UNIT 3

Support Vector Machines

Johannes Kofler

JOHANNES KEPLER UNIVERSITY LINZ

Institute for Machine Learning

Copyright statement:

Lecture Supervised Techniques: Planned Topics

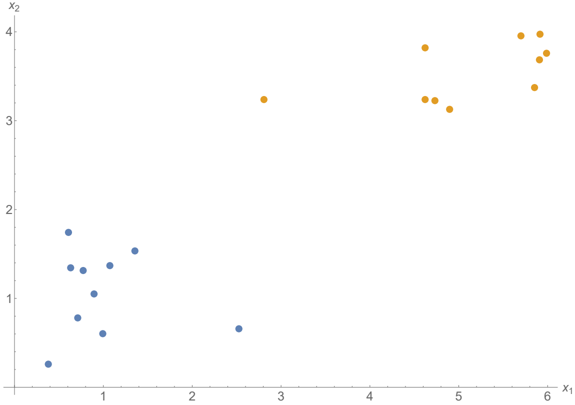
- UNIT 1: Overview of Supervised Machine Learning
- UNIT 2: Basics of Supervised Machine Learning
- UNIT 3: Support Vector Machines
- UNIT 4: Random Forests and Gradient Boosting
- UNIT 5: Logistic Regression
- UNIT 6: Artificial Neural Networks
- UNIT 7: Special Network Architectures

Planned topics for Unit 3

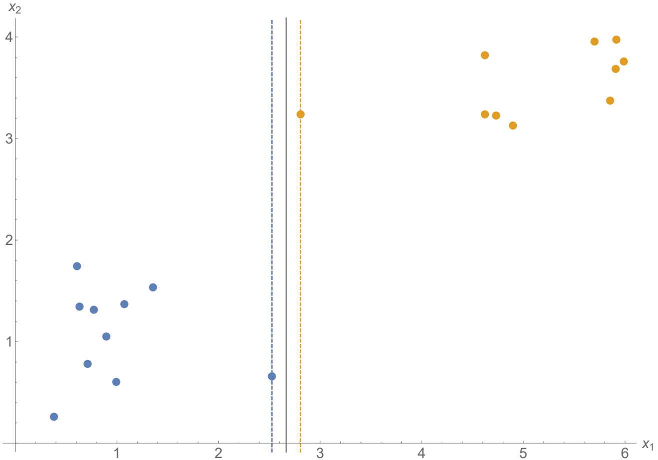
- Linear SVMs
  - Basics of convex optimization
  - Rigorous derivation for linearly separable data
  - No linear separability: basic concepts and ideas for C-SVMs
- Nonlinear SVMs
  - Kernel trick
  - Discussion of corresponding dual problem
  - How to find the right kernels?
- Multi-class SVMs
- Support vector regression (SVR): linear and nonlinear
- Pros and Cons of SVMs in general

Support Vector Machines: Basic idea (1)

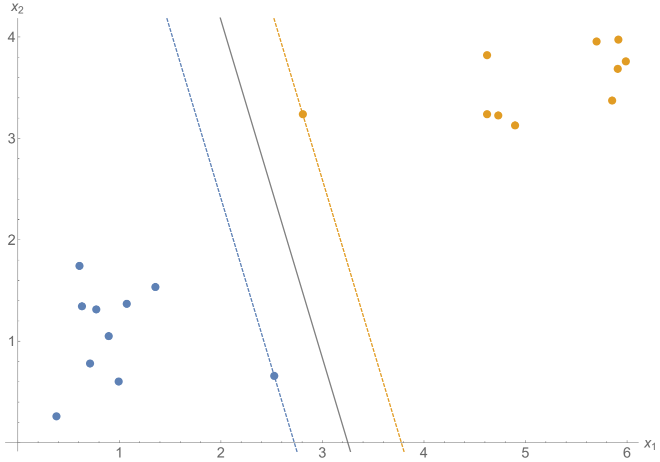
- Support Vector Machines (SVMs) are based on the idea of finding a linear classification border maximizing the margin between negative (blue) and positive (orange) samples.
- We illustrate the situation in the following pictures:



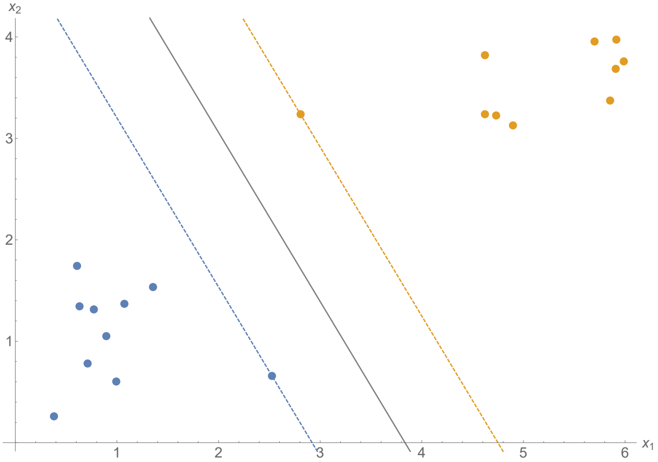
Support Vector Machines: Basic idea (2)



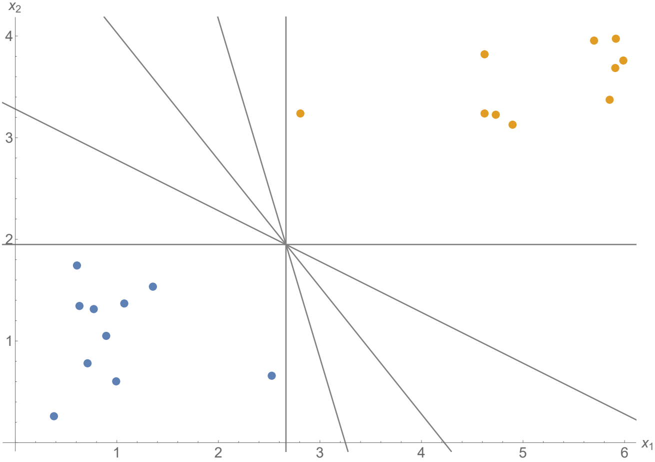
Support Vector Machines: Basic idea (3)



Support Vector Machines: Basic idea (4)



Support Vector Machines: Basic idea (5)



Main steps of historical development

- 1963: Vapnik and Chervonenkis: original idea of SVMs
- 1992: Boser, Guyon and Vapnik: suggested way to create nonlinear classifiers by using kernels
- Further important contributions to kernel methods and applications to SVMs by Schölkopf and Smola

SVMs for linear separability: The formal setup: Part 1

- Usual situation: 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 $y_i \in \{-1, 1\}$
- Recall: Hesse normal form: closest "distance" (including sign) of a point $\mathbf{x}$ to the separating hyperplane (given by $\mathbf{w}$ and b) is $\frac{\mathbf{w}\cdot\mathbf{x}-b}{\|\mathbf{w}\|}$.
- Assume that positive and negative samples are linearly separable, i.e. there exist $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$ such that:

$$\mathrm{sign}(\mathbf{w} \cdot \mathbf{x}_i - b) = y_i \quad \text{for all } i = 1, \ldots, l$$

- Criterion for linear separability: Two sets of points are linearly separable $\Leftrightarrow$ their convex hulls are disjoint.
- The hyperplane separating positive and negative samples is given as $\mathbf{w} \cdot \mathbf{x} - b = 0$.

