\right) = 0$$

holds for all $i = 1, \ldots, l$. So, if we choose an i such that $\alpha_i > 0$, we would need the value ξ_i to determine b.

LINEAR C-SVM: CONSTRUCTING THE FINAL CLASSIFIER (cont'd)

However, note that the Karush-Kuhn-Tucker conditions also imply (for the other set of constraints $\xi_i \geq 0$) that

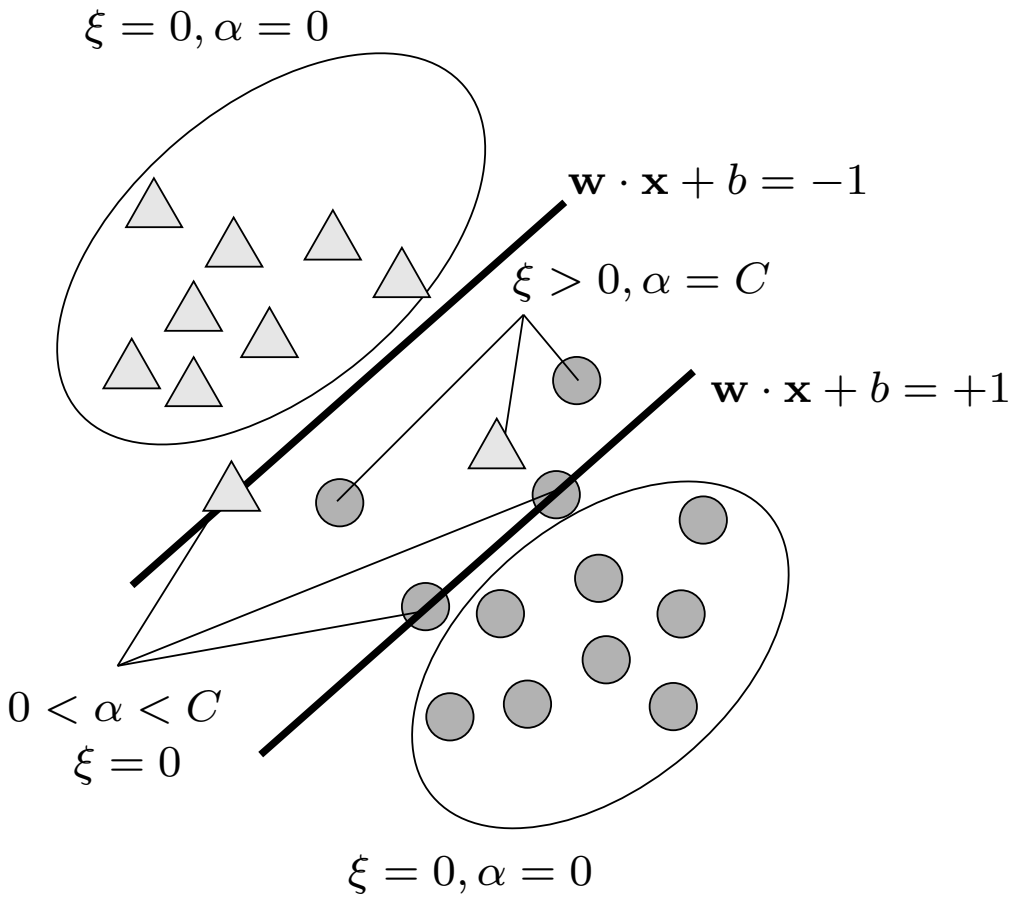
$$\lambda_i \xi_i = (C - \alpha_i)\xi_i = 0$$

holds for all $i = 1, \ldots, l$. So if we manage to find a j such that $0 < \alpha_j < C$ holds, we can infer $\xi_j = 0$ and, thus, $y^j(\mathbf{w} \cdot \mathbf{x}^j + b) - 1 = 0$, i.e. we can use the same method as described above:

$$b = y^j - \mathbf{w} \cdot \mathbf{x}^j = y^j - \sum_{i=1}^{l} \alpha_i y^i \mathbf{x}^i \cdot \mathbf{x}^j$$

It may only happen in degenerate cases that no α_j exists such that $0 < \alpha_j < C$ holds (see literature).

LINEAR C-SVM: INTERPRETING THE SOLUTION



THE NON-SEPARABLE CASE: AN ALTERNATIVE APPROACH

- In a linear C-SVM, the parameter C does not have a very intuitive interpretation (beside the obvious fact that its choice is a trade-off between minimizing the training error and maximizing the margin)
- Obviously, $\xi_i > 0$ holds if and only if

$$y^i(\mathbf{w} \cdot \mathbf{x}^i + b) < 1,$$

  i.e. $(\mathbf{x}_i, y^i)$ is a margin error with $\rho = 1$ (and, in this case, we have $\alpha_i = C$).
- An alternative linear SVM method is based on explicitly introducing a varying threshold ρ and optimizing it simultaneously. The influence of ρ on the objective function is then controlled by a factor ν. We will refer to this idea as *linear ν-SVM* in the following.

LINEAR ν-SVM: THE PRIMAL PROBLEM

For a given data set $\mathbf{Z}$, minimize

$$\frac{1}{2}\|\mathbf{w}\|^2 - \nu\rho + \frac{1}{l}\sum_{i=1}^{l}\xi_i$$

with respect to $\mathbf{w} \in \mathbb{R}^d$, $b \in \mathbb{R}$, $\rho \in \mathbb{R}$, and $(\xi_1, \ldots, \xi_l) \in \mathbb{R}^l$ subject to the following constraints:

$$\left.\begin{aligned} \rho &\geq 0 \\ y^i(\mathbf{w} \cdot \mathbf{x}^i + b) - \rho + \xi_i &\geq 0 \\ \xi_i &\geq 0 \end{aligned}\right\} \quad \text{for all } i = 1, \ldots, l$$

LINEAR ν-SVM: LAGRANGE FUNCTION

We introduce Lagrange multipliers $\alpha_1, \ldots, \alpha_l$, $\lambda_1, \ldots, \lambda_l$, and δ. Then the Lagrange function is given as

$$L(\mathbf{w}, b, \rho, \xi_1, \ldots, \xi_l; \alpha_1, \ldots, \alpha_l, \delta, \lambda_1, \ldots, \lambda_l)$$

$$= \frac{1}{2}\|\mathbf{w}\|^2 - \nu\rho + \frac{1}{l}\sum_{i=1}^{l}\xi_i - \delta\rho - \sum_{i=1}^{l}\alpha_i\left(y^i(\mathbf{w}\cdot\mathbf{x}^i + b) - \rho + \xi_i\right) - \sum_{i=1}^{l}\lambda_i\xi_i$$

$$= \frac{1}{2}\|\mathbf{w}\|^2 - \mathbf{w}\cdot\sum_{i=1}^{l}\alpha_i y^i\mathbf{x}^i - b\sum_{i=1}^{l}\alpha_i y^i$$

$$+ \rho\left(\sum_{i=1}^{l}\alpha_i - \nu - \delta\right) + \sum_{i=1}^{l}\left(\frac{1}{l} - \alpha_i - \lambda_i\right)\xi_i$$

LINEAR ν-SVM: DUAL FORMULATION (1/3)

Solving the dual problem includes minimizing L with respect to $\mathbf{w}$, b, ρ and $\xi_1, \ldots, \xi_l$ (for fixed Lagrange multipliers). This enforces the conditions

$$\begin{aligned}
\frac{\partial L}{\partial \mathbf{w}}(\mathbf{w}, b, \rho, \xi_1, \ldots, \xi_l; \alpha_1, \ldots, \alpha_l, \delta, \lambda_1, \ldots, \lambda_l) &= 0, \\
\frac{\partial L}{\partial b}(\mathbf{w}, b, \rho, \xi_1, \ldots, \xi_l; \alpha_1, \ldots, \alpha_l, \delta, \lambda_1, \ldots, \lambda_l) &= 0, \\
\frac{\partial L}{\partial \rho}(\mathbf{w}, b, \rho, \xi_1, \ldots, \xi_l; \alpha_1, \ldots, \alpha_l, \delta, \lambda_1, \ldots, \lambda_l) &= 0, \\
\frac{\partial L}{\partial \xi_i}(\mathbf{w}, b, \rho, \xi_1, \ldots, \xi_l; \alpha_1, \ldots, \alpha_l, \delta, \lambda_1, \ldots, \lambda_l) &= 0, \qquad \text{for all } i = 1, \ldots, l
\end{aligned}$$

which imply

$$\mathbf{w} = \sum_{i=1}^{l} \alpha_i y^i \mathbf{x}^i, \qquad \sum_{i=1}^{l} \alpha_i y^i = 0,$$

$\frac{1}{l} - \alpha_i - \lambda_i = 0$ (for all $i = 1, \ldots, l$), and $\sum_{i=1}^{l} \alpha_i - \nu - \delta = 0$.

LINEAR ν-SVM: DUAL FORMULATION (2/3)

The equalities $\frac{1}{l} - \alpha_i - \lambda_i = 0$ imply $\lambda_i = \frac{1}{l} - \alpha_i$ for all $i = 1, \ldots, l$. Together with $\lambda_i \geq 0$, we obtain the constraint $\alpha_i \leq \frac{1}{l}$ (for all $i = 1, \ldots, l$).
The equality $\sum_{i=1}^{l} \alpha_i - \nu - \delta = 0$ implies $\delta = \sum_{i=1}^{l} \alpha_i - \nu$, thus, by $\delta \geq 0$, we obtain the constraint $\sum_{i=1}^{l} \alpha_i \geq \nu$.

Finally, we obtain the objective function

$$\mathcal{L}(\alpha_1, \ldots, \alpha_l) = -\frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y^i y^j \mathbf{x}^i \cdot \mathbf{x}^j .$$

The final solution can be found by maximizing $\mathcal{L}$ with respect to the Lagrange multipliers α_i subject to the constraints $0 \leq \alpha_i \leq \frac{1}{l}$ (for all $i = 1, \ldots, l$), $\sum_{i=1}^{l} \alpha_i y^i = 0$, and $\sum_{i=1}^{l} \alpha_i \geq \nu$.

LINEAR ν-SVM: DUAL FORMULATION (3/3)

With the same conventions as above, we can write the dual problem as follows:

Minimize

$$\boldsymbol{\alpha}^T \mathbf{Q} \boldsymbol{\alpha}$$

with respect to $\boldsymbol{\alpha}$ subject to the constraints $\mathbf{0} \leq \boldsymbol{\alpha} \leq \frac{1}{l}\mathbf{1}$, $\boldsymbol{\alpha}^T \mathbf{y} = 0$, and $\mathbf{1}^T \boldsymbol{\alpha} \geq \nu$.

This is again a convex quadratic optimization problem with linear constraints, so we can efficiently determine a global minimum.

LINEAR ν-SVM: CONSTRUCTING THE FINAL CLASSIFIER

Analogously to above, the final classification function is given as

$$g(\mathbf{x}) = \operatorname{sign}(\mathbf{w} \cdot \mathbf{x} + b) = \operatorname{sign}\left(\sum_{i=1}^{l} \alpha_i y^i \mathbf{x}^i \cdot \mathbf{x} + b\right).$$

The computation of b is even more tricky. The Karush-Kuhn-Tucker conditions tell us that

$$\alpha_i \left(y^i (\mathbf{w} \cdot \mathbf{x}^i + b) - \rho + \xi_i\right) = 0$$

holds for all $i = 1, \ldots, l$. So, if we choose an i such that $\alpha_i > 0$, we would need the values ξ_i *and* ρ to determine b.

LINEAR ν-SVM: CONSTRUCTING THE FINAL CLASSIFIER (cont'd)

Hence, we need to take two support vectors $\mathbf{x}^r$ and $\mathbf{x}^q$ such that $0 < \alpha_r < \frac{1}{l}$, $0 < \alpha_q < \frac{1}{l}$, $y^r = +1$, and $y^q = -1$ and solve two linear equations in two variables, b and ρ. The solutions are given as follows:

$$\rho = \frac{1}{2}\mathbf{w} \cdot (\mathbf{x}^r - \mathbf{x}^q) = \frac{1}{2}(\mathbf{x}^r - \mathbf{x}^q) \cdot \sum_{i=1}^{l} \alpha_i y^i \mathbf{x}^i$$

$$b = -\frac{1}{2}\mathbf{w} \cdot (\mathbf{x}^r + \mathbf{x}^q) = -\frac{1}{2}(\mathbf{x}^r + \mathbf{x}^q) \cdot \sum_{i=1}^{l} \alpha_i y^i \mathbf{x}^i$$

It is again possible (for the sake of numerical precision) to compute the solution by averaging over two equally large sets of positive and negative support vectors all fulfilling $0 < \alpha_i < \frac{1}{l}$.

LINEAR ν-SVM: INTERPRETING THE PARAMETER ν

Theorem. Assume that we are given a ν-SVM solution according to some data set $\mathbf{Z}_l$ of l i.i.d. samples (according to a given distribution $p(\mathbf{x}, y)$) such that $\rho > 0$ holds. Then the following holds:

1. ν is an upper bound for the proportion of margin errors.
2. ν is a lower bound for the proportion of support vectors.
3. Provided that $p(\mathbf{x} \mid y = +1)$ and $p(\mathbf{x} \mid y = -1)$ do not have any discrete components, the proportions of margin errors and support vectors converge to ν with probability 1 (as l goes to infinity).

CONNECTION C-SVM – ν-SVM

Theorem. Assume that we are given a ν-SVM solution according to some data set $\mathbf{Z}_l$ such that $\rho > 0$ holds. Then exactly the same decision function (note: not necessarily the same discriminant function) would have been obtained if we had trained a C-SVM with $C = \frac{1}{\rho l}$.

SOME NOTES ON THE CHOICE OF ν

- It is clear that the constraint $\sum_{i=1}^{l} \alpha_i = \nu$ (whereas $0 \leq \alpha_i \leq \frac{1}{l}$) enforces $0 \leq \nu \leq 1$.
- Moreover, assuming that we have p positive training samples and $n = l - p$ negative training samples, the constraints $\sum_{i=1}^{l} \alpha_i = \nu$ and $\sum_{i=1}^{l} \alpha_i y^i = 0$ can only be fulfilled simultaneously if the following holds:

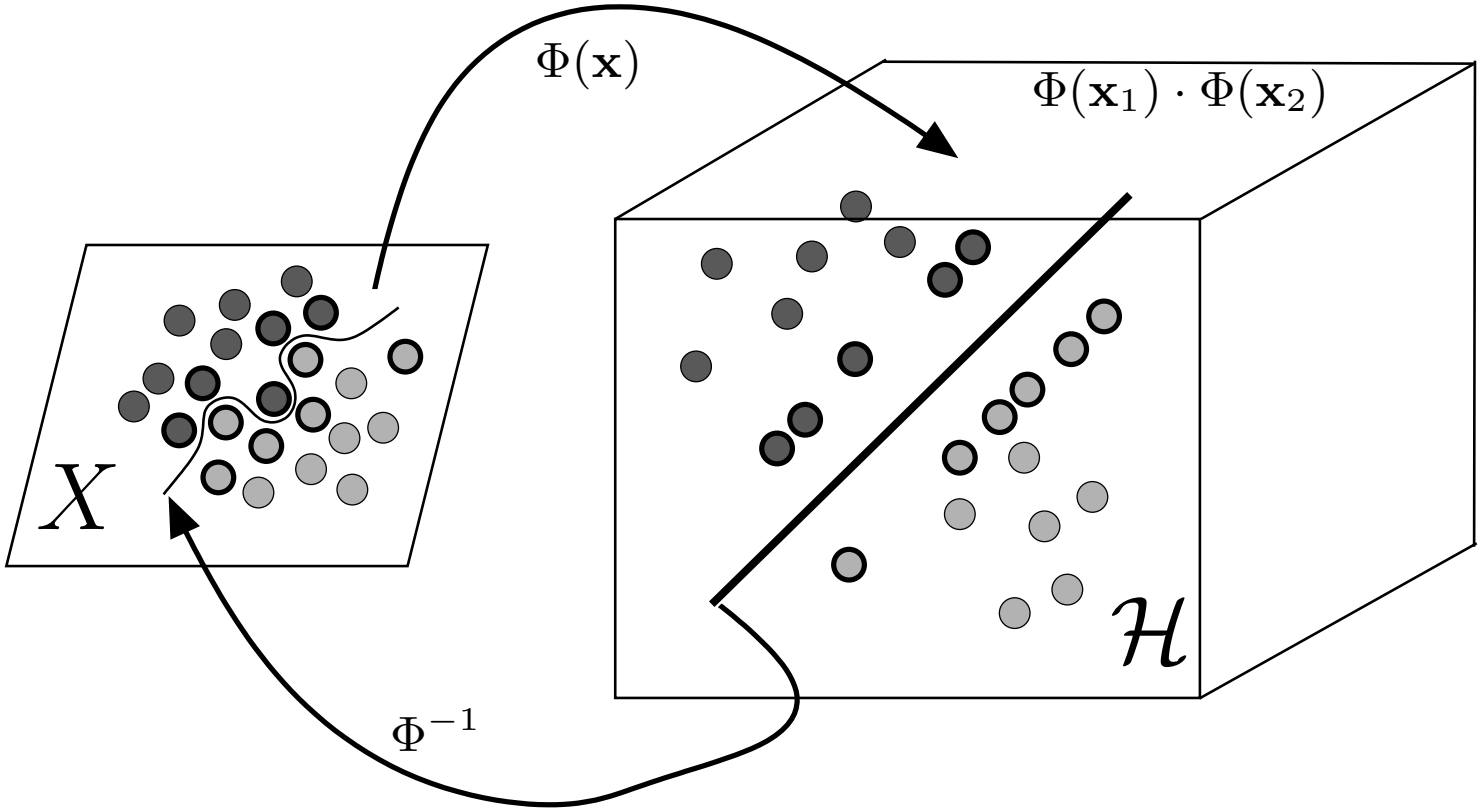
$$\nu \leq \frac{2}{l} \min(p, l - p)$$

This means that our choice of ν is strongly limited if we have a highly unbalanced data set!