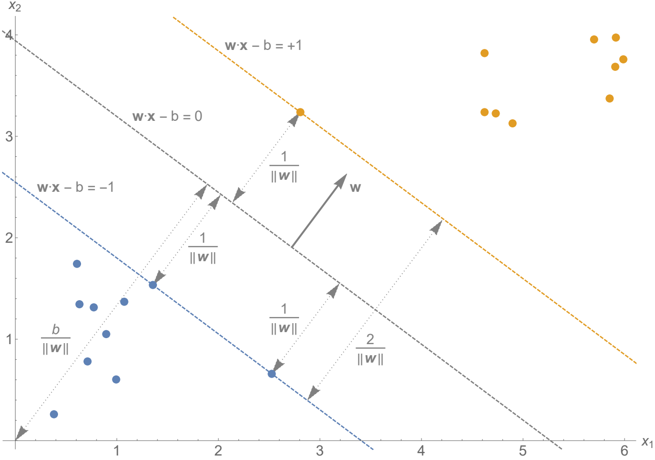
SVMs for linear separability: The formal setup: Part 2

- Additionally: assume that this hyperplane fulfills

$$\min_{i=1,\dots,l} |\mathbf{w} \cdot \mathbf{x}_i - b| = 1,$$

  and say that it is in canonical form (with respect to $\mathbf{Z}$). Can always be achieved by reparametrization (change of length of $\mathbf{w}$).
- Hence, if $\mathbf{w} \cdot \mathbf{x} - b$ is in canonical form: distance of separating hyperplane to closest data point(s) is $\frac{1}{\|\mathbf{w}\|}$.
- Main rationale: the farther a separating hyperplane is away from the data, the less likely it is to produce a misclassification.
- Objective: We look for separating hyperplane whose minimal distance to all training samples is maximal. Equivalent: maximize $\frac{2}{\|\mathbf{w}\|} \rightarrow$ margin maximization
- This intuition can be made precise using tools from statistical learning theory, however, we won't pursue this further here.

SVMs for linear separability: The formal setup: Part 3



SVMs for linear separability: The formal setup: Part 4

- We want to prevent data from falling into margins. For every sample $(\mathbf{x}_i, y_i)$, we require:

$$\text{if } \; y_i = +1 : \quad \mathbf{w} \cdot \mathbf{x}_i - b \geq 1$$
$$\text{if } \; y_i = -1 : \quad \mathbf{w} \cdot \mathbf{x}_i - b \leq -1$$

- Compact notation:

$$y_i \left(\mathbf{w} \cdot \mathbf{x}_i - b \right) - 1 \geq 0$$

SVMs for linear separability: The formal setup: Part 5: Abstract problem formulation

- Original Problem: For given linearly separable data set $\mathbf{Z}$, $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$:

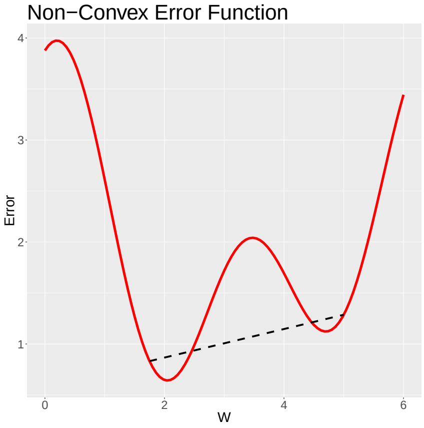
$$\begin{array}{lll} \text{Maximize} & \dfrac{2}{\|\mathbf{w}\|} & \\ \text{subject to} & y_i(\mathbf{w} \cdot \mathbf{x}_i - b) - 1 \geq 0 & \text{for } i = 1, \ldots, l. \end{array}$$

- Equivalent: Primal Problem:

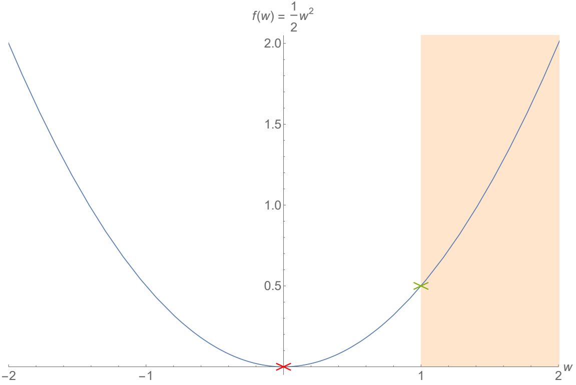
$$\begin{array}{lll} \text{Minimize} & \dfrac{1}{2}\|\mathbf{w}\|^2 = \dfrac{1}{2}\displaystyle\sum_{i=1}^{d} w_i^2 & \\ \text{subject to} & -(y_i(\mathbf{w} \cdot \mathbf{x}_i - b) - 1) \leq 0 & \text{for } i = 1, \ldots, l. \end{array}$$

- $\rightarrow$ Convex quadratic optimization problem with linear constraints. Next: recall basic mathematical tools.

Interlude: constrained convex optimization: Part 1



Interlude: constrained convex optimization: Part 2



- Red: Global minimum
- Green: Constrained minimum under constraint $w > 1$

Interlude: constrained convex optimization: Part 3

- Assumptions: we have functions f and h_i ($i = 1, \ldots, l$) from $\mathbb{R}^d$ to $\mathbb{R}$ with following requirements:
  1. Convex
  2. Twice continuously differentiable
  3. Slater condition: there exists $\mathbf{w}'$ with $h_i(\mathbf{w}') < 0$ for all $i = 1, \ldots, l$.
- Primal Problem:

$$\begin{array}{rll} \text{Minimize} & f(\mathbf{w}) & \\ \text{subject to} & h_i(\mathbf{w}) \leq 0 & \text{for } i = 1, \ldots, l. \end{array}$$

- For simplicity: we don't deal with equality constraints here
- Next slide: strategy how to solve this problem

Interlude: constrained convex optimization: Part 4

- Lagrange function:

$$L(\mathbf{w};\alpha_1,\ldots,\alpha_l) = f(\mathbf{w}) + \sum_{i=1}^{l} \alpha_i h_i(\mathbf{w})$$

$\alpha_1,\ldots,\alpha_l$: Lagrange multipliers.

- Dual Problem:

Maximize $\mathcal{L}(\alpha_1,\ldots,\alpha_l) = \inf_{\mathbf{w}} L(\mathbf{w};\alpha_1,\ldots,\alpha_l)$ wrt. $\alpha_1,\cdots,\alpha_l$

subject to $\alpha_i \geq 0$ for $i = 1,\ldots,l$.

- Karush-Kuhn-Tucker (KKT): $\mathbf{w}^*$ solves primal problem $\Leftrightarrow$ there exist non-negative Lagrange multipliers with:
  1. $\mathcal{L}(\alpha_1,\ldots,\alpha_l) = L(\mathbf{w}^*;\alpha_1,\ldots,\alpha_l)$
  2. $\alpha_1, ..., \alpha_l$ solve dual problem, i.e. they maximize $\mathcal{L}$.
  3. $\alpha_i h_i(\mathbf{w}^*) = 0$ for all $i = 1,\ldots,l$.
- For details and proofs (slightly different notation): have a look at section 6.3. of the following classic (and references therein)

Interlude: constrained convex optimization: Part 5

- Illustrative example (toy problem):

$$\begin{aligned} \text{Minimize} \quad & f(w_1, w_2) = (w_1 - 2)^2 + w_2^2 \\ \text{subject to} \quad & h_1(w_1, w_2) = -w_1 \leq 0 \\ \text{and} \quad & h_2(w_1, w_2) = w_1 - w_2 \leq 0. \end{aligned}$$

- Lagrange function:

$$L(w_1, w_2, \alpha_1, \alpha_2) = (w_1 - 2)^2 + w_2^2 - \alpha_1 \, w_1 + \alpha_2 \, (w_1 - w_2)$$

- To find $\inf_{(w_1, w_2)} L(w_1, w_2, \alpha_1, \alpha_2)$ for given (α_1, α_2), consider:

$$\frac{\partial L}{\partial w_1}(w_1, w_2, \alpha_1, \alpha_2) = 2\, w_1 - 4 - \alpha_1 + \alpha_2$$

$$\frac{\partial L}{\partial w_2}(w_1, w_2, \alpha_1, \alpha_2) = 2\, w_2 - \alpha_2$$

Interlude: constrained convex optimization: Part 6

- Setting the two derivatives to 0, we obtain $w_1^* = 2 + \frac{\alpha_1 - \alpha_2}{2}$ and $w_2^* = \frac{\alpha_2}{2}$. (This must be a minimum as L is convex and no maximum exists.)
- Furthermore:

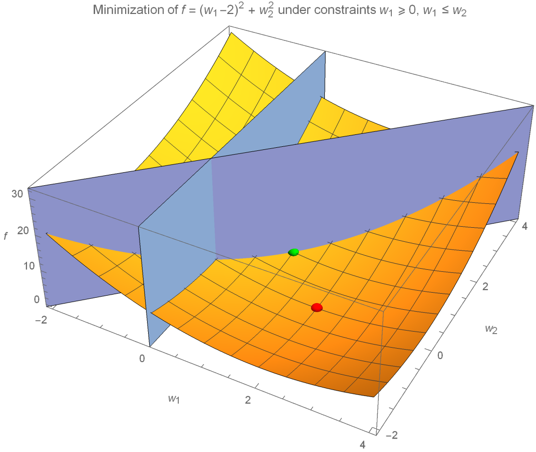
$$\begin{aligned}\mathcal{L}(\alpha_1, \alpha_2) &= L\Big(w_1^*, w_2^*, \alpha_1, \alpha_2\Big) \\ &= -\frac{1}{4}\left(\alpha_1^2 + 8\alpha_1 - 2\alpha_1\alpha_2 - 8\alpha_2 + 2\alpha_2^2\right)\end{aligned}$$

- Dual problem: maximize $\mathcal{L}(\alpha_1, \alpha_2)$ subject to $\alpha_1 \geq 0$ and $\alpha_2 \geq 0$. Yields $\alpha_1 = 0$ and $\alpha_2 = 2$ and thus $w_1^* = 1$ and $w_2^* = 1$. Can be deduced by KKT-conditions.
- In many practical situations: algorithms for solving dual problem exist, especially for SVMs (as we will see later). Here: argument for toy example using KKT conditions.

Interlude: constrained convex optimization: Part 7

- By the first KKT condition: $\alpha_1 h_1(w_1^*, w_2^*) = 0$.
- Assume $h_1(w_1^*, w_2^*) < 0$. Then $\alpha_1 = 0$. Now look at the second KKT-condition $\alpha_2 h_2(w_1^*, w_2^*) = 0$.
  - If $\alpha_2 = 0$: $h_2(w_1^*, w_2^*) = 2 > 0$ which contradicts constraint on h_2 $\rightarrow$ can be ruled out.
  - If $h_2(w_1^*, w_2^*) = 0$ we can deduce $w_1^* = w_2^*$ and thus $\alpha_2 = 2$. (recall: $\alpha_1 = 0$). All the constraints are satisfied, thus we are done, since we have unique solution because of convexity. For sake of completeness, we will also provide arguments ruling out other possibilities:
- Assume $h_1(w_1^*, w_2^*) = 0$. Then $\alpha_2 = 4 + \alpha_1$.
  - If $\alpha_2 = 0$ then $\alpha_1 = -4 \rightarrow$ can be ruled out as $\alpha_i \geq 0$.
  - If $\alpha_2 \neq 0$ then $h_2(w_1^*, w_2^*) = 0$, which implies $\alpha_2 = \frac{\alpha_1}{2} + 2$, i.e. $4 + \alpha_1 = \frac{\alpha_1}{2} + 2$, i.e. $\alpha_1 = -4 \rightarrow$ can be ruled out again.
- Solution: $\alpha_1 = 0, \alpha_2 = 2$, yielding: $w_1^* = 1, w_2^* = 1$ and $f(w_1^*, w_2^*) = 2$.

Interlude: constrained convex optimization: Part 8



- Red: global minimum (not obeying the constraints)
- Green: solution for minimization under constraints

Back to linear SVMs: Part 1

- Associated Lagrange function is given as

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

$$= \frac{1}{2}\left\|\mathbf{w}\right\|^2 - \mathbf{w} \cdot \sum_{i=1}^{l} \alpha_i\, y_i\, \mathbf{x}_i + b \sum_{i=1}^{l} \alpha_i\, y_i + \sum_{i=1}^{l} \alpha_i$$

- Solving the dual problem enforces the conditions

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

- This implies:

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

- Furthermore:

$$\mathcal{L}(\alpha_1, \ldots, \alpha_l) = \sum_{i=1}^{l} \alpha_i - \tfrac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i\, \alpha_j\, y_i\, y_j\, \mathbf{x}_i \cdot \mathbf{x}_j .$$

- Final step: maximize $\mathcal{L}$ with respect to α_i subject to $\alpha_i \geq 0$ (for all $i = 1, \ldots, l$) and $\sum_{i=1}^{l} \alpha_i\, y_i = 0$.

Back to linear SVMs: Part 2

- Introduce

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

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

- The dual problem can be written as follows:

$$\text{Maximize} \qquad \mathcal{L} = -\frac{1}{2}\boldsymbol{\alpha}^T \mathbf{Q} \boldsymbol{\alpha} + \mathbf{1}^T \boldsymbol{\alpha} \;\; \text{wrt. } \boldsymbol{\alpha}$$

$$\text{subject to} \qquad \boldsymbol{\alpha} \geq \mathbf{0} \text{ and } \boldsymbol{\alpha}^T \mathbf{y} = 0.$$

- Easy observation: $\mathbf{Q}$ is positive semi-definite.
- $\rightarrow$ convex quadratic optimization problem with linear constraints: only global minima: uniqueness if positive definite.

Back to linear SVMs: Part 3

- Once we solved the dual problem $\rightarrow$ obtain $\alpha_1, \ldots, \alpha_l$ which also solve primal problem. By the KKT-conditions:
$$\alpha_i\big(y_i(\mathbf{w}\cdot\mathbf{x}_i - b) - 1\big) = 0 \quad \text{for} \quad i = 1, \ldots, l.$$
- Thus, for $i = 1, \ldots, l$:
  - either $\alpha_i = 0$
  - or $y_i(\mathbf{w}\cdot\mathbf{x}_i - b) - 1 = 0$
  - or both.
- Samples $\mathbf{x}_i$ for which $\alpha_i > 0$ holds (i.e. $y_i(\mathbf{w}\cdot\mathbf{x}_i - b) - 1 = 0$, i.e. $\mathbf{w}\cdot\mathbf{x}_i - b = \pm 1$) lie on the margin border and are called support vectors (encircled in red in next-but-one slide).

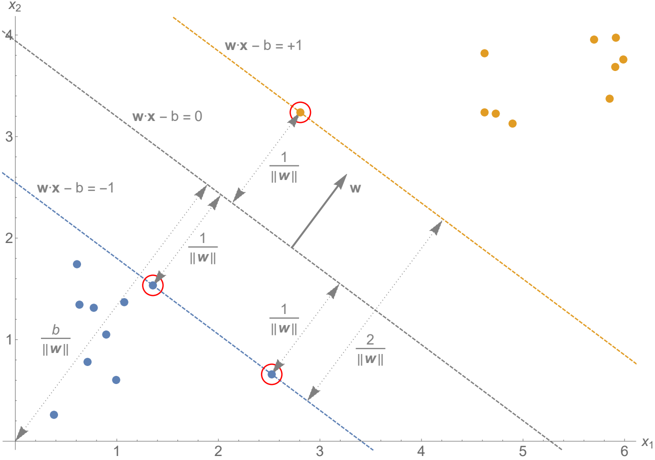
Back to linear SVMs: Part 4

- For arbitrary support vector $\mathbf{x}_j$ (with $\alpha_j > 0$), the KKT condition implies $y_j(\mathbf{w} \cdot \mathbf{x}_j - b) = 1$, and thus

$$b = -y_j + \mathbf{w} \cdot \mathbf{x}_j = -y_j + \sum_{i=1}^{l} \alpha_i \, y_i \, \mathbf{x}_i \cdot \mathbf{x}_j$$

- Recommended: don't base the computation of b on only one support vector (for reasons of numerical precision), but compute a b value for each support vector and use their average.
- Under specific conditions: it may be useful to adjust b according to some other quality measure after training.