NONLINEAR SUPPORT VECTOR MACHINES: INTRODUCTION

- Clearly, linear separability is a very restrictive assumption. The higher the dimensionality, however, the easier we can achieve linear separability for a given number of samples l.
- Nonlinear support vector machines are based on the idea of transforming the data into a higher-dimensional space in a way that the given problem hopefully becomes (almost) linearly separable in this space, i.e. we choose a Hilbert space $\mathcal{H}$ and a (nonlinear) mapping $\Phi : X \to \mathcal{H}$.
- Then, hypothetically, we could apply the linear method presented previously in the space $\mathcal{H}$.
- The obvious problem is how to specify $\mathcal{H}$ and Φ.

NONLINEAR SVMs: INTRODUCTION (cont'd)



NONLINEAR SVMs: THE BASIC IDEA

In solving the dual problem and computing the final classification function, we have *only scalar products of pairs of samples* appear. Therefore, it is not necessary to explicitly know $\mathcal{H}$ and Φ.

- For solving the dual problem, it is sufficient to know $\Phi(\mathbf{x}^i) \cdot \Phi(\mathbf{x}^j)$ for all pairs of training samples $\mathbf{x}^i, \mathbf{x}^j$ $(i, j = 1, \ldots, l)$.
- For computing the classification of a new sample $\mathbf{x}$, it is sufficient to know $\Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}^i)$ for all $i = 1, \ldots, l$.

So suppose we are given a mapping $k : X \times X \to \mathbb{R}$ (the so-called *kernel*) for which we know that there exists a Hilbert space $\mathcal{H}$ and a mapping $\Phi : X \to \mathcal{H}$ such that $k(\mathbf{x}, \mathbf{y}) = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{y})$ holds for all $\mathbf{x}, \mathbf{y} \in X$.

NONLINEAR SVMs: THE KERNEL TRICK

- Normally, one would assume that the kernel k should be chosen specifically suited to the given learning task. However, this is often too hard to do.
- Instead, it is usual to make an a priori choice of the kernel k using common sense and, if available, prior knowledge about the problem.
- To replace scalar products by an a priori choice of a kernel in order to "non-linearize" a given algorithm is often termed *"kernel trick"*. It can be applied to any algorithm that uses only scalar products—including, among a lot of others, support vector machines.

C-SVM: DUAL PROBLEM (1/3)

Applying the kernel trick to the linear C-SVM, we obtain the following optimization problem:[a]

Maximize

$$\mathcal{L}(\alpha_1, \ldots, \alpha_l) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y^i y^j k(\mathbf{x}^i, \mathbf{x}^j).$$

with respect to the Lagrange multipliers α_i subject to the constraints $0 \leq \alpha_i \leq C$ (for all $i = 1, \ldots, l$) and $\sum_{i=1}^{l} \alpha_i y^i = 0$.

[a] Note that we do not bother about considering the separable case here. Most often, we cannot check/guarantee linear separability in the (unknown) Hilbert space $\mathcal{H}$ anyway.

C-SVM: DUAL PROBLEM (2/3)

Let the vectors $\mathbf{0}, \mathbf{1}, \boldsymbol{\alpha}$ be defined as above. With the definition

$$\mathbf{Q} = \left(y^i y^j k(\mathbf{x}^i, \mathbf{x}^j)\right)_{i=1,\ldots,l}^{j=1,\ldots,l},$$

we can write the dual problem as follows:

Minimize

$$\frac{1}{2}\boldsymbol{\alpha}^T \mathbf{Q} \boldsymbol{\alpha} - \mathbf{1}^T \boldsymbol{\alpha}$$

w.r.t. $\boldsymbol{\alpha}$ subject to the constraints $\mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{1}$ and $\boldsymbol{\alpha}^T \mathbf{y} = 0$.

C-SVM: DUAL PROBLEM (3/3)

If we can be sure that $k(\mathbf{x}, \mathbf{y}) = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{y})$ holds for some choice of $\mathcal{H}$ and Φ, we know that $\mathbf{K} = (k(\mathbf{x}^i, \mathbf{x}^j))_{i=1,\ldots,l}^{j=1\ldots,l}$ is a positive semi-definite Gram matrix. Hence, $\mathbf{Q}$ is also positive semi-definite and the optimization problem above is again convex and quadratic with linear constraints. Regardless of the possibly non-linear kernel, we can apply the same methods for solving it.

C-SVM: CONSTRUCTING THE FINAL CLASSIFIER

Analogously to above, the final classification function is given as

$$g(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{l} \alpha_i y^i k(\mathbf{x}^i, \mathbf{x}) + b\right).$$

The threshold b can be computed as

$$b = y^j - \sum_{i=1}^{l} \alpha_i y^i k(\mathbf{x}^i, \mathbf{x}^j)$$

for a given support vector $\mathbf{x}^j$ fulfilling $0 < \alpha_j < C$ (or as an average of this value for several support vectors fulfilling this condition).

ν-SVM: DUAL PROBLEM

Applying the kernel trick to the linear ν-SVM, we obtain the following optimization problem:

Maximize

$$\mathcal{L}(\alpha_1, \ldots, \alpha_l) = -\frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y^i y^j k(\mathbf{x}^i, \mathbf{x}^j).$$

with respect to the Lagrange multipliers α_i subject to the constraints $0 \leq \alpha_i \leq \frac{1}{l}$ (for all $i = 1, \ldots, l$), $\sum_{i=1}^{l} \alpha_i y^i = 0$, and $\sum_{i=1}^{l} \alpha_i \geq \nu$.

ν-SVM: DUAL PROBLEM (cont'd)

With the same conventions as above, we can write the dual problem as follows:

Minimize

$$\boldsymbol{\alpha}^T \mathbf{Q} \boldsymbol{\alpha}$$

with respect to $\boldsymbol{\alpha}$ subject to the constraints $\mathbf{0} \leq \boldsymbol{\alpha} \leq \frac{1}{l}\mathbf{1}$, $\boldsymbol{\alpha}^T \mathbf{y} = 0$, and $\mathbf{1}^T \boldsymbol{\alpha} \geq \nu$.

This is again a convex quadratic optimization problem with linear constraints, so we can efficiently determine a global minimum.

ν-SVM: CONSTRUCTING THE FINAL CLASSIFIER

Analogously to above, the final classification function is given as

$$g(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{l} \alpha_i y^i k(\mathbf{x}^i, \mathbf{x}) + b\right).$$

Now choose two support vectors $\mathbf{x}^r$ and $\mathbf{x}^q$ such that $0 < \alpha_r < \frac{1}{l}$, $0 < \alpha_q < \frac{1}{l}$, $y^r = +1$, and $y^q = -1$. Then ρ and b can be computed as follows:

$$\rho = \frac{1}{2}\left(\sum_{i=1}^{l} \alpha_i y^i k(\mathbf{x}^i, \mathbf{x}^r) - \sum_{i=1}^{l} \alpha_i y^i k(\mathbf{x}^i, \mathbf{x}^q)\right)$$

$$b = -\frac{1}{2}\left(\sum_{i=1}^{l} \alpha_i y^i k(\mathbf{x}^i, \mathbf{x}^r) + \sum_{i=1}^{l} \alpha_i y^i k(\mathbf{x}^i, \mathbf{x}^q)\right)$$

ν-SVM: SOME NOTES

- Analogously to the linear case, ρ and b can be computed using more than just a pair of support vectors.
- The theorem concerning the interpretation of ν also holds for the general ν-SVM, with only minor modifications. 1. and 2. hold in the same way. For 3., we have to assume that the kernel is not constant and analytic.
- The theorem establishing the connection C-SVM – ν-SVM holds without any modification.
- The notes concerning the choice of ν apply in the same way. As a consequence, the limitations of the ν-SVM for highly unbalanced data sets persist.

WHICH MAPPINGS $k(.,.)$ ARE APPROPRIATE?

It is clear that we cannot choose $k(.,.)$ completely arbitrarily. *Mercer's theorem* provides us with a necessary and sufficient condition under which a mapping k can be considered a meaningful kernel.

Theorem. A continuous two-place mapping $k : X^2 \to \mathbb{R}$ can be represented by $k(\mathbf{x}, \mathbf{y}) = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{y})$ for some choice of a Hilbert space $\mathcal{H}$ and an $X \to \mathcal{H}$ mapping Φ if any only if

$$\int_{X^2} k(\mathbf{x}, \mathbf{y}) f(\mathbf{x}) f(\mathbf{y}) d\mathbf{x} d\mathbf{y} \geq 0 \tag{3}$$

holds for all square-integrable functions $f \in L^2(X)$.

WHICH MAPPINGS $k(.,.)$ ARE APPROPRIATE? (cont'd)

Mercer's condition (3) can be understood as the positive semi-definiteness of k. If it is fulfilled, we can be sure that the Gram matrix $\mathbf{K} = (k(\mathbf{x}^i, \mathbf{x}^j))_{i=1,\ldots,l}^{j=1\ldots,l}$ is positive semi-definite for any choice of training data $\mathbf{x}^1, \ldots, \mathbf{x}^l$; thus, the dual problem is a convex quadratic optimization problem. Moreover, we can be sure that generalized derivatives exist such that solving the dual problem is equivalent to solving a (hypothetical) primal problem.

STANDARD KERNELS

The following kernels are often used in practice:

Linear: $k(\mathbf{x}, \mathbf{y}) = \mathbf{x} \cdot \mathbf{y}$

Polynomial: $k(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + \beta)^{\alpha}$

Gaussian/RBF:[a] $k(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{x} - \mathbf{y}\|^2\right)$

Sigmoid: $k(\mathbf{x}, \mathbf{y}) = \tanh(\alpha \mathbf{x} \cdot \mathbf{y} + \beta)$

[a] RBF = Radial Basis Function

STANDARD KERNELS: SOME NOTES

- The sigmoid kernel is not a very popular choice; moreover, it is not positive semi-definite for all choices of α and β.
- *The RBF kernel is the most popular choice.*
- As the RBF kernel can only take values from $[0, 1]$, it maps into a hyper-sphere of radius 1.
- The VC dimension of SVMs with RBF kernel is infinite.
- The Hilbert space corresponding to the RBF kernel is infinitely dimensional.

CUSTOM KERNELS

It is not as difficult to define new kernels as it may seem at first glance:

- If we can define the Hilbert space $\mathcal{H}$ (most often $\mathbb{R}^k$) and the mapping Φ explicitly, we are safe (e.g. spectrum and mismatch kernel in bioinformatics).
- Products, weighted sums (and a lot more operations) applied to positive semi-definite kernels give semi-definite kernels.
- Suppose that we have a mapping $\Psi : X \to Y$, where Y is some *feature space*, and a semi-definite kernel $k : Y^2 \to \mathbb{R}$. Then $k' : X^2 \to \mathbb{R}$, defined as $k'(\mathbf{x}, \mathbf{y}) = k(\Psi(\mathbf{x}), \Psi(\mathbf{y}))$ is also a positive semi-definite kernel.

ENSURING POSITIVE SEMI-DEFINITE GRAM MATRICEs

- It is easy to see that adding a constant e to the diagonal of a symmetric matrix shifts all eigenvalues by e; hence, we can make an indefinite symmetric matrix positive semi-definite by subtracting the smallest eigenvalue (or a lower bound for it).
- We can apply this trick to SVMs: if we have a "Gram matrix" $\mathbf{K}$ which is not positive semi-definite, we can make it positive semi-definite by adding a sufficiently large constant to the diagonal.
- This heuristic lacks mathematical foundation, but often works well in practice (e.g. Smith-Waterman "kernel" for sequence analysis).

SOME NOTES ON COMPLEXITY OF SVMS

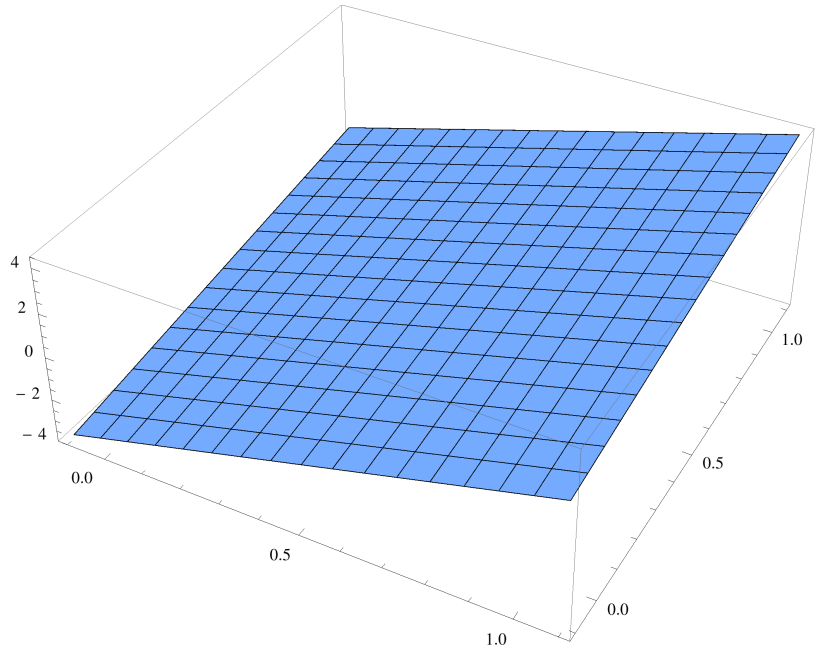
Although SVMs are motivated by simultaneously minimizing complexity, there are issues related to complexity left.

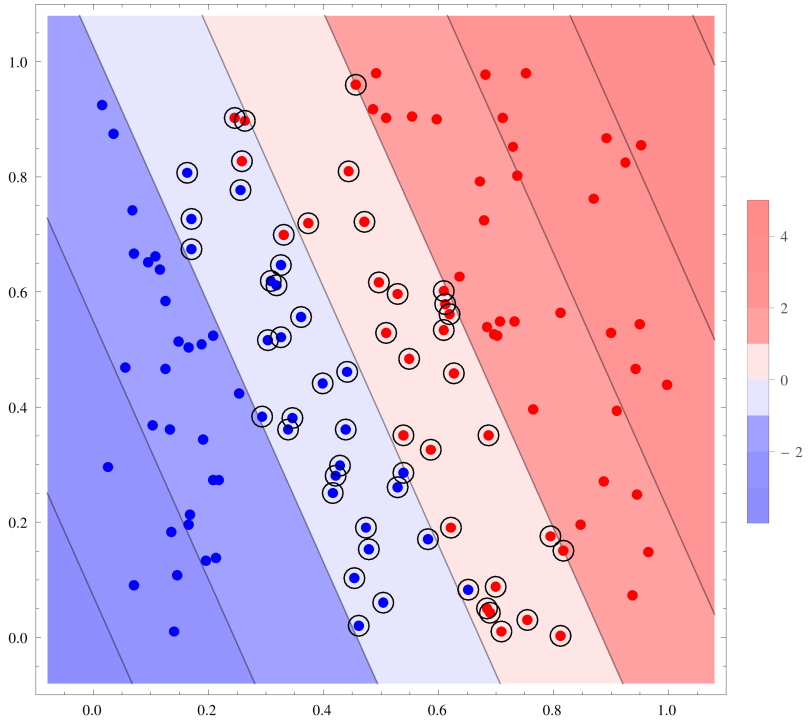
- If the RBF kernel is used, the choice of σ is crucial (note: infinite VC dimension if we admit any choice of σ): too large $\sigma \rightarrow$ underfitting; too small $\sigma \rightarrow$ overfitting.
- If the polynomial kernel is used, the degree α is crucial; the VC dimension grows polynomially with α.
- The choices of C or ν also influence complexity. The higher C and ν, the more we punish misclassifications, hence, the higher the tendency of the SVM to produce a more complex model.

It is often unavoidable to use cross validation to find good choices for hyperparameters.

C-SVM EXAMPLE #1:

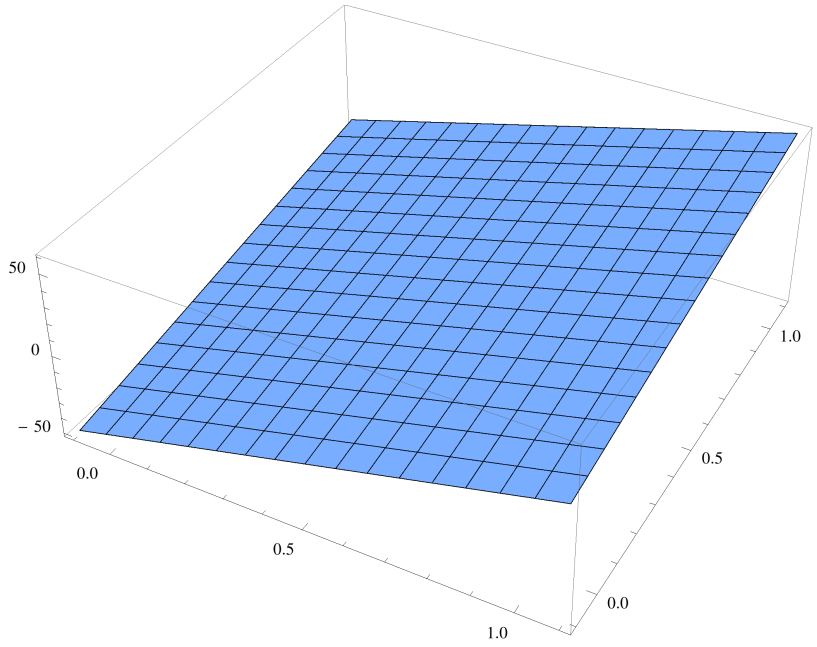
$C = 1$, LINEAR KERNEL

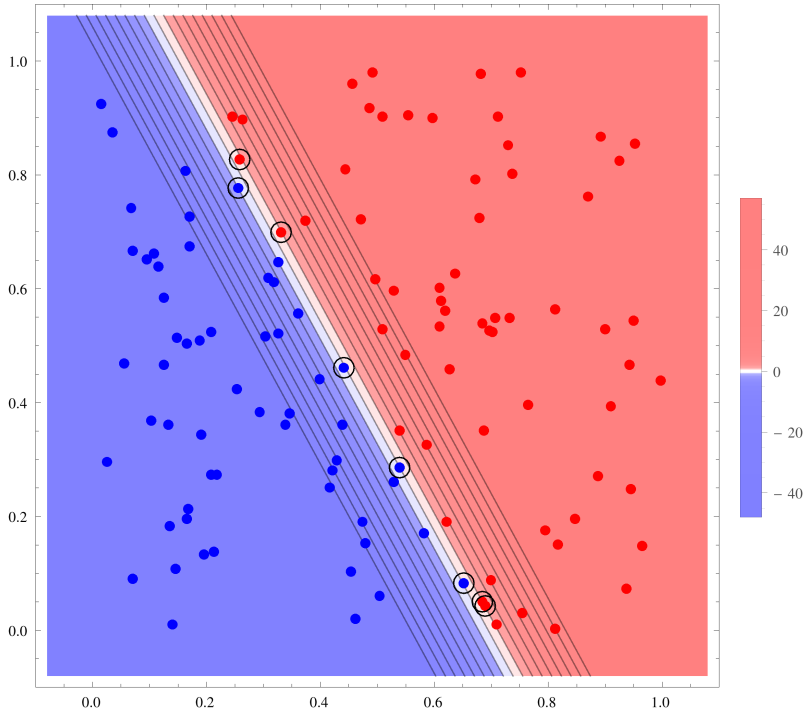




C-SVM EXAMPLE #1:

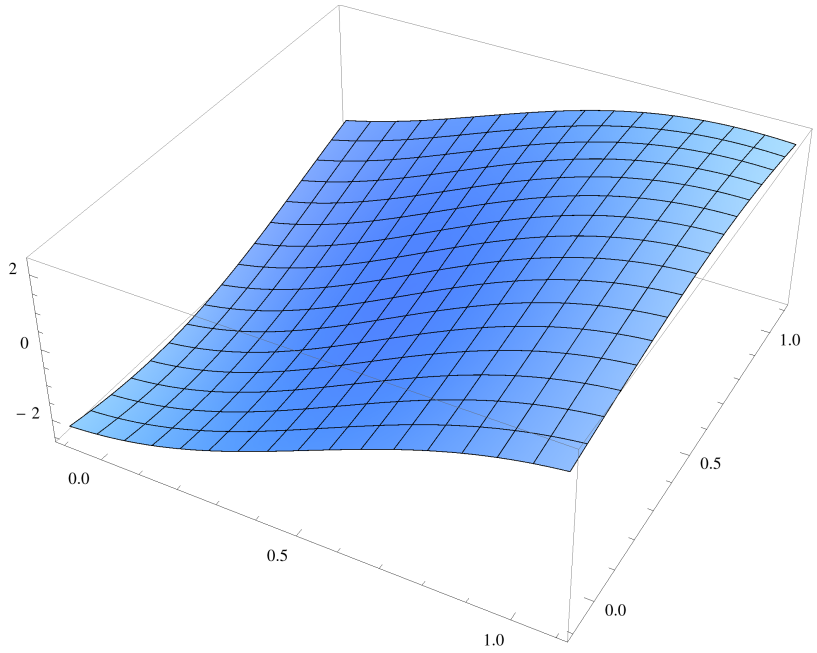
$C = 1000$, LINEAR KERNEL

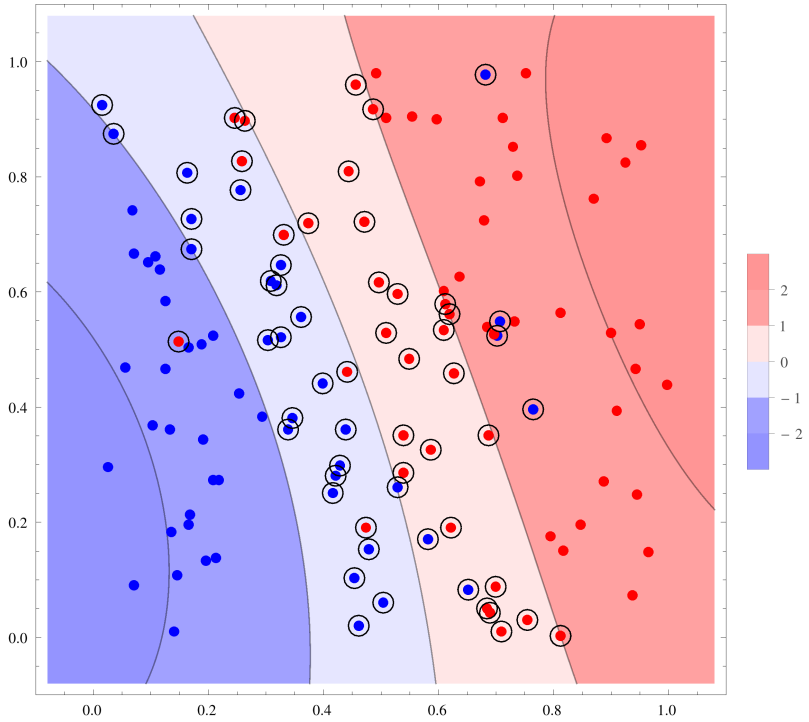




C-SVM EXAMPLE #2:

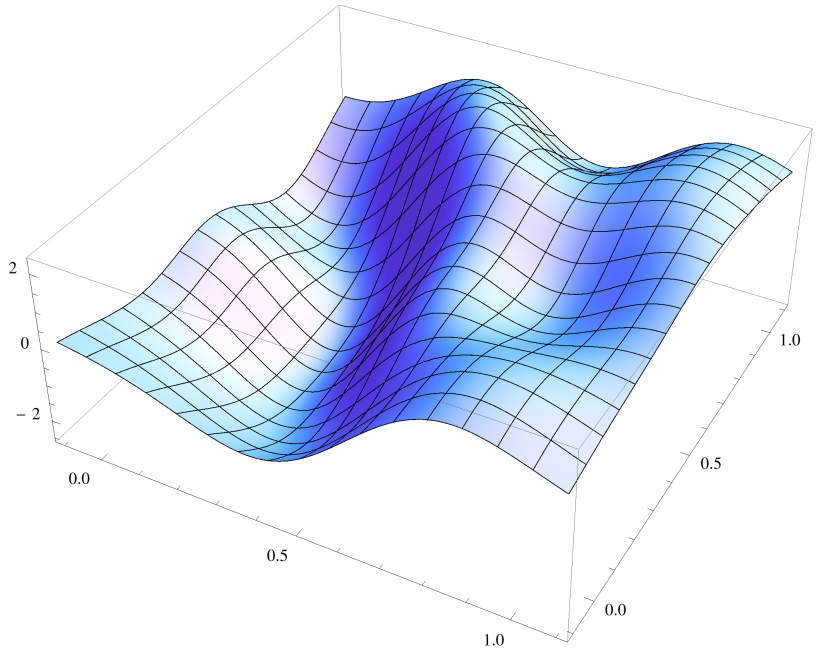
$C = 1$, RBF KERNEL, $\frac{1}{2\sigma^2} = 1$

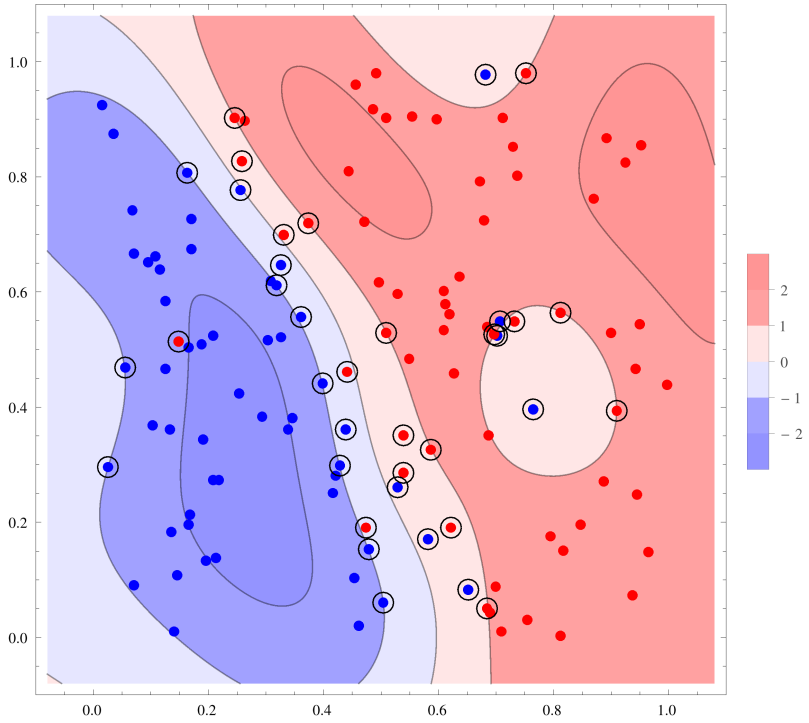




C-SVM EXAMPLE #2:

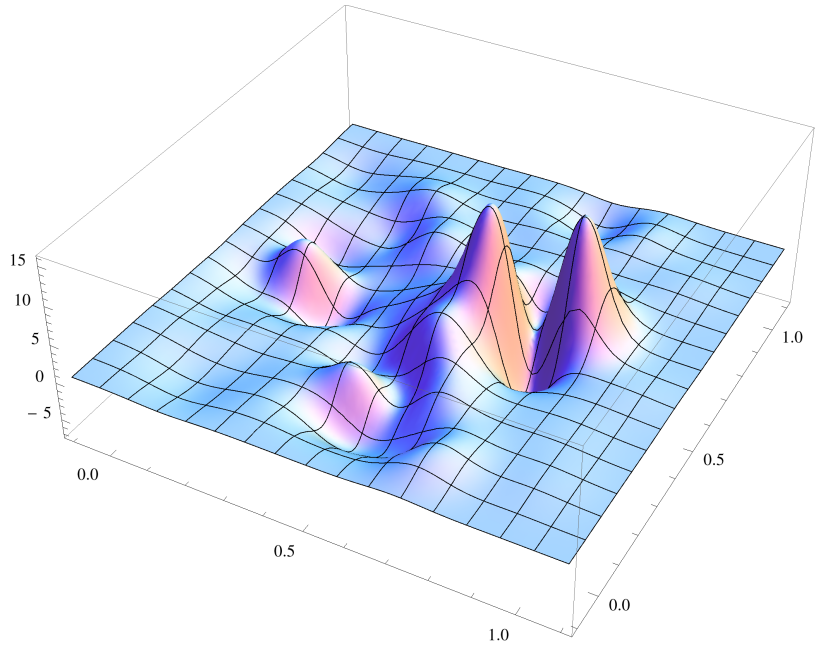
$C = 10$, RBF KERNEL, $\frac{1}{2\sigma^2} = 10$

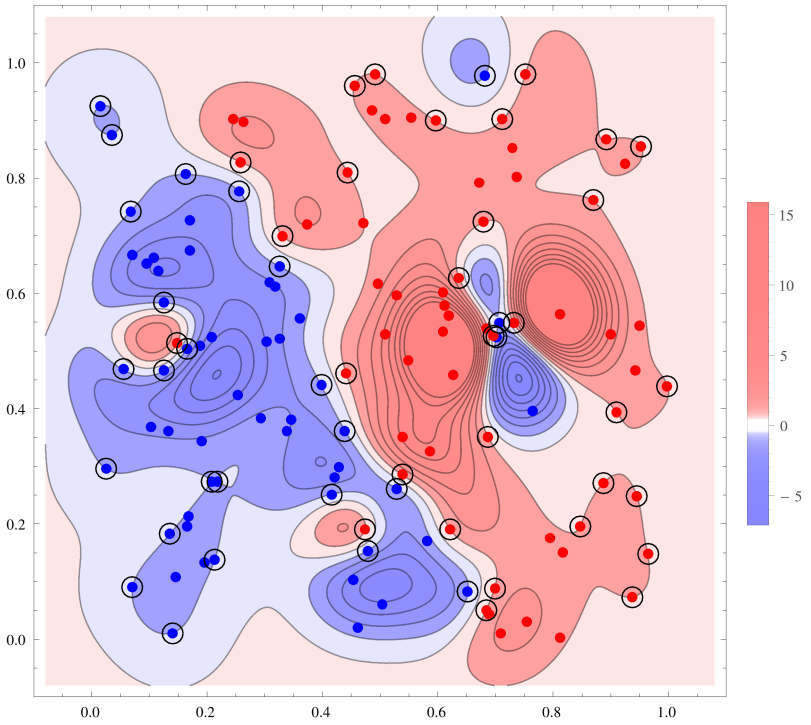




C-SVM EXAMPLE #2:

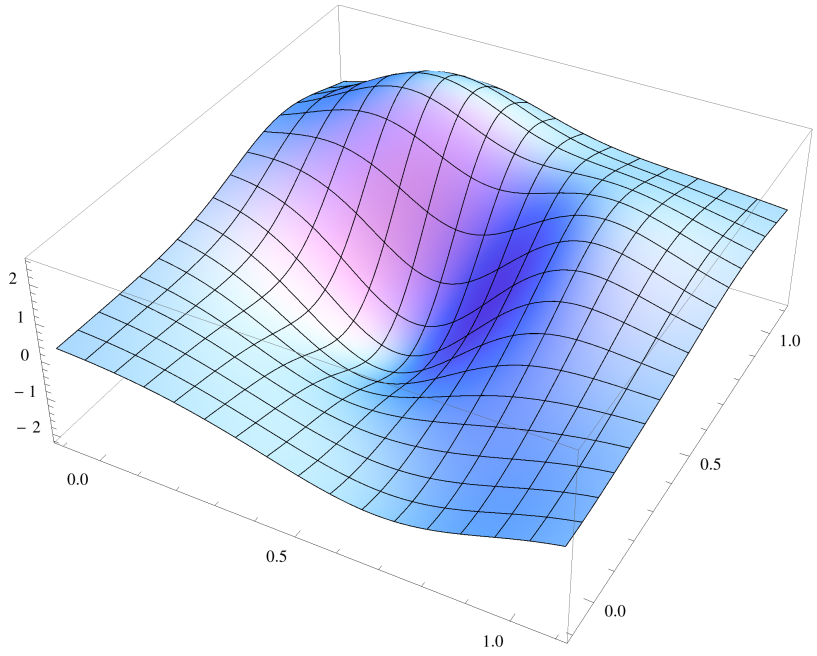
$C = 1000$, RBF KERNEL, $\frac{1}{2\sigma^2} = 100$