Back to linear SVMs: Part 5



Back to linear SVMs: Part 6

- Given Lagrange multipliers $\alpha_1, \ldots, \alpha_l$ solving the primal problem, we can construct the (average) solution for b and the solution for $\mathbf{w}$:

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

- $\rightarrow$ Final classification function (for new input $\mathbf{x}$), i.e. the linear Support Vector Machine (SVM), is given as

$$g(\mathbf{x}) = \operatorname{sign}(\mathbf{w} \cdot \mathbf{x} - b) = \operatorname{sign}\Bigg(\underbrace{\sum_{i=1}^{l} \alpha_i \, y_i \, \mathbf{x}_i \cdot \mathbf{x} - b}_{\text{discriminant function } \bar{g}(\mathbf{x})} \Bigg).$$

C-SVMs: Non-linear separability: Part 1

- If positive and negative samples are not linearly separable, the constraints
$$y_i\left(\mathbf{w}\cdot\mathbf{x}_i - b\right) \geq 1 \quad (i = 1, \ldots, l)$$
cannot be all fulfilled simultaneously.
- Introduce non-negative slack variables $\xi_i \geq 0$:
$$y_i\left(\mathbf{w}\cdot\mathbf{x}_i - b\right) \geq 1 - \xi_i \quad (i = 1, \ldots, l)$$
- Require slack variables to be as small as possible. Are scaled by factor $C > 0$.
- The adapted primal problem (called C-SVM) is given as:

$$\begin{aligned} \text{Minimize} \quad & \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{l}\xi_i \\ \text{subject to} \quad & -\left(y_i\left(\mathbf{w}\cdot\mathbf{x}_i - b\right) - 1 + \xi_i\right) \leq 0 \\ \text{and} \quad & -\xi_i \leq 0 \end{aligned}$$

C-SVMs: Non-linear separability: Part 2

- Using the same techniques as for linear case (i.e. formulate dual problem and apply KKT-Theorem), the problem can be cast into the framework of convex quadratic optimization again.
- Classification function g also has similar structure, however, KKT-conditions are a bit more involved
- Overview of calculations: next slides.
- As meaning of C is not very intuitive: different variant called ν-SVM also exists (see later slides). KKT Theorem applies again.
- Connection to hinge loss: $L_h(y_i, \bar{g}(\mathbf{x}_i)) = \max(0, 1 - y_i \bar{g}(\mathbf{x}_i))$. In case of SVMs:
  - L_h is zero $\Leftrightarrow$ data point lies on correct side of margin.
  - If not: loss value is proportional to distance from margin.

C-SVMs: Mathematical details: Part 1

- Again introduce $\alpha_1, \ldots, \alpha_l$ and $\lambda_1, \ldots, \lambda_l$. Then:

$$L(\mathbf{w}, b, \xi_1, \ldots, \xi_l; \alpha_1, \ldots, \alpha_l, \lambda_1, \ldots, \lambda_l)$$
$$= \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l}\xi_i - \sum_{i=1}^{l}\alpha_i\big(y_i(\mathbf{w}\cdot\mathbf{x}_i - b) - 1 + \xi_i\big) - \sum_{i=1}^{l}\lambda_i\xi_i$$

- For dual problem: minimize L for $\mathbf{w}$, b and $\xi_1, \ldots, \xi_l$.
- This enforces:

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

- Which implies (the first two conditions are the same as for linear SVMs):

$$\mathbf{w} = \sum_{i=1}^{l}\alpha_i\, y_i\, \mathbf{x}_i,$$
$$\sum_{i=1}^{l}\alpha_i\, y_i = 0,$$
$$C - \alpha_j - \lambda_j = 0 \;\; \text{for all } j = 1, \ldots, l$$

C-SVMs: Mathematical details: Part 2

- Equalities $C - \alpha_j - \lambda_j = 0$ imply $\lambda_j = C - \alpha_j$.
- Constraints $\lambda_j \geq 0$ 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

$$\mathcal{L}(\alpha_1, \ldots, \alpha_l) = \sum_{i=1}^{l} \alpha_i - \tfrac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \, \alpha_j \, y_i \, y_j \, \mathbf{x}_i \cdot \mathbf{x}_j .$$

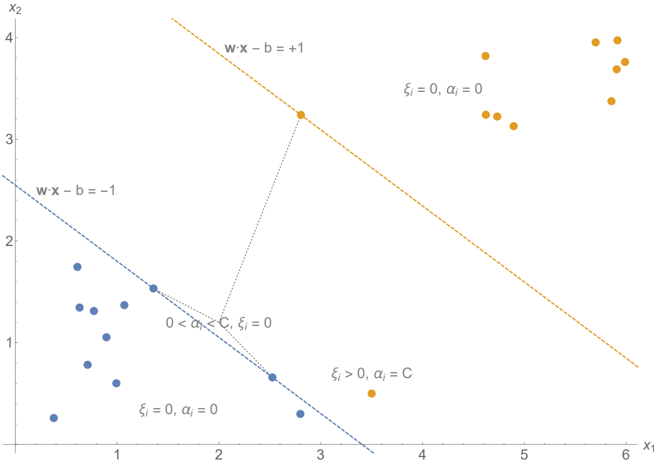
- Solution: maximize $\mathcal{L}$ with respect to α_i subject to $\alpha_i \geq 0$ (for all $i = 1, \ldots, l$), $\sum_{i=1}^{l} \alpha_i y_i = 0$, and additional constraints $\alpha_i \leq C$ (for all $i = 1, \ldots, l$).
- Dual problem:

$$\begin{array}{rl} \text{Minimize} & \dfrac{1}{2}\boldsymbol{\alpha}^T \mathbf{Q} \boldsymbol{\alpha} - \mathbf{1}^T \boldsymbol{\alpha} \\ \text{wrt.} & \boldsymbol{\alpha} \\ \text{subject to} & \boldsymbol{\alpha}^T \mathbf{y} = 0 \text{ and } \mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{1}. \end{array}$$

C-SVMs: Mathematical details: Part 3

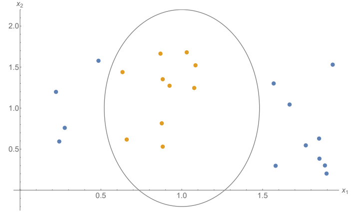
- $g(\mathbf{x}) = \text{sign}(\mathbf{w} \cdot \mathbf{x} - b) = \text{sign}\Big(\sum_{i=1}^{l} \alpha_i \, y_i \, \mathbf{x}_i \cdot \mathbf{x} - b \Big)$.
- Computation of b, however, requires a bit more caution.
- In non-separable case, the KKT-conditions tell us that $\alpha_i \big(y_i (\mathbf{w} \cdot \mathbf{x}_i - b) - 1 + \xi_i \big) = 0$ holds for all $i = 1, \dots, l$. If we choose an i such that $\alpha_i > 0$, we would need ξ_i to determine b.
- However, note that KKT 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 $i = 1, \dots, l$.
- If we find j with $0 < \alpha_j < C$, we can infer $\xi_j = 0$ and thus $y_j (\mathbf{w} \cdot \mathbf{x}_j - b) - 1 = 0$, i.e. can use same method as before.
- Every $\alpha_j > 0$ corresponds to a support vector $\mathbf{x}_j$.
- More sophisticated versions like ν-SVM also exist. (C ranges from 0 to ∞, while additional parameter ν is between 0 and 1). ν is an upper bound on the fraction of margin errors and a lower bound of the fraction of support vectors relative to the total number of training samples. E.g., $\nu = 0.05$: guaranteed to find at most 5% of training samples being misclassified and at least 5% of training samples being support vectors.

C-SVMs: Illustration

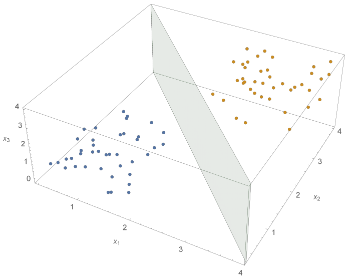


Nonlinear SVM: Part 1

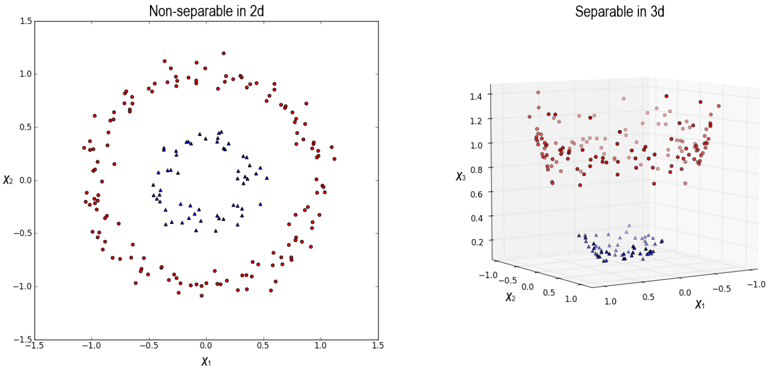
- Linear separability is very restrictive.
- The higher the dimensionality, however, the easier linear separability can be achieved.



$$\underset{\Phi^{-1}}{\overset{\Phi}{\rightleftharpoons}}$$



Nonlinear SVM: Part 2



- Left: (x_1, x_2), data not linearly separable
- Right: $(x_1, x_2, x_3 = x_1^2 + x_2^2)$, data linearly separable

Nonlinear SVM: Part 3

- Basic idea of nonlinear SVMs: transform data into a higher-dimensional space such that problem hopefully becomes linearly separable there.
- More formal: choose a Hilbert space $\mathcal{H}$ and a (nonlinear) mapping $\Phi : X \to \mathcal{H}$.
- Then try to apply linear method (presented in earlier slides) in the space $\mathcal{H}$.
- Problem: how to specify $\mathcal{H}$ and Φ?

Nonlinear SVM: Part 4

- Recall: In solving the dual problem and computing the final classification function: need only scalar products of pairs of samples. Therefore: not necessary to explicitly know $\mathcal{H}$ and Φ.
- Only need $\langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle_{\mathcal{H}}$ for all $\mathbf{x}_i, \mathbf{x}_j$ $(i, j = 1, \ldots, l)$.
- Required for computing the classification of a new sample $\mathbf{x}$: $\langle \Phi(\mathbf{x}), \Phi(\mathbf{x}_i) \rangle_{\mathcal{H}}$ for all $i = 1, \ldots, l$.
- Suppose we are given a mapping $k : X \times X \to \mathbb{R}$ (the kernel) for which we know that there exists Hilbert space $\mathcal{H}$ and mapping $\Phi : X \to \mathcal{H}$ such that $k(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle_{\mathcal{H}}$ for all $\mathbf{x}, \mathbf{x}' \in X$.

Nonlinear SVM: Part 5

- This is the case ⇔(Aronszajn) k is positive semi-definite and symmetric, i.e.
  1. $\sum_{i,j} c_i k(\mathbf{x}_i, \mathbf{x}_j) c_j \geq 0$
  2. $k(\mathbf{x}_i, \mathbf{x}_j) = k(\mathbf{x}_j, \mathbf{x}_i)$

  for $i, j = 1, ..., l$, $c_i, c_j \in \mathbb{R}$, $\mathbf{x}_i, \mathbf{x}_j \in X$.
- Equivalent formulation: Gram matrix $\mathbf{K} = (k_{ij})_{i=1,\ldots,l}^{j=1\ldots,l} = (k(\mathbf{x}_i, \mathbf{x}_j))_{i=1,\ldots,l}^{j=1\ldots,l}$ is positive semi-definite and symmetric.
- In practice: make an a priori choice of k using common sense and, if available, prior knowledge about problem: $\rightarrow$ "kernel trick".

Nonlinear SVM: Part 6

- Which kernels? $\rightarrow$ Assume $X = \mathbb{R}^n$ and $k : X^2 \rightarrow \mathbb{R}$ continuous. The following statements are equivalent:
  - k is a kernel
  - For $(Af)(\mathbf{x}) = \int_X k(\mathbf{x}, \mathbf{x}') f(\mathbf{x}') \mathrm{d}\mathbf{x}'$ the inequality
  $$\langle Af, f \rangle_{L^2(X)} = \int_{X^2} k(\mathbf{x}, \mathbf{x}')\, f(\mathbf{x})\, f(\mathbf{x}')\, \mathrm{d}\mathbf{x}\, \mathrm{d}\mathbf{x}' \geq 0.$$
  holds for all square-integrable functions $f \in L^2(X)$.
- Mercer's Theorem: If in addition the diagonal $k(\mathbf{x}, \mathbf{x})$ is integrable:
  - There are sequences $(\varphi_m)_{m \in \mathbb{N}}$ of continuous eigenfunctions and positive eigenvalues $(\sigma_m)_{m \in \mathbb{N}}$ of A.
  - $k(\mathbf{x}, \mathbf{x}') = \sum_{m \geq 1} \sigma_m \varphi_m(\mathbf{x}) \varphi_m(\mathbf{x}')$ and sum converges uniformly on compact sets of X^2.
- More details with proofs: e.g. these notes, chapter 3.5.
- Standard kernels (here: $\mathbf{x}, \mathbf{x}' \in X = \mathbb{R}^d$, "$\cdot$": Euclidean inner product):
  1. Linear: $k(\mathbf{x}, \mathbf{x}') = \mathbf{x} \cdot \mathbf{x}'$
  2. Polynomial: $k(\mathbf{x}, \mathbf{x}') = (\mathbf{x} \cdot \mathbf{x}' + \beta)^\alpha$
  3. Gaussian / RBF: $k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{x} - \mathbf{x}'\|^2\right)$
  4. Sigmoid: $k(\mathbf{x}, \mathbf{x}') = \tanh(\alpha\, \mathbf{x} \cdot \mathbf{x}' + \beta)$

Nonlinear SVM: Part 7

- The sigmoid kernel is not a very popular choice; moreover, it is not positive semi-definite for all choices of α and β.
- Information on RBF-kernel:
  1. Most popular choice
  2. Maps into a hyper-sphere of radius 1.
  3. Hilbert space corresponding to RBF kernel is infinitely dimensional.
- How to construct kernels in real-world applications?
  1. If we can define $\mathcal{H}$ (most often $\mathbb{R}^n$) and Φ explicitly $\rightarrow$ done
  2. Products, weighted sums, etc applied to positive semi-definite kernels give semi-definite kernels.
  3. Suppose that we have a mapping $\Psi : X \rightarrow X'$, where X' is some feature space, and a positive semi-definite kernel $k' : X'^2 \rightarrow \mathbb{R}$. Then $k : X^2 \rightarrow \mathbb{R}$, defined as $k(\mathbf{x}, \mathbf{x}') = k'(\Psi(\mathbf{x}), \Psi(\mathbf{x}'))$ is also a positive semi-definite kernel.