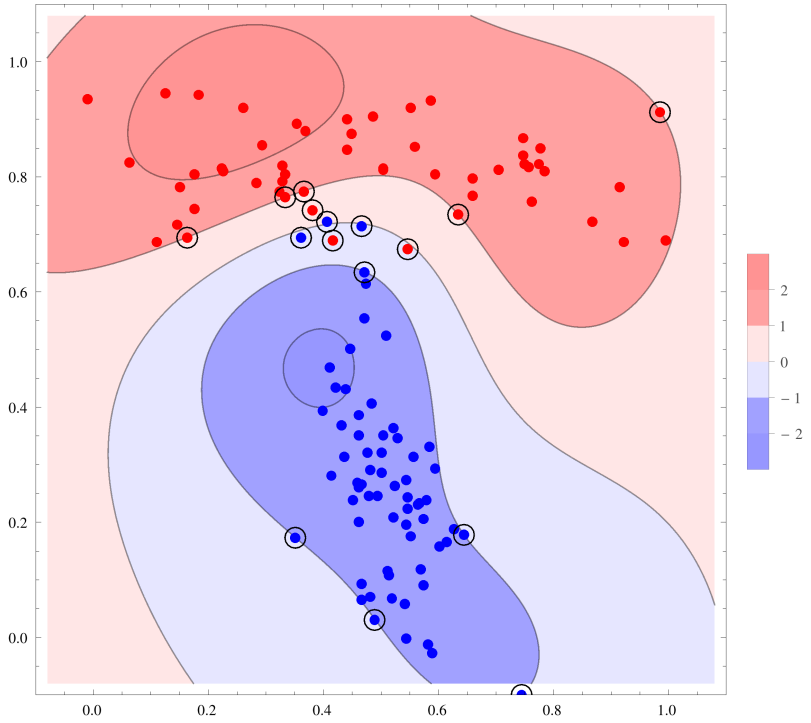




C-SVM EXAMPLE #3:

C-SVM, $C = 10$, RBF KERNEL, $\frac{1}{2\sigma^2} = 10$





KERNELS FOR BIOLOGICAL SEQUENCES

The following family of kernels is quite common for biological sequences:

$$k(x, y) = \sum_{m \in \mathcal{M}} N(m, x) \cdot N(m, y),$$

where $\mathcal{M}$ is a set of *patterns* and $N(m, x)$ denotes the number of occurrences/matches of pattern m in string x. Obviously, the explicit representation of the mapping φ is given as follows:

$$\varphi(x) = (N(m, x))_{m \in \mathcal{M}}.$$

COMMON SEQUENCE KERNELS

Spectrum kernel: $\mathcal{M}$ is the set of all K-length strings (exact matches)

Mismatch kernel: $\mathcal{M}$ is the set of all K-length strings (matches with up to M mismatches)

Motif kernel: $\mathcal{M}$ is a predefined set of problem-specific patterns (possibly including wildcards, maybe even general RegExp's)

Gappy pair kernel: $\mathcal{M}$ is the set of pairs of symbols with at most M positions in between

EXAMPLES OF SEQUENCE KERNELS (FEATURES)

Spectrum kernel ($K = 3$):

```
MKQLEDKVEELLSKTYHLENEVARL
MKQ
 KQL
  QLE
   LED
    EDK
     DKV
      KVE
       VEE
        EEL
         ELL
          LLS
           LSK
            SKT
             KTY
              TYH
               YHL
                HLE
                 LEN
                  ENE
                   NEV
                    EVA
                     VAR
                      ARL
```

Gappy pair kernel ($M = 3$):

```
MKQLEDKVEELLSKTYHLENEVARL
MK
M.Q
M..L
M...E
 KQ
 K.L
 K..E
 K...D
  QL
  Q.E
  Q..D
  Q...K
   LE
   L.D
   L..K
   L...V
    ED
    E.K
    E..V
    E...E
     DK
     D.V
     D..E [...]
```

EASY WEIGHT EXTRACTION

$$
\begin{aligned}
f(x) &= b + \sum_{i=1}^{l} \alpha_i \cdot y^i \cdot k(x, x^i) \\
&= b + \sum_{i=1}^{l} \alpha_i \cdot y^i \cdot \sum_{m \in \mathcal{M}} N(m, x) \cdot N(m, x^i) \\
&= b + \sum_{m \in \mathcal{M}} N(m, x) \cdot \underbrace{\sum_{i=1}^{l} \alpha_i \cdot y^i \cdot N(m, x^i)}_{=w(m)}
\end{aligned}
$$

For each pattern m, the absolute value of $w(m)$ provides information about the importance of pattern and the sign of $w(m)$ tells for which class an occurrence of m is indicative.

OTHER SEQUENCE KERNELS

Position-dependent kernels: the above kernels can be generalized to take positions of patterns into account (includes weighted degree kernel and shifted weighted degree kernels as special cases)

Pairwise kernel: the vector of scores of pairwise alignments to a given reference/training set is used as input features

Smith-Waterman "kernel": the score of the optimal local alignment is used; note: not generally positive semi-definite!

Local alignment kernel: based on Smith-Waterman-like local alignments, but guaranteed to be positive semi-definite

...

KERNELS FOR SIGNAL AND IMAGE PROCESSING

- One possible approach is to extract features (frequency spectrum, wavelets, filters, etc.) and to use regular SVMs on feature vectors
- If the signals/images are not excessively large, also standard kernels like the linear kernel or the RBF kernel can be used on the data directly; note that this usually requires many training samples to obtain decent models
- Another quite common approach (at least in image processing) is to use small patches of images and to use standard kernels (linear, RBF, correlation, etc.) on these patches; this approach is particularly useful for detection of relatively small objects in images (geometric shapes, faces, letters, digits)

EXAMPLE: FACE DETECTION

(Osuna et al., 1997)

SVM-BASED APPROACHES TO MULTI-CLASS PROBLEMS

- Support vector machines are intrinsically based on the idea of separating two classes by maximizing the margin between them. So there is no obvious way to extend them to multi-class problems.
- All approaches introduced so far are based on breaking down the multi-class problem into several binary classification problems.

MULTI-CLASS SVM APPROACHES: ONE VERSUS THE REST

Given a training set $\mathbf{Z}_l = (\mathbf{x}_i, y_i)_{i=1,\ldots,l}$, where $y_i \in \{1, \ldots, M\}$, M SVM classifiers are trained to separate one class from the remaining $M-1$ ones, i.e. we train M binary SVM classifiers ($j = 1, \ldots, M$)

$$\bar{g}_j(\mathbf{x}) = \sum_{i=1}^{l} \alpha_{ij} y_j^i k(\mathbf{x}^i, \mathbf{x}) + b_j,$$

where

$$y_j^i = \begin{cases} +1 & \text{if } y^i = j, \\ -1 & \text{otherwise.} \end{cases}$$

MULTI-CLASS SVM APPROACHES: ONE VERSUS THE REST (cont'd)

Then the final classification for a given sample $\mathbf{x}$ is defined as

$$\arg \max_{j=1,\ldots,M} \bar{g}_j(\mathbf{x}),$$

which is basically a "winner-takes-it-all" approach.

Disadvantages:

- Most likely, all M sub-problems are unbalanced, even if the classes are evenly distributed.
- There is no way to guarantee that the discriminant functions g_j are on comparable scales.

MULTI-CLASS SVM APPROACHES: MULTI-CLASS OBJECTIVE (1/3)

Consider a training set as above. Then the primal multi-class problem is given as follows (we restrict to the linear case first):

Minimize

$$\frac{1}{2}\sum_{j=1}^{M}\|\mathbf{w}_j\|^2 + C\sum_{i=1}^{l}\sum_{j\neq y_i}\xi_{ij}$$

with respect to $\mathbf{w}_j \in \mathbb{R}^d$, $b_j \in \mathbb{R}$, and $(\xi_{ij})_{i=1,\ldots,l}^{j=1,\ldots,M}$ (where $j = 1,\ldots,M$) subject to the constraints

$$\mathbf{w}_{y_i}\cdot\mathbf{x}^i + b_{y_i} \geq \mathbf{w}_j\cdot\mathbf{x}^i + b_j + 2 - \xi_{ij} \qquad \xi_{ij} \geq 0$$

(for all $i = 1,\ldots,l$ and all $j = 1,\ldots,M$ such that $j \neq y_i$).

MULTI-CLASS SVM APPROACHES: MULTI-CLASS OBJECTIVE (2/3)

- Once the optimization problem has been solved, the final classification of a new sample $\mathbf{x}$ is computed as

$$\arg \max_{j=1,\ldots,M} \mathbf{w}_j \cdot \mathbf{x} + b_j,$$

  i.e. this corresponds to a one-versus-the-rest approach with the difference that all M classifiers are simultaneously trained by solving one joint optimization problem.
- The generalization to the non-linear case is straightforward if the dual problem is considered.

MULTI-CLASS SVM APPROACHES: MULTI-CLASS OBJECTIVE (2/3)

- The problem of different scalings of discriminant function does not occur here, as they are jointly optimized with “coupled” slack variables. That is why this is considered a very elegant approach.
- However, the results do not generally outperform the one-against-all approach and the computational effort for solving the multi-objective problem is significantly higher (see literature).

MULTI-CLASS SVM APPROACHES: PAIRWISE CLASSIFICATION

Consider a training set as above. For every pair of indices $j, k \in \{1, \ldots, M\}$ (without loss of generality, assume $j < k$), we select those samples from the training set for which y^i is j or k; let us denote these training sets with $\mathbf{Z}_{jk}$. Similar to above, we assign labels $+1$ to the samples that originally belonged to class j and -1 to the samples that originally belonged to class k and train a binary SVM classifier on this binary problem. So, in total $\frac{M(M-1)}{2}$ SVMs are trained.

Once this is done, a new sample $\mathbf{x}$ is assigned to that class that has obtained the most "votes" from the pairwise classifiers.

MULTI-CLASS SVM APPROACHES: PAIRWISE CLASSIFICATION (cont'd)

- The computational effort for training $\frac{M(M-1)}{2}$ pairwise classifiers is, in average, not higher than for the one-versus-the-rest classifiers, as the sizes of the training sets are smaller. Taking into account that the effort for training an SVM grows superlinearly with the number of samples, the asymptotic complexity of pairwise classification is even lower than for one-versus-the-rest classification.
- The classification of new samples, however, may be slower, yet some improvements are possible (see literature).
- *Presently, pairwise classification is the most common approach.*

SUPPORT VECTOR REGRESSION (SVR): INTRODUCTION

- So far, we have mainly been interested in the sign of the discriminant function of a support vector machine. The constraints in the resulting optimization problems were designed to maintain equal signs of the training labels and the discriminant function, but the magnitude of the discriminant function was neglected (except inside the margin).
- The SVMs considered so far are, therefore, useless for regression tasks.
- However, if we managed to reformulate the constraints such that the value of the discriminant function at a certain training input is pushed to the actual label value, we could generalize the SVM idea to regression.

ε-INSENSITIVE LOSS AND ε-TUBES

The *ε-insensitive loss function* L_ε is defined as

$$L_\varepsilon(y, g(\mathbf{x})) = \max(0, |y - g(\mathbf{x})| - \varepsilon)$$

Obviously, $L_\varepsilon(y, g(\mathbf{x})) = 0$ if and only if $|y - g(\mathbf{x})| \leq \varepsilon$. Hence, the ε-insensitive loss defines an ε-tube around the regression function g and checks for a given sample whether it is inside this ε-tube. If not, the loss of the sample is defined as the distance to the ε-tube.

The basic idea behind support vector regression is to adjust the regression function such that the data points are within the/an ε-tube.

LINEAR ε-SVR: THE PRIMAL PROBLEM

For a given data set $\mathbf{Z}$, minimize

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l}(\xi_i^+ + \xi_i^-)$$

with respect to $\mathbf{w} \in \mathbb{R}^d$, $b \in \mathbb{R}$, $(\xi_1^+, \ldots, \xi_l^+) \in \mathbb{R}^l$, and $(\xi_1^-, \ldots, \xi_l^-) \in \mathbb{R}^l$ subject to the following constraints:

$$\left.\begin{aligned} y^i - (\mathbf{w}\cdot\mathbf{x}^i + b) &\leq \varepsilon + \xi_i^+ \\ (\mathbf{w}\cdot\mathbf{x}^i + b) - y_i &\leq \varepsilon + \xi_i^- \\ \xi_i^+ &\geq 0 \\ \xi_i^- &\geq 0 \end{aligned}\right\} \quad \text{for all } i = 1, \ldots, l$$

LINEAR ε-SVR: INTERPRETATION (1/3)

- We still try to minimize $\frac{1}{2}\|\mathbf{w}\|^2$ which is nothing else but the steepness of the regression function. Of course, this has nothing to do with margin maximization anymore, but it can still be understood as a measure of complexity.
- Obviously, the slack variables ξ_i^+ measure to which extent y^i is above the ε-tube around the regression function; the values ξ_i^- measure to which extent y^i is below this ε-tube. The sum of slack variables is added to the objective function to ensure simultaneous minimization of the slack values.
- The parameter C controls the trade-off between accuracy (low slack values) and complexity (flat regression function).

LINEAR ε-SVR: INTERPRETATION (2/3)

In the case $\varepsilon = 0$, we can reformulate the optimization problem as follows:

Minimize

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l}|\mathbf{w}\cdot\mathbf{x}^i + b - y_i|$$

with respect to $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$ (without any constraints).

Hence, for very large C, we can interpret the ε-SVR with $\varepsilon = 0$ as simple data fitting according to the absolute value (norm/loss). For smaller C, the importance of the term $\frac{1}{2}\|\mathbf{w}\|^2$ increases.

LINEAR ε-SVR: INTERPRETATION (3/3)

Finally, we can state that the ε-SVR is a kind of ε-insensitive minimization of the training error according to the absolute value loss (corresponding to the sum of slack values). The term $\frac{1}{2}\|\mathbf{w}\|^2$ is rather a *regularization/capacity term* than the primary objective.

LINEAR ε-SVR: LAGRANGE FUNCTION

For brevity, denote

$$\boldsymbol{\xi}^+ = (\xi_1^+, \ldots, \xi_l^+)^T, \qquad \boldsymbol{\xi}^- = (\xi_1^-, \ldots, \xi_l^-)^T,$$
$$\boldsymbol{\alpha}^+ = (\alpha_1^+, \ldots, \alpha_l^+)^T, \qquad \boldsymbol{\alpha}^- = (\alpha_1^-, \ldots, \alpha_l^-)^T,$$
$$\boldsymbol{\lambda}^+ = (\lambda_1^+, \ldots, \lambda_l^+)^T, \qquad \boldsymbol{\lambda}^- = (\lambda_1^-, \ldots, \lambda_l^-)^T.$$

Then the Lagrange function is given as follows:

$$\begin{aligned} &L(\mathbf{w}, b, \boldsymbol{\xi}^+, \boldsymbol{\xi}^-; \boldsymbol{\alpha}^+, \boldsymbol{\alpha}^-, \boldsymbol{\lambda}^+, \boldsymbol{\lambda}^-) \\ &\quad = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l}(\xi_i^+ + \xi_i^-) - \sum_{i=1}^{l}\alpha_i^+(\varepsilon + \xi_i^+ - y^i + \mathbf{w}\cdot\mathbf{x}^i + b) \\ &\qquad - \sum_{i=1}^{l}\alpha_i^-(\varepsilon + \xi_i^- + y^i - \mathbf{w}\cdot\mathbf{x}^i - b) - \sum_{i=1}^{l}\lambda_i^+\xi_i^+ - \sum_{i=1}^{l}\lambda_i^-\xi_i^- \end{aligned}$$

LINEAR ε-SVR: LAGRANGE FUNCTION (cont'd)

We can rewrite the Lagrange function as follows:

$$
\begin{aligned}
&L(\mathbf{w}, b, \boldsymbol{\xi}^+, \boldsymbol{\xi}^-; \boldsymbol{\alpha}^+, \boldsymbol{\alpha}^-, \boldsymbol{\lambda}^+, \boldsymbol{\lambda}^-) \\
&\quad = \frac{1}{2}\|\mathbf{w}\|^2 - \mathbf{w} \cdot \sum_{i=1}^{l}(\alpha_i^+ - \alpha_i^-)\mathbf{x}^i - b\sum_{i=1}^{l}(\alpha_i^+ - \alpha_i^-) \\
&\qquad + \sum_{i=1}^{l}(C - \alpha_i^+ - \lambda_i^+)\xi_i^+ + \sum_{i=1}^{l}(C - \alpha_i^- - \lambda_i^-)\xi_i^- \\
&\qquad - \varepsilon\sum_{i=1}^{l}(\alpha_i^+ + \alpha_i^-) + \sum_{i=1}^{l} y^i(\alpha_i^+ - \alpha_i^-)
\end{aligned}
$$

LINEAR ε-SVR: THE DUAL PROBLEM (1/3)

Minimizing the Lagrange function with respect to $\mathbf{w}$, b, $\boldsymbol{\xi}^+$ and $\boldsymbol{\xi}^-$ enforces the following:

$$\frac{\partial L}{\partial \mathbf{w}}(\mathbf{w}, b, \boldsymbol{\xi}^+, \boldsymbol{\xi}^-; \boldsymbol{\alpha}^+, \boldsymbol{\alpha}^-, \boldsymbol{\lambda}^+, \boldsymbol{\lambda}^-) = \mathbf{w} - \sum_{i=1}^{l}(\alpha_i^+ - \alpha_i^-)\mathbf{x}^i = \mathbf{0}$$

$$\frac{\partial L}{\partial b}(\mathbf{w}, b, \boldsymbol{\xi}^+, \boldsymbol{\xi}^-; \boldsymbol{\alpha}^+, \boldsymbol{\alpha}^-, \boldsymbol{\lambda}^+, \boldsymbol{\lambda}^-) = -\sum_{i=1}^{l}(\alpha_i^+ - \alpha_i^-) = 0$$

$$\frac{\partial L}{\partial \xi_i^+}(\mathbf{w}, b, \boldsymbol{\xi}^+, \boldsymbol{\xi}^-; \boldsymbol{\alpha}^+, \boldsymbol{\alpha}^-, \boldsymbol{\lambda}^+, \boldsymbol{\lambda}^-) = -(C - \alpha_i^+ - \lambda_i^+) = 0$$

$$\frac{\partial L}{\partial \xi_i^-}(\mathbf{w}, b, \boldsymbol{\xi}^+, \boldsymbol{\xi}^-; \boldsymbol{\alpha}^+, \boldsymbol{\alpha}^-, \boldsymbol{\lambda}^+, \boldsymbol{\lambda}^-) = -(C - \alpha_i^- - \lambda_i^-) = 0$$

Hence, we obtain $\mathbf{w} = \sum_{i=1}^{l}(\alpha_i^+ - \alpha_i^-)\mathbf{x}^i$ and the constraint $\sum_{i=1}^{l}(\alpha_i^+ - \alpha_i^-) = 0$.

LINEAR ε-SVR: THE DUAL PROBLEM (2/3)

Moreover, analogously to the C-SVM, we can eliminate the Lagrange multipliers λ_i^+ and λ_i^- by simply adding the constraints $\alpha_i^+ \leq C$ and $\alpha_i^- \leq C$. Thus, we obtain the following dual problem:
Maximize

$$\mathcal{L}(\boldsymbol{\alpha}^+, \boldsymbol{\alpha}^-) = -\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i^+ - \alpha_i^-)(\alpha_j^+ - \alpha_j^-)\mathbf{x}^i \cdot \mathbf{x}^j$$
$$- \varepsilon\sum_{i=1}^{l}(\alpha_i^+ + \alpha_i^-) + \sum_{i=1}^{l} y^i(\alpha_i^+ - \alpha_i^-)$$

with respect to $\boldsymbol{\alpha}^+$ and $\boldsymbol{\alpha}^-$ subject to the constraints $0 \leq \alpha_i^+ \leq C$, $0 \leq \alpha_i^- \leq C$ $(i = 1, \ldots, l)$, and $\sum_{i=1}^{l}(\alpha_i^+ - \alpha_i^-) = 0$.

LINEAR ε-SVR: THE DUAL PROBLEM (3/3)

With the above notations and the convention $\mathbf{K} = \mathbf{X}\mathbf{X}^T$, we can rewrite the dual problem as follows:

Minimize

$$\frac{1}{2}(\boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-)^T\mathbf{K}(\boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-) + \varepsilon\mathbf{1}^T(\boldsymbol{\alpha}^+ + \boldsymbol{\alpha}^-) - \mathbf{y}^T(\boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-)$$

with respect to $\boldsymbol{\alpha}^+$ and $\boldsymbol{\alpha}^-$ subject to the constraints $\mathbf{0} \leq \boldsymbol{\alpha}^+ \leq C\mathbf{1}$, $\mathbf{0} \leq \boldsymbol{\alpha}^+ \leq C\mathbf{1}$, and $\mathbf{1}^T(\boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-) = 0$.

This is again a convex quadratic optimization problem with linear constraints.

LINEAR ε-SVR: THE FINAL REGRESSION FUNCTION

Once the dual problem has been solved, the final regression function is given as

$$g(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b = \sum_{i=1}^{l} (\alpha_i^+ - \alpha_i^-) \mathbf{x}^i \cdot \mathbf{x} + b.$$

To compute b, we have to consider the Karush-Kuhn-Tucker conditions again which, in this case, enforce the following (for all $i = 1, \ldots, l$):

$$\begin{aligned}
\alpha_i^+ (\varepsilon + \xi_i^+ - y^i + \mathbf{w} \cdot \mathbf{x}^i + b) &= 0 \\
\alpha_i^- (\varepsilon + \xi_i^- + y^i - \mathbf{w} \cdot \mathbf{x}^i - b) &= 0 \\
(C - \alpha_i^+) \xi_i^+ &= 0 \\
(C - \alpha_i^-) \xi_i^- &= 0
\end{aligned}$$

LINEAR ε-SVR: THE FINAL REGRESSION FUNCTION (cont'd)

So, for any α_j^+ such that $0 < \alpha_j^+ < C$, we can infer $\xi_j^+ = 0$ and compute b as

$$b = y^j - \mathbf{w}\mathbf{x}^j - \varepsilon = y^j - \sum_{i=1}^{l}(\alpha_i^+ - \alpha_i^-)\mathbf{x}^i \cdot \mathbf{x}^j - \varepsilon.$$

This can be done in the same way for any α_j^- such that $0 < \alpha_j^- < C$. It is again numerically safer to consider all Lagrange multipliers from $]0, C[$ and to compute the average b value.

We further note that $0 < \alpha_i^+ < C$ means that $\xi_i^+ = 0$ and $y^i - \mathbf{w} \cdot \mathbf{x}^i - b = \varepsilon$ hold simultaneously, i.e. the sample $(\mathbf{x}^i, y^i)$ is on the upper border of the ε-tube around the regression function. Analogously, $0 < \alpha_i^- < C$ means that $(\mathbf{x}^i, y^i)$ is on the lower border of this ε-tube.

LINEAR ε-SVR: SUPPORT VECTORS

We can also infer the following from the Karush-Kuhn-Tucker conditions:

- For $\varepsilon > 0$, $\alpha_i^+ \alpha_i^- = 0$ holds, i.e. only one of the two Lagrange multipliers of a sample can be non-zero.
- If α_i^+ and α_i^- are both zero, this means that $\xi_i^+ = 0$ and $\xi_i^- = 0$, i.e. the i-th sample is inside the ε-tube which, in the case of the ε-SVR means that this sample does not contribute to the final regression function.
- If either $\alpha_i^+ > 0$ or $\alpha_i^- > 0$ holds, the i-th sample contributes to the regression function. In this case, we say that $(\mathbf{x}^i, y^i)$ is a *support vector*.

LINEAR ε-SVR: SUPPORT VECTORS (cont'd)

- $0 < \alpha_i^+ < C$ means that $\xi_i^+ = 0$ and $y^i - \mathbf{w} \cdot \mathbf{x}^i - b = \varepsilon$ hold simultaneously, i.e. the sample $(\mathbf{x}^i, y^i)$ is on the upper border of the ε-tube around the regression function. Analogously, $0 < \alpha_i^- < C$ means that $(\mathbf{x}^i, y^i)$ is on the lower border of the ε-tube.
- If either $\alpha_i^+ = C$ or $\alpha_i^- = C$ holds, we know that $(\mathbf{x}^i, y^i)$ is outside the ε-tube around the regression function, thus a "classification error".
- Unlike most other regression methods, accuracy is not the only goal of support vector regression. Instead, it tries to find the least complex (flattest) solution fitting into the ε-tube.

LINEAR ν-SVR: MOTIVATION

- For the ε-SVR, the choice of ε is crucial for obtaining good results.
- In practice, however, ε must be chosen according to the noise level, which is often unknown.
- The idea of ν-SVR is the following: instead of specifying ε a priori, it is optimized simultaneously, where a large ε is penalized and traded against smoothness and accuracy. The importance of ε in the objective function is weighted with a factor ν.

LINEAR ν-SVR: THE PRIMAL PROBLEM

For a given data set $\mathbf{Z}$, minimize

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\left(\nu\varepsilon + \frac{1}{l}\sum_{i=1}^{l}(\xi_i^+ + \xi_i^-)\right)$$

with respect to $\mathbf{w} \in \mathbb{R}^d$, $b \in \mathbb{R}$, $\varepsilon \in \mathbb{R}$, $(\xi_1^+, \ldots, \xi_l^+) \in \mathbb{R}^l$, and $(\xi_1^-, \ldots, \xi_l^-) \in \mathbb{R}^l$ subject to the constraints $\varepsilon \geq 0$ and

$$\left.\begin{aligned} y^i - (\mathbf{w}\cdot\mathbf{x}^i + b) &\leq \varepsilon + \xi_i^+ \\ (\mathbf{w}\cdot\mathbf{x}^i + b) - y_i &\leq \varepsilon + \xi_i^- \\ \xi_i^+ &\geq 0 \\ \xi_i^- &\geq 0 \end{aligned}\right\} \quad \text{for all } i = 1, \ldots, l.$$

LINEAR ν-SVR: LAGRANGE FUNCTION

$$
\begin{aligned}
&L(\mathbf{w}, b, \varepsilon, \boldsymbol{\xi}^+, \boldsymbol{\xi}^-; \boldsymbol{\alpha}^+, \boldsymbol{\alpha}^-, \delta, \boldsymbol{\lambda}^+, \boldsymbol{\lambda}^-) \\
&\quad = \frac{1}{2}\|\mathbf{w}\|^2 + C\Big(\nu\varepsilon + \frac{1}{l}\sum_{i=1}^{l}(\xi_i^+ + \xi_i^-)\Big) - \sum_{i=1}^{l}\alpha_i^+(\varepsilon + \xi_i^- - y^i + \mathbf{w}\cdot\mathbf{x}^i + b) \\
&\qquad - \sum_{i=1}^{l}\alpha_i^-(\varepsilon + \xi_i^- + y^i - \mathbf{w}\cdot\mathbf{x}^i - b) - \delta\varepsilon - \sum_{i=1}^{l}\lambda_i^+\xi_i^+ - \sum_{i=1}^{l}\lambda_i^-\xi_i^- \\
&\quad = \frac{1}{2}\|\mathbf{w}\|^2 - \mathbf{w}\cdot\sum_{i=1}^{l}(\alpha_i^+ - \alpha_i^-)\mathbf{x}^i - b\sum_{i=1}^{l}(\alpha_i^+ - \alpha_i^-) \\
&\qquad + \sum_{i=1}^{l}(\frac{C}{l} - \alpha_i^+ - \lambda_i^+)\xi_i^+ + \sum_{i=1}^{l}(\frac{C}{l} - \alpha_i^- - \lambda_i^-)\xi_i^- + \sum_{i=1}^{l}y^i(\alpha_i^+ - \alpha_i^-) \\
&\qquad + \varepsilon\Big(C\nu - \sum_{i=1}^{l}(\alpha_i^+ + \alpha_i^-) - \delta\Big)
\end{aligned}
$$

LINEAR ν-SVR: THE DUAL PROBLEM (1/4)

Minimizing the Lagrange function with respect to $\mathbf{w}$, b, ε, $\boldsymbol{\xi}^+$ and $\boldsymbol{\xi}^-$ enforces the following:

$$\frac{\partial L}{\partial \mathbf{w}}(\mathbf{w}, b, \boldsymbol{\xi}^+, \boldsymbol{\xi}^-; \boldsymbol{\alpha}^+, \boldsymbol{\alpha}^-, \boldsymbol{\lambda}^+, \boldsymbol{\lambda}^-) = \mathbf{w} - \sum_{i=1}^{l}(\alpha_i^+ - \alpha_i^-)\mathbf{x}^i = \mathbf{0}$$

$$\frac{\partial L}{\partial b}(\mathbf{w}, b, \boldsymbol{\xi}^+, \boldsymbol{\xi}^-; \boldsymbol{\alpha}^+, \boldsymbol{\alpha}^-, \boldsymbol{\lambda}^+, \boldsymbol{\lambda}^-) = -\sum_{i=1}^{l}(\alpha_i^+ - \alpha_i^-) = 0$$

$$\frac{\partial L}{\partial \varepsilon}(\mathbf{w}, b, \boldsymbol{\xi}^+, \boldsymbol{\xi}^-; \boldsymbol{\alpha}^+, \boldsymbol{\alpha}^-, \boldsymbol{\lambda}^+, \boldsymbol{\lambda}^-) = C\nu - \sum_{i=1}^{l}(\alpha_i^+ + \alpha_i^-) - \delta = 0$$

$$\frac{\partial L}{\partial \xi_i^+}(\mathbf{w}, b, \boldsymbol{\xi}^+, \boldsymbol{\xi}^-; \boldsymbol{\alpha}^+, \boldsymbol{\alpha}^-, \boldsymbol{\lambda}^+, \boldsymbol{\lambda}^-) = -(\frac{C}{l} - \alpha_i^+ - \lambda_i^+) = 0$$

$$\frac{\partial L}{\partial \xi_i^-}(\mathbf{w}, b, \boldsymbol{\xi}^+, \boldsymbol{\xi}^-; \boldsymbol{\alpha}^+, \boldsymbol{\alpha}^-, \boldsymbol{\lambda}^+, \boldsymbol{\lambda}^-) = -(\frac{C}{l} - \alpha_i^- - \lambda_i^-) = 0$$

LINEAR ν-SVR: THE DUAL PROBLEM (2/4)

Hence, we again obtain the following:

$$\mathbf{w} = \sum_{i=1}^{l} (\alpha_i^+ - \alpha_i^-) \mathbf{x}^i \qquad \sum_{i=1}^{l} (\alpha_i^+ - \alpha_i^-) = 0$$

Moreover, as before, we can eliminate the Lagrange multipliers λ_i^+ and λ_i^- by simply adding the constraints $\alpha_i^+ \leq \frac{C}{l}$ and $\alpha_i^- \leq \frac{C}{l}$. The Lagrange multiplier δ can also be eliminated by the additional constraint

$$\sum_{i=1}^{l} (\alpha_i^+ + \alpha_i^-) \leq C\nu.$$

LINEAR ν-SVR: THE DUAL PROBLEM (3/4)

Thus, we obtain the following dual problem:

Maximize

$$\mathcal{L}(\boldsymbol{\alpha}^+, \boldsymbol{\alpha}^-) = -\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i^+ - \alpha_i^-)(\alpha_j^+ - \alpha_j^-)\mathbf{x}^i \cdot \mathbf{x}^j + \sum_{i=1}^{l} y^i(\alpha_i^+ - \alpha_i^-)$$

with respect to $\boldsymbol{\alpha}^+$ and $\boldsymbol{\alpha}^-$ subject to the constraints $0 \leq \alpha_i^+ \leq \frac{C}{l}$, $0 \leq \alpha_i^- \leq \frac{C}{l}$ $(i = 1, \ldots, l)$, $\sum_{i=1}^{l}(\alpha_i^+ - \alpha_i^-) = 0$ and $\sum_{i=1}^{l}(\alpha_i^+ + \alpha_i^-) \leq C\nu$.

LINEAR ν-SVR: THE DUAL PROBLEM (4/4)

With the notations from above, we can rewrite the dual problem as follows:

Minimize

$$\frac{1}{2}(\boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-)^T \mathbf{K}(\boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-) - \mathbf{y}^T(\boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-)$$

with respect to $\boldsymbol{\alpha}^+$ and $\boldsymbol{\alpha}^-$ subject to the constraints $\mathbf{0} \leq \boldsymbol{\alpha}^+ \leq \frac{C}{l}\mathbf{1}$, $\mathbf{0} \leq \boldsymbol{\alpha}^+ \leq \frac{C}{l}\mathbf{1}$, $\mathbf{1}^T(\boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-) = 0$, and $\mathbf{1}^T(\boldsymbol{\alpha}^+ + \boldsymbol{\alpha}^-) \leq C\nu$.

This is again a convex quadratic optimization problem with linear constraints.