Nonlinear SVM: Part 8

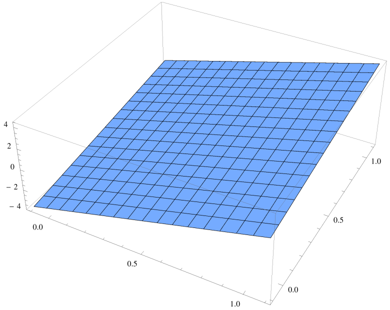
- The dual problem is now given as follows:

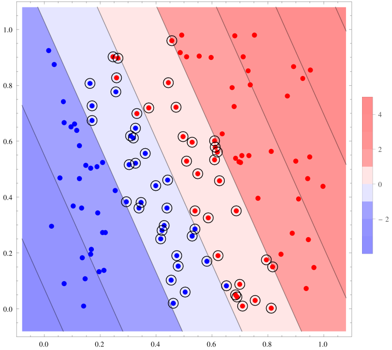
$$\text{Maximize} \quad \mathcal{L}(\alpha_1, \ldots, \alpha_l) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j).$$

$$\text{subject to} \quad 0 \leq \alpha_i \text{ and } \sum_{i=1}^{l} \alpha_i \, y_i = 0 \quad \text{for } i = 1, \ldots, l$$

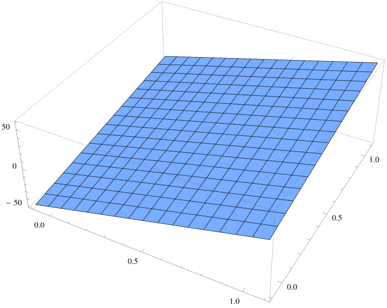
- Can also be formulated as quadratic optimization problem as before, if we use $\mathbf{Q} = \left(y_i y_j k(\mathbf{x}_i, \mathbf{x}_j)\right)_{i=1,\ldots,l}^{j=1,\ldots,l}$,
- $\rightarrow$ same tools apply, as $\mathbf{Q}$ is positive semi-definite, regardless of the possible non-linearity of kernel
- Classification function g can also be computed in similar way
- Nice source on kernels, kernel SVMs and related topics: have a look at this course.

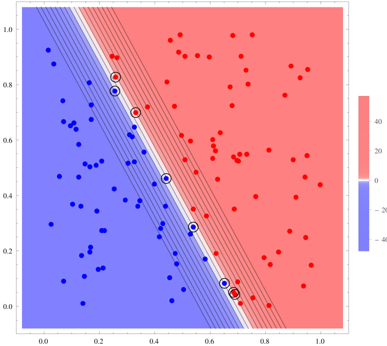
C-SVM Illustration Part 1: $C = 1$, Kernel = linear



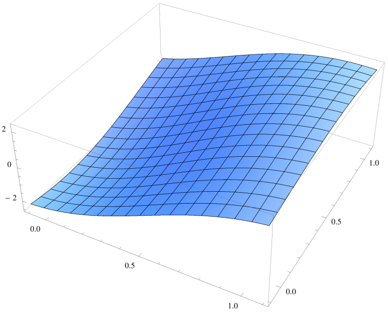


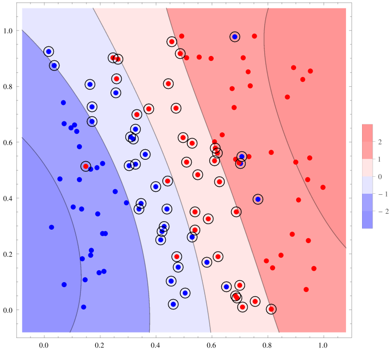
C-SVM Illustration Part 2: $C = 1000$, Kernel = linear



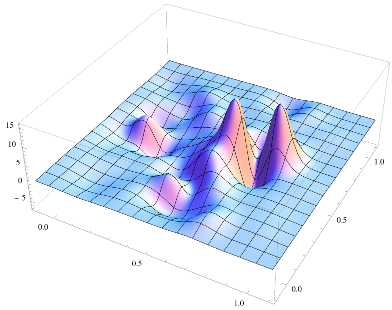


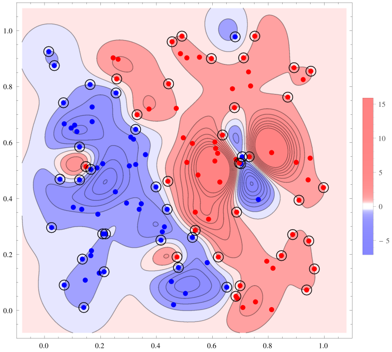
C-SVM Illustration Part 3: $C = 1$, Kernel = RBF, $\frac{1}{2\sigma^2} = 1$





C-SVM Illustration Part 4: $C = 1000$, Kernel = RBF, $\frac{1}{2\sigma^2} = 100$





SVM-based approaches to multi-class problems: Part 1

- SVMs are based on idea of separating two classes $\rightarrow$ no obvious way to extend them to multi-class problems.
- $\rightarrow$ Divide multi-class problem into several binary classification tasks.
- One of the labels versus the rest (one-versus-all):
  - Training set $\mathbf{Z}_l = (\mathbf{x}_i, y_i)_{i=1,\ldots,l}$, $y_i \in \{1, \ldots, M\}$.
  - Train M SVM classifiers to separate one class from the remaining $M-1$ ones, i.e. for $j = 1, \ldots, M$ define:
  $$\bar{g}_j(\mathbf{x}) = \sum_{i=1}^{l} \alpha_{ij} y_i^j k(\mathbf{x}_i, \mathbf{x}) - b_j,$$
  where
  $$y_i^j = \begin{cases} +1 & \text{if } y_i = j, \\ -1 & \text{otherwise.} \end{cases}$$
  - Final classification (largest certainty): $\operatorname{argmax}_{j=1,\ldots,M} \bar{g}_j(\mathbf{x})$

SVM-based approaches to multi-class problems: Part 2

- Pairwise classification (one-versus-one):
  - For $j, k \in \{1, \ldots, M\}$ select samples for which y_i is j or k.
  - Assign labels $+1$ to samples from class j and -1 to those from class k.
  - Train a binary SVM classifier on this problem. In total: $\frac{M(M-1)}{2}$ SVMs are trained.
  - New sample is assigned to class with most 'votes' from pairwise classifiers
- Which approach to use?
  - As training effort for SVMs grows faster than linear with sample number $\rightarrow$ training pairwise classifiers usually less costly (smaller training sets).
  - Classification of new samples may be slower, but improvements are possible.
  - Presently, pairwise classification is most common.

Support vector regression (SVR): Introduction: Part 1

- So far: mainly interested in sign of discriminant function of SVM.
- Constraints in optimization problems were designed to maintain equal signs of training labels and discriminant function.
- Magnitude of discriminant function was neglected (except inside the margin).
- $\rightarrow$ SVMs considered so far are useless for regression.
- $\rightarrow$ Way out: reformulate constraints such that value of discriminant function at certain training input is pushed to the actual label value.

Support vector regression (SVR): Introduction: Part 2

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

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

- Obviously: $L_\varepsilon(y, g(\mathbf{x})) = 0 \Leftrightarrow |y - g(\mathbf{x})| \leq \varepsilon$.
- $\rightarrow$ ε-insensitive loss defines ε-tube around the regression function g and checks for given sample whether it is inside. $L_\varepsilon = 0$ iff data point lies inside the tube.
- If not, loss of the sample is defined as the distance to the ε-tube.
- $\rightarrow$ basic idea behind SVR: adjust regression function such that data points are within ε-tube.

Linear ε-SVR: Part 1: Primal problem

$$
\begin{array}{rl}
\text{Minimize} & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l}(\xi_i^+ + \xi_i^-) \\
\text{for} & \mathbf{w} \in \mathbb{R}^d \text{ , } b \in \mathbb{R} \text{ ,} \\
 & (\xi_1^+, \ldots, \xi_l^+) \in \mathbb{R}^l, \text{ and } (\xi_1^-, \ldots, \xi_l^-) \in \mathbb{R}^l \\
\text{subject to} & y_i - (\mathbf{w}\cdot\mathbf{x}_i - b) \leq \varepsilon + \xi_i^+ \\
 & (\mathbf{w}\cdot\mathbf{x}_i - b) - y_i \leq \varepsilon + \xi_i^- \\
 & \xi_i^+ \geq 0 \\
 & \xi_i^- \geq 0 \\
 & \text{for } i = 1, \ldots, l
\end{array}
$$

Can again be solved via KKT-Theorem in the usual way.

Linear ε-SVR: Part 2: Interpretation

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

Linear ε-SVR: Part 3: Interpretation

- For $\varepsilon = 0$: reformulate optimization problem as follows:

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

$$\text{for} \qquad \mathbf{w} \in \mathbb{R}^d \text{ and } b \in \mathbb{R}$$

- Implications:
  - For very large C: can interpret ε-SVR with $\varepsilon = 0$ as simple data fitting according to absolute value.
  - For small C: importance of term $\frac{1}{2}\|\mathbf{w}\|^2$ increases.
- $\rightarrow$ ε-SVR is kind of ε-insensitive minimization of training error according to absolute value loss (corresponds to sum of slack values).
- $\frac{1}{2}\|\mathbf{w}\|^2$ is rather a regularization term than primary objective.

Linear ε-SVR: Part 4: Regression Function

- After solving dual problem, final regression function is given as:

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

  α_i denote associated Lagrange parameters
- To compute b: consider KKT-conditions, i.e. for $i = 1, \ldots, l$:

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

- For $0 < \alpha_j^+ < C \rightarrow \xi_j^+ = 0$ and

  $$b = y_j - \mathbf{w} \cdot \mathbf{x}_j - \varepsilon = y_j - \sum_{i=1}^{l} (\alpha_i^+ - \alpha_i^-)\, \mathbf{x}_i \cdot \mathbf{x}_j - \varepsilon.$$
- Same for α_j^- such that $0 < \alpha_j^- < C$.

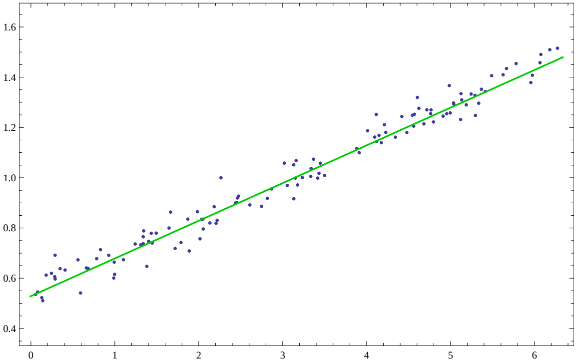
Linear ε-SVR: Part 5: Regression Function

- $0 < \alpha_i^+ < C \rightarrow \xi_i^+ = 0$ and $y_i - \mathbf{w} \cdot \mathbf{x}_i + b = \varepsilon$ hold simultaneously $\rightarrow (\mathbf{x}_i, y_i)$ is on upper border of ε-tube.
- $0 < \alpha_i^- < C \rightarrow \xi_i^- = 0$ and $-y_i + \mathbf{w} \cdot \mathbf{x}_i - b = \varepsilon$ hold simultaneously $(\mathbf{x}_i, y_i)$ is on lower border.
- $\varepsilon > 0 \rightarrow \alpha_i^+ \alpha_i^- = 0 \rightarrow$ only one of two Lagrange multipliers of sample can be non-zero.
- $\alpha_i^+ = \alpha_i^- = 0 \rightarrow \xi_i^+ = 0$ and $\xi_i^- = 0 \rightarrow i$-th sample is inside ε-tube
- If either $\alpha_i^+ > 0$ or $\alpha_i^- > 0 \rightarrow i$-th sample contributes to regression function: support vector.
- If either $\alpha_i^+ = C$ or $\alpha_i^- = C \rightarrow (\mathbf{x}_i, y_i)$ is outside ε-tube: "classification error" (and support vector).

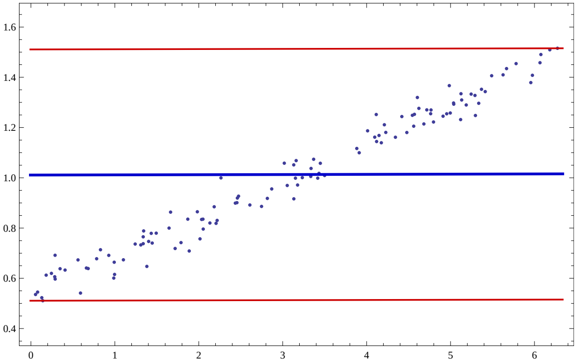
Linear ν-SVR: Intuition

- In general: Accuracy isn't only goal of SVR. Also tries to find least complex (flattest) solution fitting into ε-tube.
- For ε-SVR: choice of ε is crucial for obtaining good results.
- In practice: ε must be chosen according to the noise level, which is often unknown.
- Way out: ν-SVR: instead of specifying ε a priori, it is optimized simultaneously, where large ε is penalized and traded against smoothness and accuracy.
- The importance of ε in the objective function is weighted with a factor ν.
- The parameter ν determines the proportion of support vectors with respect to the total number of samples, i.e. which fraction of samples is on the tube border or outside.
- Will not be discussed further here. For details consider lecture notes.

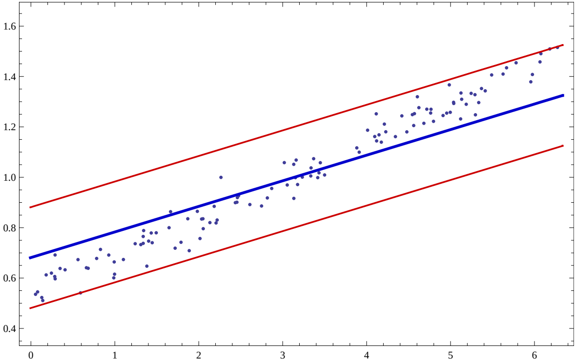
Linear SVR example: Affine linear function plus noise



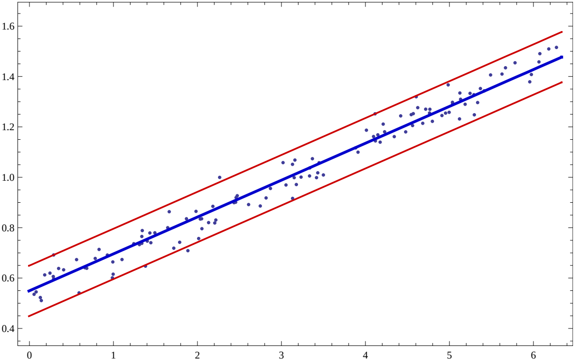
Linear SVR example: ε-SVR, $\varepsilon = 0.5$, $C = 1$



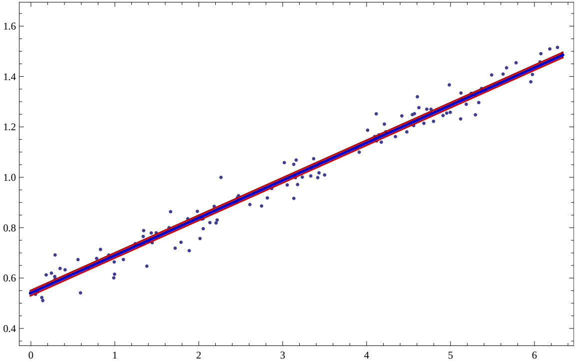
Linear SVR example: ε-SVR, $\varepsilon = 0.2$, $C = 1$