LINEAR ν-SVR: THE FINAL REGRESSION FUNCTION

Once the dual problem has been solved, the final regression function is again given as

$$g(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b = \sum_{i=1}^{l} (\alpha_i^+ - \alpha_i^-) \mathbf{x}^i \cdot \mathbf{x} + b.$$

To compute b, we have to consider the Karush-Kuhn-Tucker conditions (for all $i = 1, \ldots, l$):

$$\alpha_i^+ (\varepsilon + \xi_i^+ - y^i + \mathbf{w} \cdot \mathbf{x}^i + b) = 0 \qquad (\frac{C}{l} - \alpha_i^+)\xi_i^+ = 0$$

$$\alpha_i^- (\varepsilon + \xi_i^- + y^i - \mathbf{w} \cdot \mathbf{x}^i - b) = 0 \qquad (\frac{C}{l} - \alpha_i^-)\xi_i^- = 0$$

and, additionally,

$$(C\nu - \sum_{i=1}^{l} (\alpha_i^+ + \alpha_i^-))\varepsilon = 0. \tag{4}$$

LINEAR ν-SVR: THE FINAL REGRESSION FUNCTION (cont'd)

Suppose that there is an α_p^+ such that $0 < \alpha_p^+ < \frac{C}{l}$ (hence $\xi_p^+ = 0$) and an α_q^- such that $0 < \alpha_q^- < \frac{C}{l}$ (hence $\xi_q^- = 0$). Then we can compute b and ε by solving the following system of two linear equations:

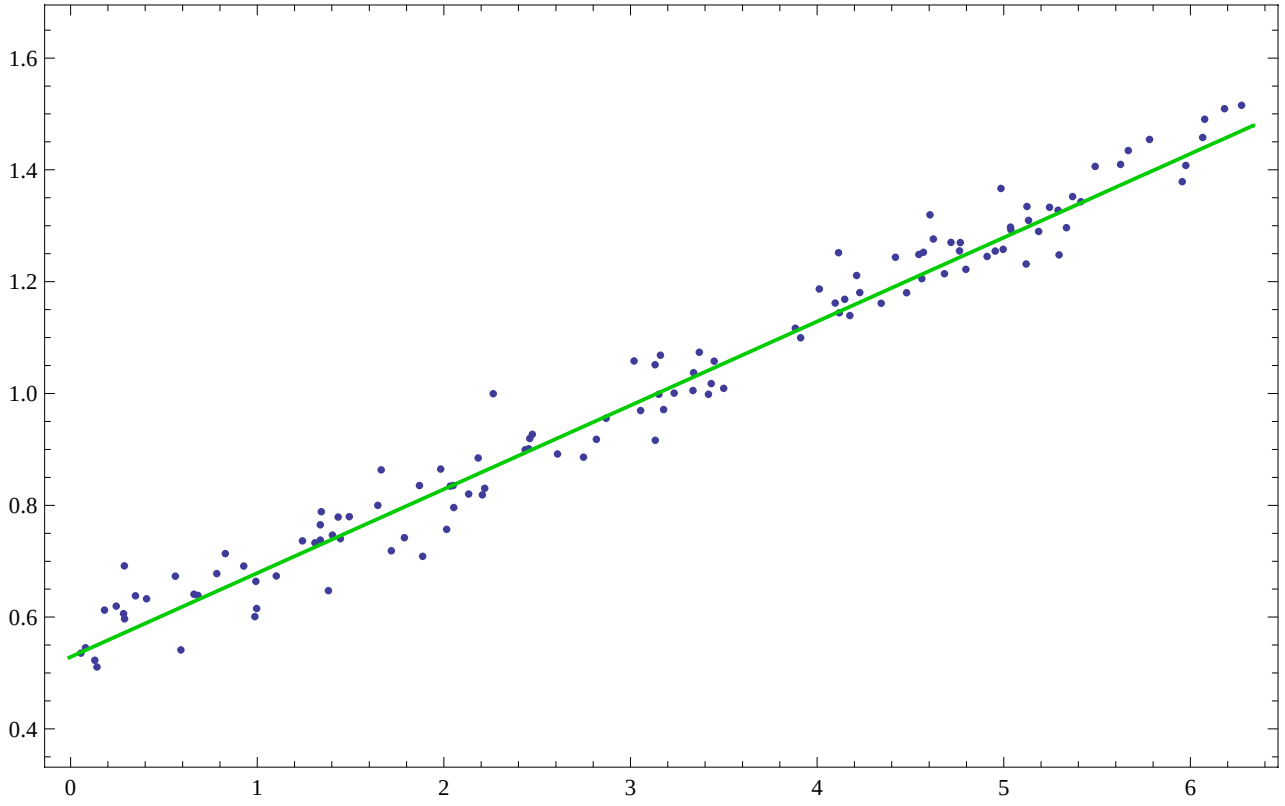
$$\varepsilon - y^p + \mathbf{w} \cdot \mathbf{x}^p + b = 0$$
$$\varepsilon + y^q - \mathbf{w} \cdot \mathbf{x}^q - b = 0$$

This gives the following solutions:

$$b = \frac{1}{2}\left((y^p - \mathbf{w} \cdot \mathbf{x}^p) + (y^q - \mathbf{w} \cdot \mathbf{x}^q)\right)$$
$$\varepsilon = \frac{1}{2}\left((y^p - \mathbf{w} \cdot \mathbf{x}^p) - (y^q - \mathbf{w} \cdot \mathbf{x}^q)\right)$$

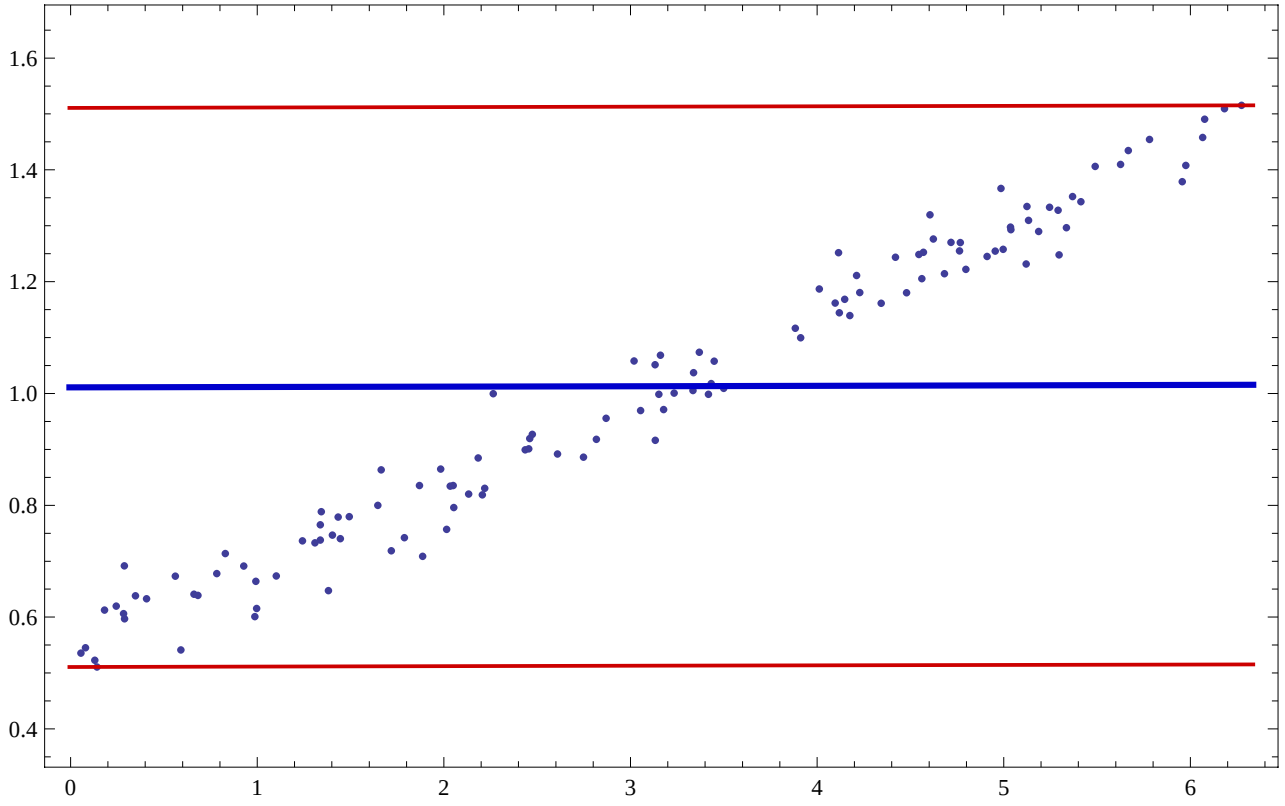
Averaging over several such pairs is again possible, of course.

LINEAR SVR EXAMPLE: AFFINE LINEAR FUNCTION PLUS NOISE



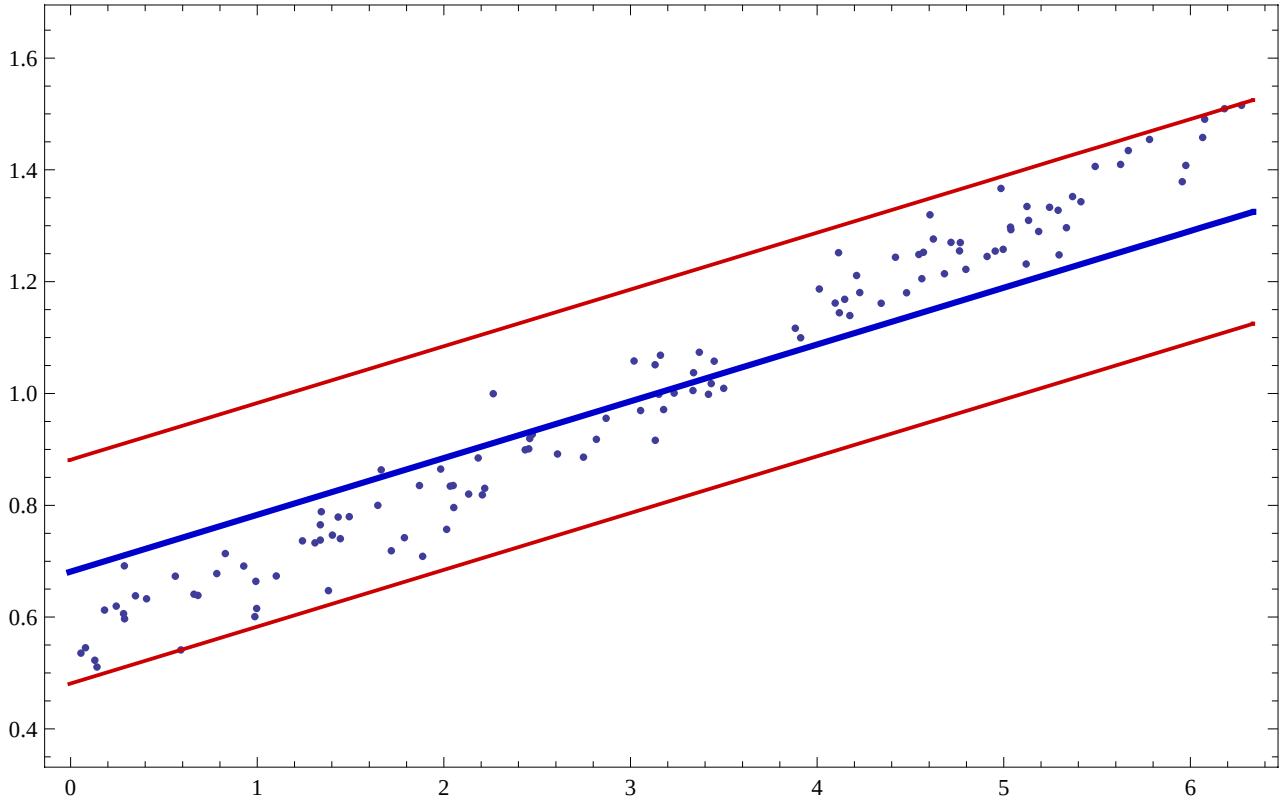
LINEAR SVR EXAMPLE:

ε-SVR, $\varepsilon = 0.5$, $C = 1$



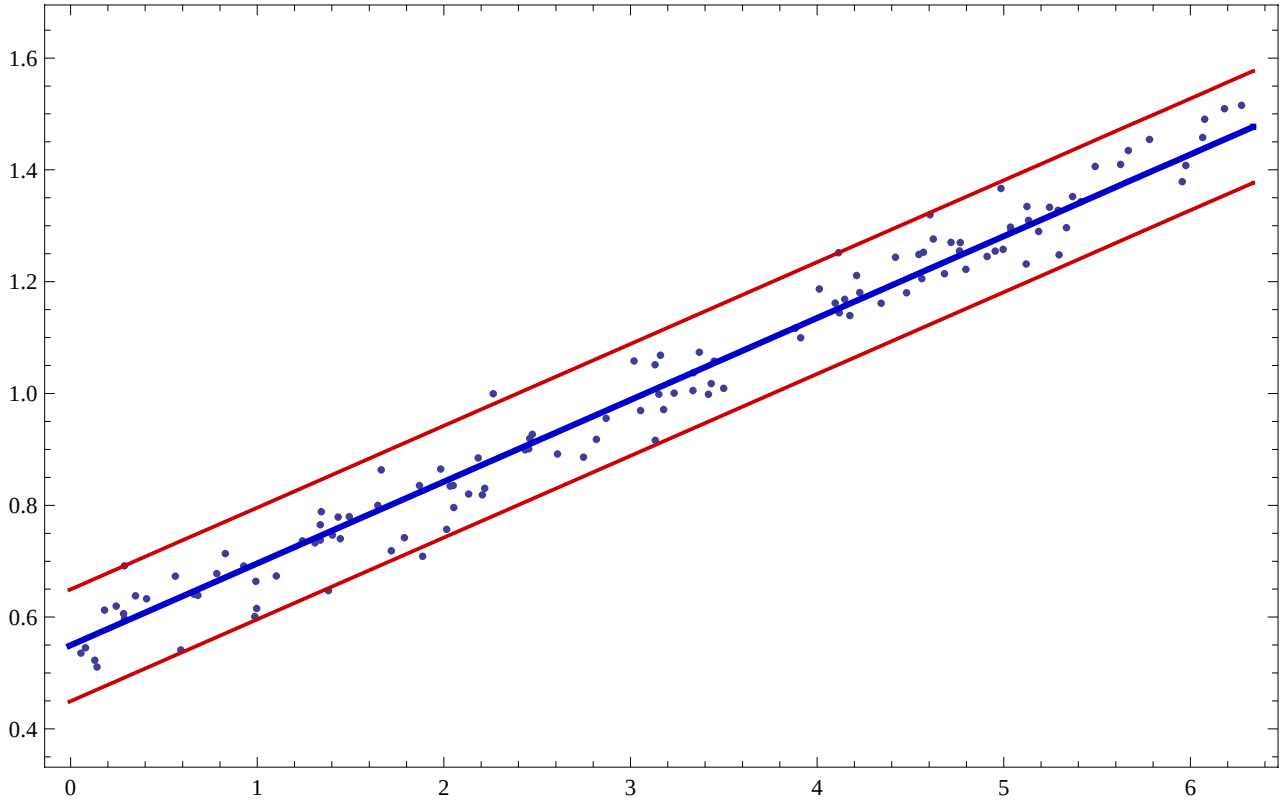
LINEAR SVR EXAMPLE:

ε-SVR, $\varepsilon = 0.2$, $C = 1$



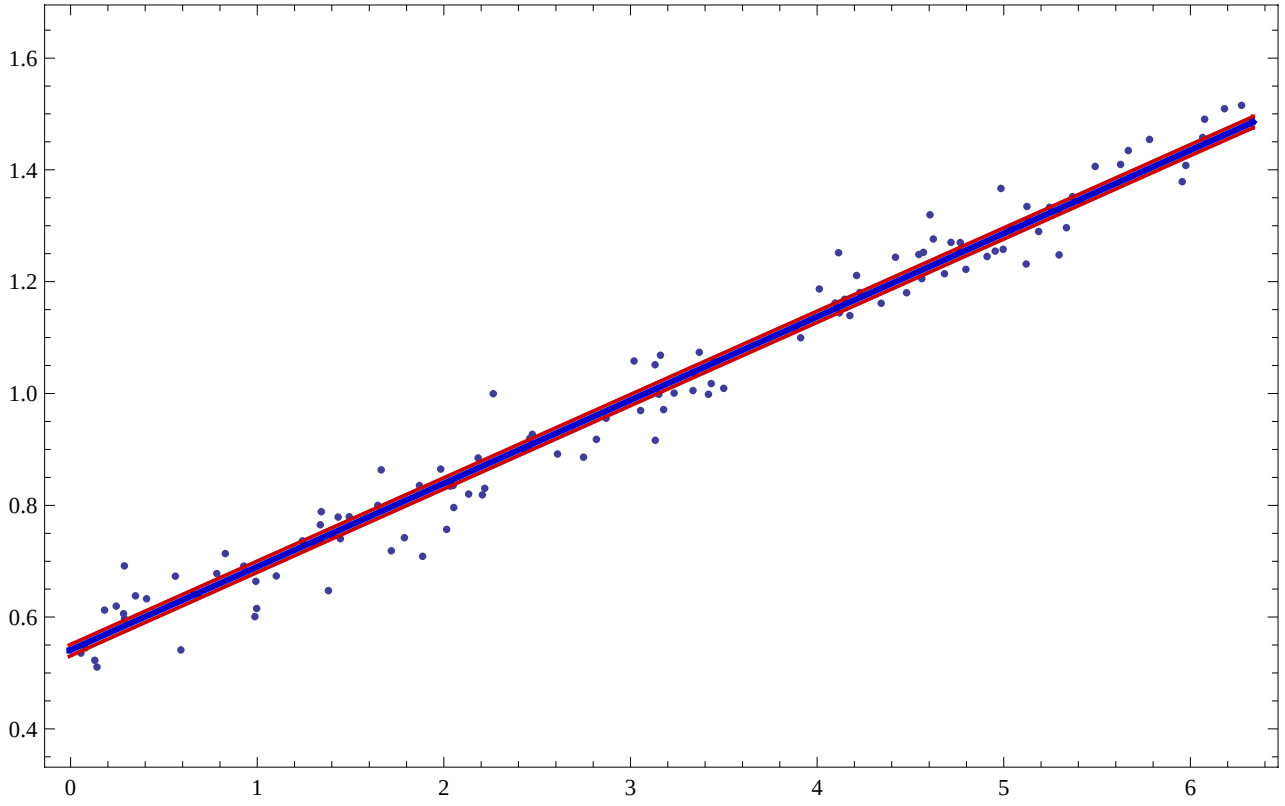
LINEAR SVR EXAMPLE:

ε-SVR, $\varepsilon = 0.1$, $C = 1$



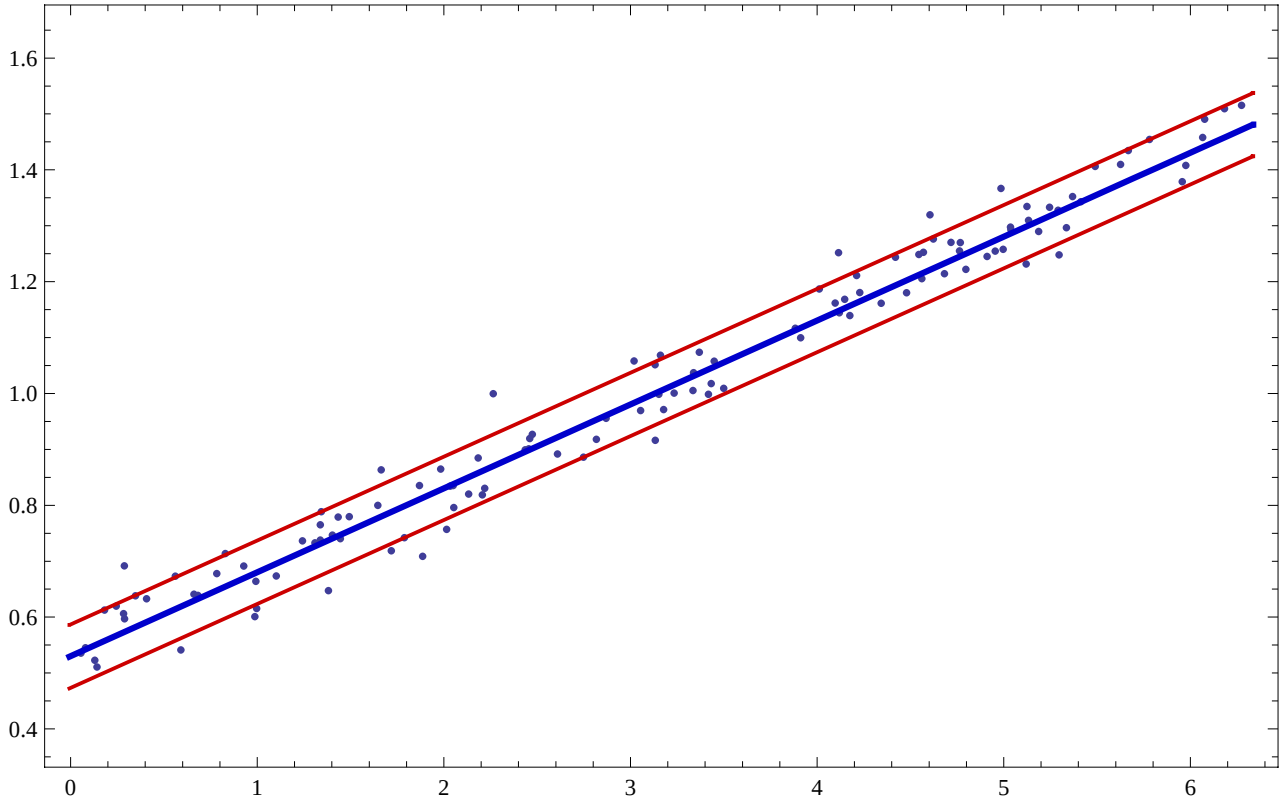
LINEAR SVR EXAMPLE:

ε-SVR, $\varepsilon = 0.01$, $C = 1$



LINEAR SVR EXAMPLE:

ν-SVR, $\nu = 0.2$, $C = 100 \rightarrow \varepsilon = 0.056633$



NONLINEAR SUPPORT VECTOR REGRESSION: INTRODUCTION

- It is clear that the usefulness of linear support vector regression is rather limited.
- Just like for classification, the generalization to a non-linear setting is done by using a non-linear kernel and considering the dual problem only.

ε-SVR: THE DUAL PROBLEM

Maximize

$$\mathcal{L}(\boldsymbol{\alpha}^+, \boldsymbol{\alpha}^-) = -\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i^+ - \alpha_i^-)(\alpha_j^+ - \alpha_j^-)k(\mathbf{x}^i, \mathbf{x}^j) - \varepsilon\sum(\alpha_i^+ + \alpha_i^-) + \sum y^i(\alpha_i^+ - \alpha_i^-)$$

with respect to $\boldsymbol{\alpha}^+$ and $\boldsymbol{\alpha}^-$ subject to the constraints $0 \leq \alpha_i^+ \leq C$, $0 \leq \alpha_i^- \leq C$ ($i = 1, \ldots, l$), and $\sum_{i=1}^{l}(\alpha_i^+ - \alpha_i^-) = 0$.

ε-SVR: THE DUAL PROBLEM (cont'd)

With the above notations and the convention $\mathbf{K} = \left(k(\mathbf{x}^i, \mathbf{x}^j)\right)_{i=1,\ldots,l}^{j=1,\ldots,l}$, we can rewrite the dual problem as follows:

Minimize

$$\frac{1}{2}(\boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-)^T \mathbf{K}(\boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-) + \varepsilon \mathbf{1}^T(\boldsymbol{\alpha}^+ + \boldsymbol{\alpha}^-) - \mathbf{y}^T(\boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-)$$

with respect to $\boldsymbol{\alpha}^+$ and $\boldsymbol{\alpha}^-$ subject to the constraints $\mathbf{0} \leq \boldsymbol{\alpha}^+ \leq C\mathbf{1}$, $\mathbf{0} \leq \boldsymbol{\alpha}^+ \leq C\mathbf{1}$, and $\mathbf{1}^T(\boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-) = 0$.

This is again a convex quadratic optimization problem with linear constraints.

ε-SVR: THE FINAL REGRESSION FUNCTION

Once the dual problem has been solved, the final regression function is given as

$$g(\mathbf{x}) = \sum_{i=1}^{l} (\alpha_i^+ - \alpha_i^-) k(\mathbf{x}^i, \mathbf{x}) + b.$$

For any α_j^+ such that $0 < \alpha_j^+ < C$, we can infer $\xi_j^+ = 0$ from the Karush-Kuhn-Tucker conditions and compute b as follows:

$$b = y^j - \sum_{i=1}^{l} (\alpha_i^+ - \alpha_i^-) k(\mathbf{x}^i, \mathbf{x}^j) - \varepsilon.$$

ε-SVR: SUPPORT VECTORS

Again we can infer the following from the Karush-Kuhn-Tucker conditions:

- For $\varepsilon > 0$, $\alpha_i^+ \alpha_i^- = 0$ holds.
- If $\alpha_i^+ = 0$ and $\alpha_i^- = 0$, the i-th sample is inside the ε-tube and does not contribute to $g(\mathbf{x})$.
- If $\alpha_i^+ > 0$ or $\alpha_i^- > 0$, the i-th sample is a *support vector*.
- If $0 < \alpha_i^+ < C$, $(\mathbf{x}^i, y^i)$ is on the upper border of the ε-tube. If $0 < \alpha_i^- < C$, $(\mathbf{x}^i, y^i)$ is on the lower border of the ε-tube.
- If $\alpha_i^+ = C$ or $\alpha_i^- = C$ holds, $(\mathbf{x}^i, y^i)$ is outside the ε-tube around $g(\mathbf{x})$.

ν-SVR: THE DUAL PROBLEM

Maximize

$$\mathcal{L}(\boldsymbol{\alpha}^+, \boldsymbol{\alpha}^-) = -\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i^+ - \alpha_i^-)(\alpha_j^+ - \alpha_j^-)k(\mathbf{x}^i, \mathbf{x}^j) + \sum y^i(\alpha_i^+ - \alpha_i^-)$$

with respect to $\boldsymbol{\alpha}^+$ and $\boldsymbol{\alpha}^-$ subject to the constraints $0 \leq \alpha_i^+ \leq \frac{C}{l}$, $0 \leq \alpha_i^- \leq \frac{C}{l}$ $(i = 1, \ldots, l)$, $\sum_{i=1}^{l}(\alpha_i^+ - \alpha_i^-) = 0$ and $\sum_{i=1}^{l}(\alpha_i^+ + \alpha_i^-) \leq C\nu$.

ν-SVR: THE DUAL PROBLEM (cont'd)

With the notations from above, we can rewrite the dual problem as follows:

Minimize

$$\frac{1}{2}(\boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-)^T \mathbf{K}(\boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-) - \mathbf{y}^T(\boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-)$$

with respect to $\boldsymbol{\alpha}^+$ and $\boldsymbol{\alpha}^-$ subject to the constraints $\mathbf{0} \leq \boldsymbol{\alpha}^+ \leq \frac{C}{l}\mathbf{1}$, $\mathbf{0} \leq \boldsymbol{\alpha}^+ \leq \frac{C}{l}\mathbf{1}$, $\mathbf{1}^T(\boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-) = 0$, and $\mathbf{1}^T(\boldsymbol{\alpha}^+ + \boldsymbol{\alpha}^-) \leq C\nu$.

This is again a convex quadratic optimization problem with linear constraints.

ν-SVR: THE FINAL REGRESSION FUNCTION

Once the dual problem has been solved, the final regression function is again given as

$$g(\mathbf{x}) = \sum_{i=1}^{l} (\alpha_i^+ - \alpha_i^-) k(\mathbf{x}^i, \mathbf{x}) + b.$$

To compute b and ε, choose an α_p^+ such that $0 < \alpha_p^+ < \frac{C}{l}$ and an α_q^- such that $0 < \alpha_q^- < \frac{C}{l}$. Then the solutions are given as follows:

$$b = \frac{1}{2}\Big(\big(y^p - \sum_{i=1}^{l} (\alpha_i^+ - \alpha_i^-) k(\mathbf{x}^i, \mathbf{x}^p)\big) + \big(y^q - \sum_{i=1}^{l} (\alpha_i^+ - \alpha_i^-) k(\mathbf{x}^i, \mathbf{x}^q)\big) \Big)$$

$$\varepsilon = \frac{1}{2}\Big(\big(y^p - \sum_{i=1}^{l} (\alpha_i^+ - \alpha_i^-) k(\mathbf{x}^i, \mathbf{x}^p)\big) - \big(y^q - \sum_{i=1}^{l} (\alpha_i^+ - \alpha_i^-) k(\mathbf{x}^i, \mathbf{x}^q)\big) \Big)$$

Averaging over several such pairs is again possible, of course.

ν-SVR: INTERPRETING THE PARAMETER ν

Theorem. Assume that we are given a ν-SVR solution such that $\varepsilon > 0$. Then the following holds:

1. ν is an upper bound for the proportion of errors (training samples outside the ε-tube).
2. ν is a lower bound for the proportion of support vectors.

Under some technical assumptions, it is possible to show that both the proportion of errors and the proportion of support vectors tend to ν as l goes to infinity.

Note that $\varepsilon > 0$ is only possible if $\nu \leq 1$. If $\nu > 1$, $\varepsilon = 0$ follows.

CONNECTION ε-SVR – ν-SVR

Theorem. Assume that we are given a ν-SVR solution according to some data set $\mathbf{Z}_l$ and a cost factor C such that $\varepsilon > 0$ holds. Then exactly the same regression function would have been obtained if we had trained an ε-SVR with the same ε and $C' = \frac{C}{l}$.

This result says that ν-SVR is basically nothing else but an ε-SVR which automatically finds a good choice for the error threshold ε.

SUPPORT VECTOR REGRESSION: FURTHER NOTES

- We can interpret support vector regression as a linear combination of basis functions (plus a constant term b)

$$g(\mathbf{x}) = \sum_{i=1}^{l} \mu_i g_i(\mathbf{x}) + b,$$

  where $g_i(\mathbf{x}) = k(\mathbf{x}^i, \mathbf{x})$ and $\mu_i = \alpha_i^+ - \alpha_i^-$.
- Traditional nonlinear regression is usually concerned with optimizing the factors μ_i such that the regression functions fits the data best. Support vector regression, instead, tries to adjust the factors μ_i such that the data fit into the ε-tube around the regression function. The parameter C controls how large the factors μ_i may get to achieve this goal.

SUPPORT VECTOR REGRESSION: FURTHER NOTES (cont'd)

- In case that we have an upper bound D for the norm of the derivative

$$\| \frac{\partial k}{\partial \mathbf{x}} (\mathbf{y}, \mathbf{x}) \|,$$

we can directly infer that the magnitude of the Lagrange multipliers are connected to the norm of the derivative of the regression function:

$$\| \frac{\partial g}{\partial \mathbf{x}} (\mathbf{x}) \| \leq \sum_{i=1}^{l} |\alpha_i^+ - \alpha_i^-| \cdot \| \frac{\partial k}{\partial \mathbf{x}} (\mathbf{y}, \mathbf{x}) \| \leq \begin{cases} ClD & \text{for } \varepsilon\text{-SVR} \\ CD & \text{for } \nu\text{-SVR} \end{cases}$$

This means that the cost factor C directly limits the derivative of the final regression function.

- For the RBF kernel, for instance, we have such an upper bound: $D = 1/(\sigma\sqrt{e})$

SUPPORT VECTOR REGRESSION: COMPLEXITY-RELATED ISSUES

Similar to support vector classification, the choices of the parameters are crucial for the final outcome:

- If the RBF kernel is used, the choice of σ is crucial. too large $\sigma \rightarrow$ underfitting; too small $\sigma \rightarrow$ overfitting (see previous slide!).
- The choice of C also influences complexity. The higher C, the more we punish errors, hence, the higher the tendency of the SVR to produce a more complex regression function. Analogously for ν.

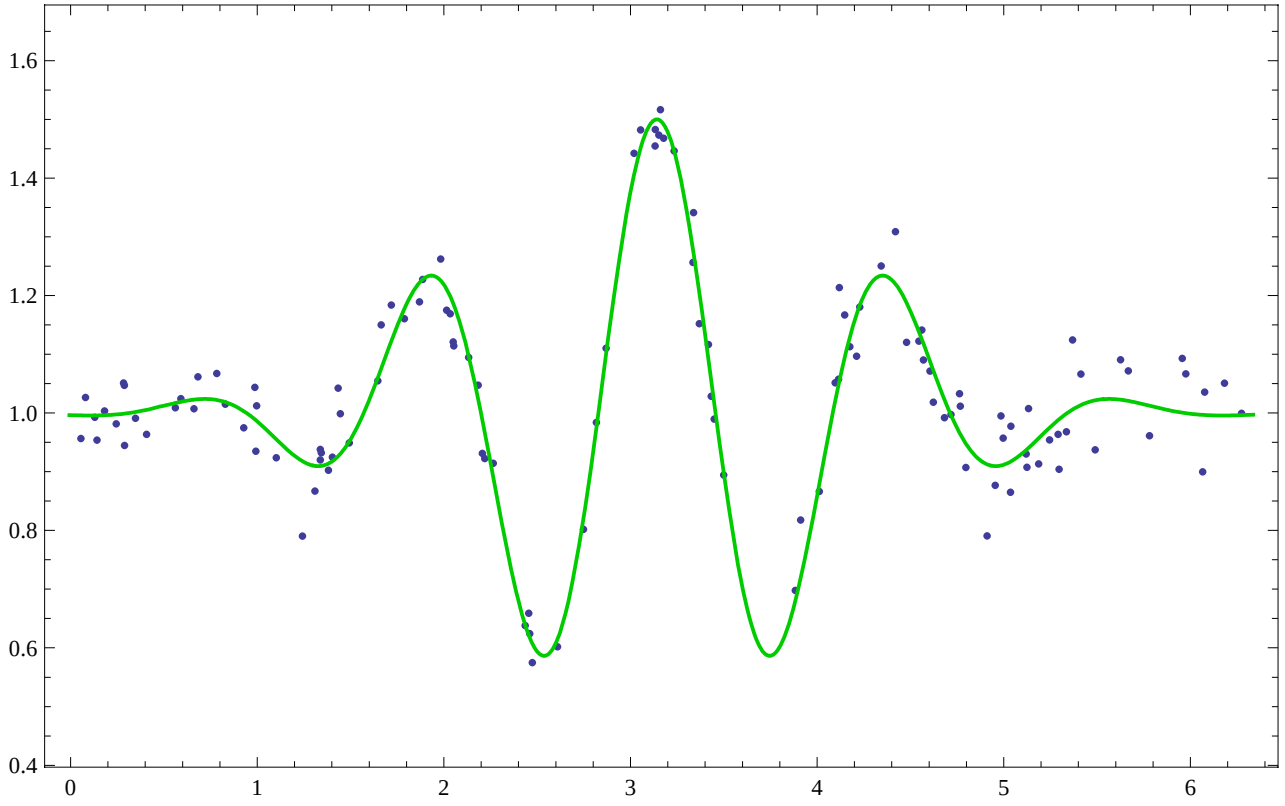
It is often unavoidable to use cross validation to find good choices for hyperparameters.