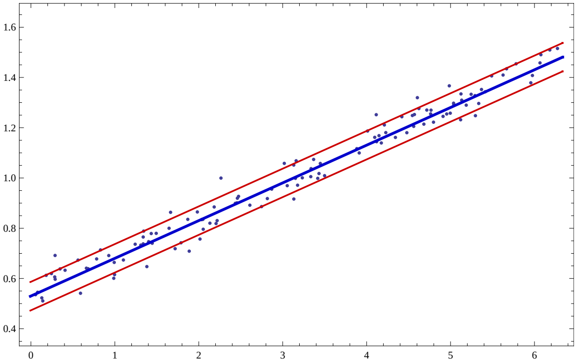
Linear SVR example: ε-SVR, $\varepsilon = 0.1$, $C = 1$



Linear SVR example: ε-SVR, $\varepsilon = 0.01$, $C = 1$



Linear SVR example: ν-SVR, $\nu = 0.2$, $C = 100$ $\to \varepsilon = 0.057$



Nonlinear SVR: Part 1

- It is clear that the usefulness of linear SVR is rather limited.
- Just like for classification: generalization to non-linear setting is done by using non-linear kernel and considering dual problem only.
- Once dual problem has been solved, the final regression function is given as

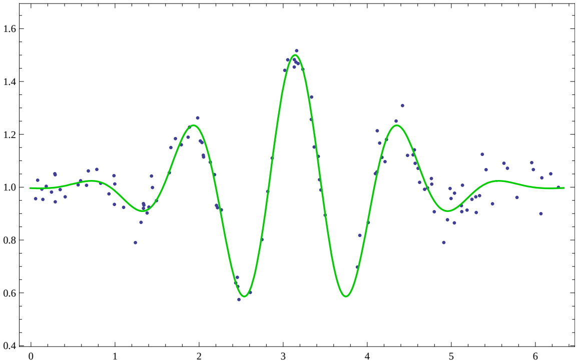
$$g(\mathbf{x}) = \sum_{i=1}^{l} (\alpha_i^+ - \alpha_i^-)\, k(\mathbf{x}_i, \mathbf{x}) - b.$$

  α_i denote again associated Lagrange parameters
- KKT-conditions are similar to linear case, similar conclusions can be drawn as well
- Also a corresponding nonlinear ν-SVR variant exists, for details: lecture notes.

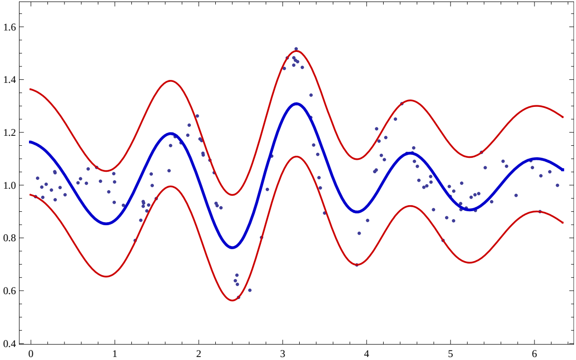
Nonlinear SVR: Part 2

- Can interpret SVR as linear combination of basis functions (plus 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 concerned with optimizing factors μ_i such that regression function fits data best.
- SVR instead tries to adjust factors μ_i such that data fit into ε-tube around regression function.
- C controls how large factors μ_i may get to achieve this goal.

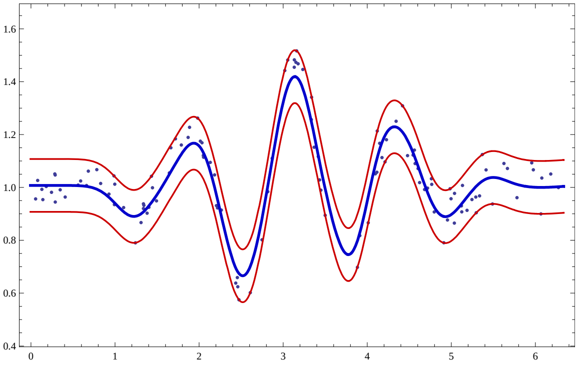
Nonlinear SVR example: $f(x) = 1 + \frac{1}{2}\cos(5(x-\pi)) \cdot \exp(-\frac{1}{2}(x-\pi)^2)$ plus noise



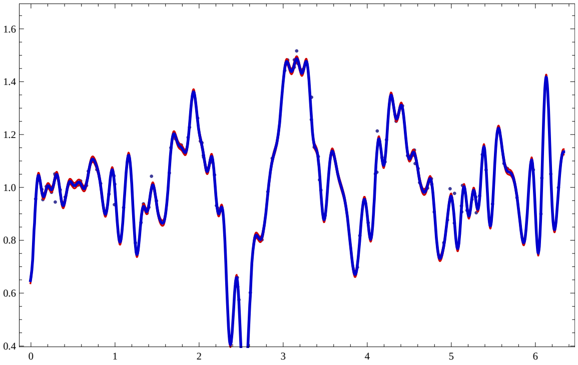
Nonlinear SVR example: ε-SVR, $\varepsilon = 0.2$, $C = 10$, kernel=RBF, $\frac{1}{2\sigma^2} = 1$



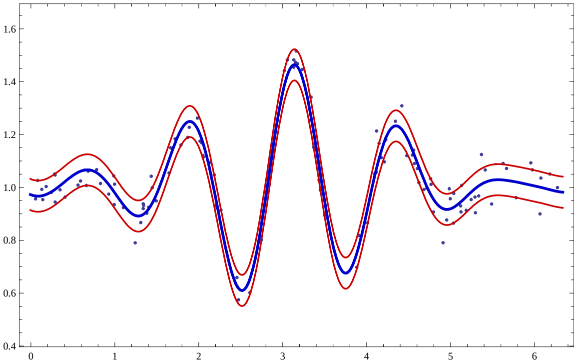
Nonlinear SVR example: ε-SVR, $\varepsilon = 0.1$, $C = 10$, kernel=RBF, $\frac{1}{2\sigma^2} = 10$



Nonlinear SVR example: ε-SVR, $\varepsilon = 0.01$, $C = 100$, kernel=RBF, $\frac{1}{2\sigma^2} = 100$



Nonlinear SVR example: ν-SVR, $\nu = 0.2$, $C = 1000$, kernel=RBF, $\frac{1}{2\sigma^2} = 1 \rightarrow \varepsilon = 0.059$



What we (unfortunately) didn't discuss

- 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; also useful for feature selection.
- How are SVMs implemented? Especially with algorithms used to solve convex quadratic optimization problems, e.g. SMO Algorithm.

Why not use SVMs for any supervised task? Pros and cons

- Pros
  - Built on a solid theoretical foundation.
  - Both training and testing are deterministic and fast.
  - Optimization problem has global solution (not true for most other ML-algorithms).
  - Effective if number of dimensions $d \gg$ number of samples l.
- Cons
  - SVMs are not suitable for large data sets.
  - Not suitable if number of features for each data point $d \ll$ number of training data samples l.
  - Bad performance if data set has a lot of noise, i.e. if target classes are overlapping.
  - SVM puts data points above and below classifying hyper plane $\rightarrow$ no probabilistic explanation for classification.

Summary

- Linear SVMs
  - Basics of convex optimization
  - Rigorous derivation for linearly separable data
  - Nonlinear separability: basic concepts and ideas for C-SVMs
- Nonlinear SVMs
  - Kernel trick
  - Discussion of corresponding dual problem
  - How to find the right kernels?
- Multi-class SVMs
- Support vector regression(SVR): linear (with more details) + nonlinear (no details)
- Pros and Cons of SVMs in general

Next: further state of the art methods that are more suitable for large data sets