SVR: COMPLEXITY-RELATED ISSUES (cont'd)

- We have not introduced a theoretical concept of complexity of real-valued functions.
- It seems intuitively reasonable that complexity relates both to the number of support vectors and the magnitude of the Lagrange multipliers.
- For the RBF kernel, this is obvious:
  1. The more support vectors, the more local minima/maxima.
  2. The larger the Lagrange multipliers, the steeper the regression function may be.

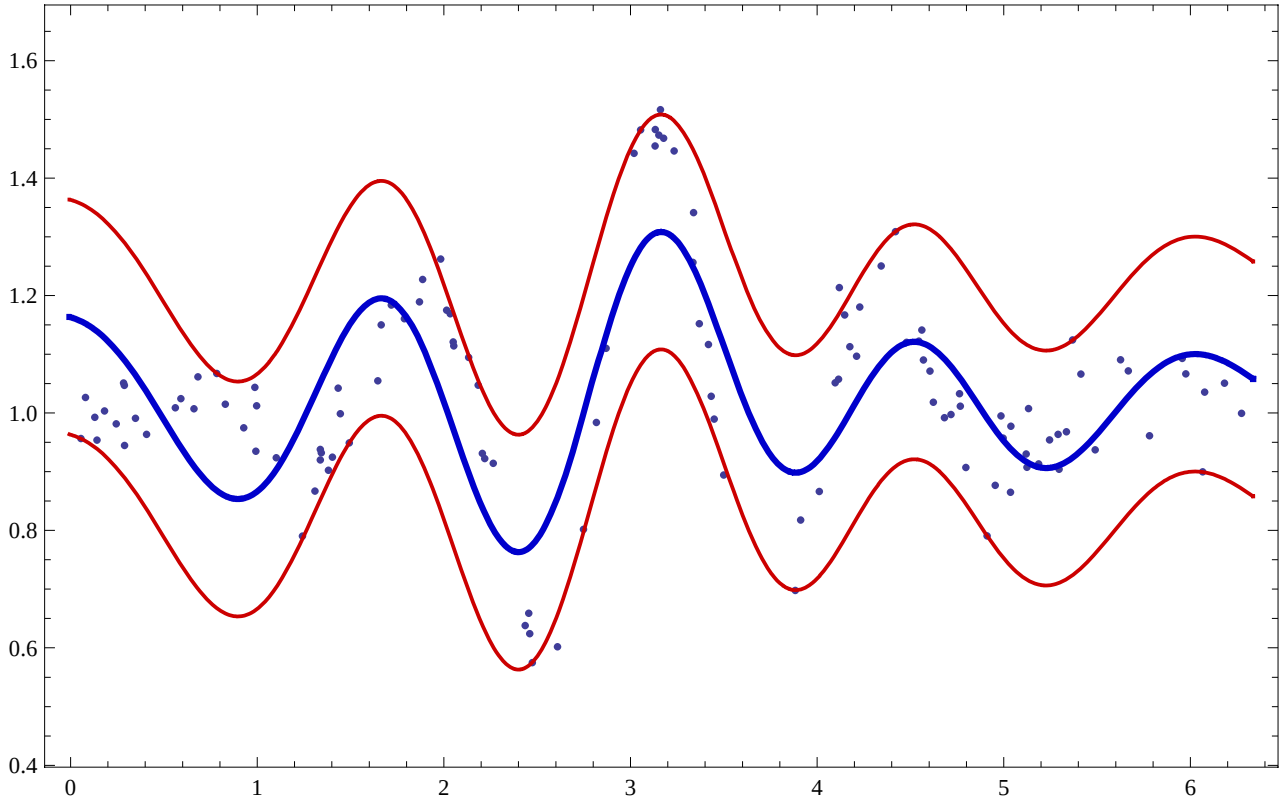
NONLINEAR SVR EXAMPLE:

$f(x) = 1 + \frac{1}{2}\cos(5(x-\pi)) \cdot \exp(-\frac{1}{2}(x-\pi)^2)$ **PLUS NOISE**



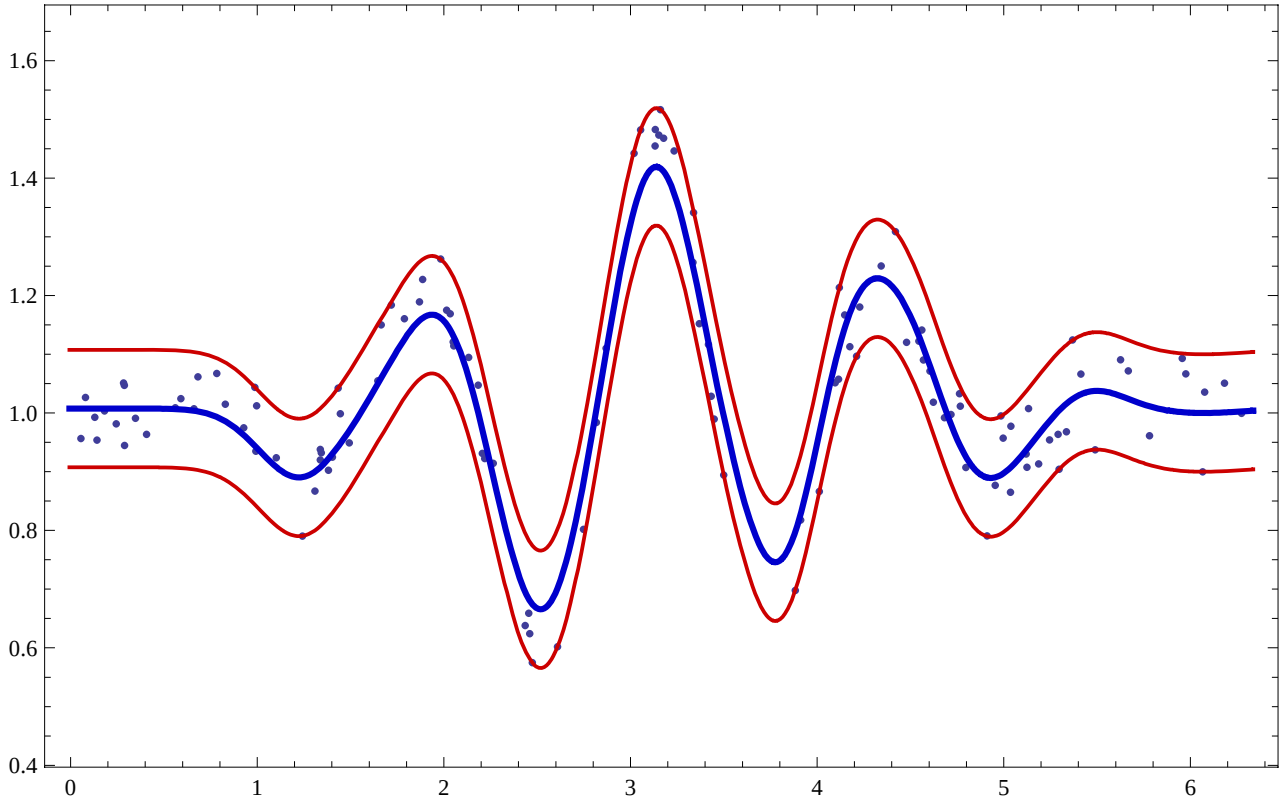
NONLINEAR SVR EXAMPLE:

ε-SVR, $\varepsilon = 0.2$**,** $C = 10$**, RBF KERNEL,** $\frac{1}{2\sigma^2} = 1$



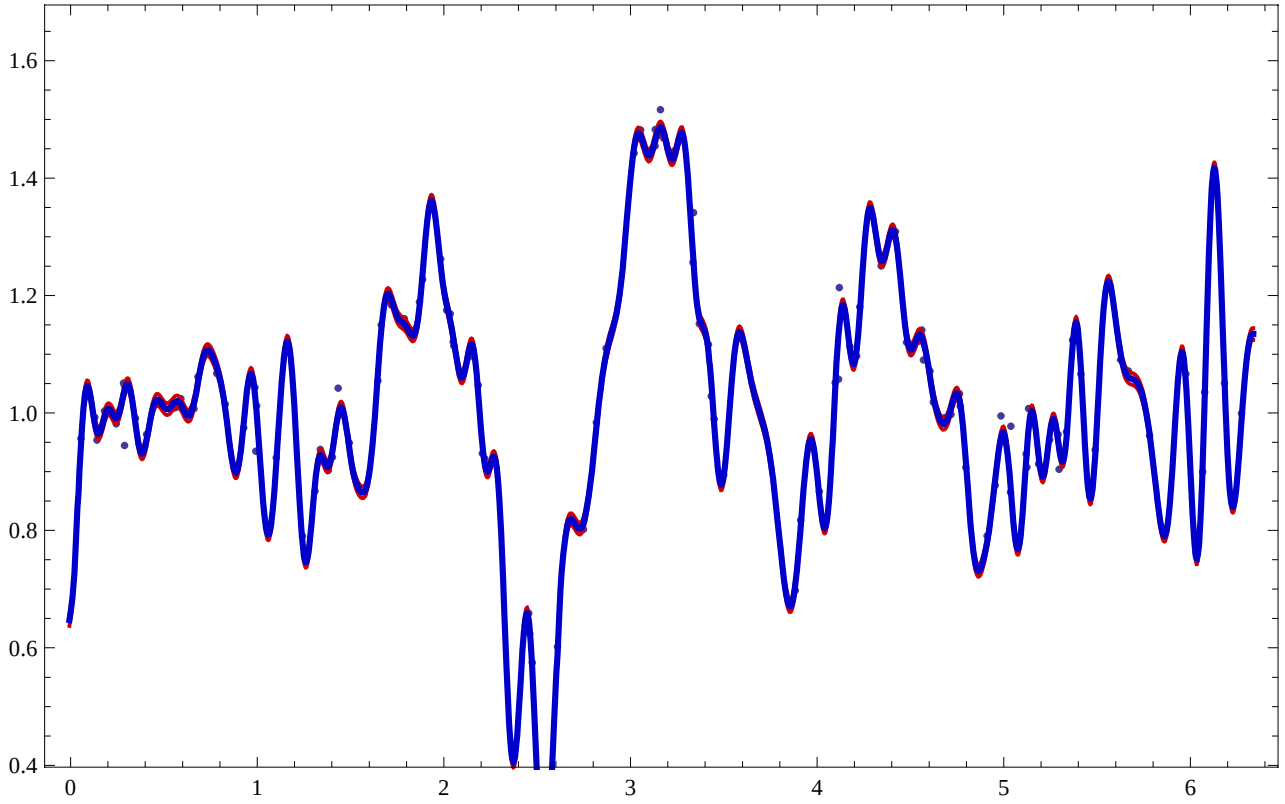
NONLINEAR SVR EXAMPLE:

ε-SVR, $\varepsilon = 0.1$**,** $C = 10$**, RBF KERNEL,** $\frac{1}{2\sigma^2} = 10$



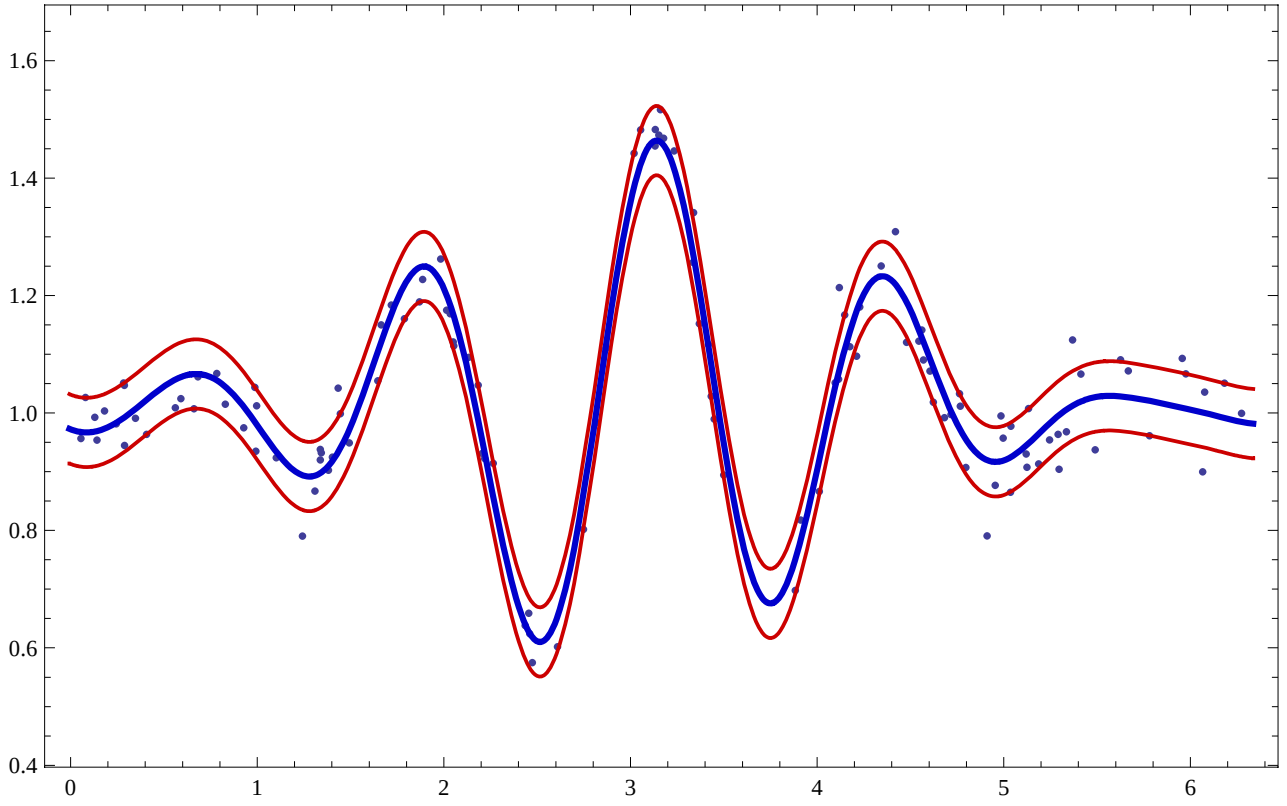
NONLINEAR SVR EXAMPLE:

ε-SVR, $\varepsilon = 0.01$**,** $C = 100$**, RBF KERNEL,** $\frac{1}{2\sigma^2} = 100$



NONLINEAR SVR EXAMPLE:

ν-SVR, $\nu = 0.2$**,** $C = 1000$**, RBF KERNEL,** $\frac{1}{2\sigma^2} = 1 \rightarrow \varepsilon = 0.058992$



SOFTWARE: LIBSVM

- Free software available from

  http://www.csie.ntu.edu.tw/~cjlin/libsvm/

  as source code; Windows and Linux binaries are also available.
- Has already become a kind of standard.
- Basically consists of two command line tools, one for training an SVM, the second for testing it on new data.
- Implements all four SVMs discussed here plus the unsupervised one-class SVM; multi-class classification is implemented via pairwise classifiers.

SOFTWARE: LIBSVM (cont'd)

- Implements the four kernels discussed here; additionally, arbitrary kernels can be used by supplying the whole pre-computed kernel matrix $\mathbf{K}$.
- Optimization (uses sequential minimal optimization) is extremely fast and robust.
- Little drawback: discriminant values are not directly accessible; to compute ROC curves (or anything similar), the source code must be modified.
- A lot of tools and interfaces are available (Matlab, Perl, Python, R via the `e1071` package).

SOFTWARE: SVM*light*

- Software available from http://svmlight.joachims.org/ as source code; Windows and Linux binaries are also available. Free for academic users.
- Works similarly to libSVM (two command line tools); also the input file format is the same.
- Implements the same kernels and allows to use pre-computed kernel matrices.
- Implements C-SVM, ε-SVR, and preference ranking. Does not support multi-class classification (a multi-class variant is available which, however, only supports linear classification).
- Model evaluation and optimization can be adjusted more flexibly than for libSVM.

SOFTWARE: R PACKAGE `kernlab`

- R package available via CRAN
- Implements many different SVMs and other kernel methods.
- Implements many different kernels and allows for seamless integration of user-written custom kernels.

FURTHER TOPICS THAT WOULD HAVE BEEN WORTH A LOOK...

... if there had been more time:

- **One-class SVM:** unsupervised SVM useful for novelty detection, data filtering, etc.
- **P-SVM:** scale-invariant SVM that is able to work with dyadic data and "kernel matrices" that are not positive semi-definite; it is also useful for feature selection.
- SVM optimization, in particular, Sequential Minimal Optimization (SMO).

CONCLUDING REMARKS

- Support vector machines are easy-to-use machine learning workhorses that have become part of the standard repertoire of machine learning methods.
- SVMs have won numerous machine learning competitions.
- They are built on a solid theoretical foundation.
- Both training and testing are deterministic and fast (further note that solving the optimization problem gives a global solution which is not true for most other machine learning algorithms).

CONCLUDING REMARKS (cont'd)

- SVMs can be used for any problem for which it is possible to define a positive semi-definite comparison measure (the kernel), including, strings/sequences, signals, images, trees, etc.
- Although SVMs are motivated by simultaneously minimizing complexity, the choice of hyperparameters remains crucial; often cross validation remains the only remedy.