Support Vector Machines for Regression

Provided with n training data points $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_n, y_n)\} \subseteq \mathbb{R}^s \times \mathbb{R}$ we seek a function f for a fixed $\epsilon > 0$ such that:

$$|f(\boldsymbol{x}_i) - y_i| \leq \epsilon \tag{1}$$

Let us consider the space of linear functions for now and take $f(\boldsymbol{x}_i) = \langle \boldsymbol{w} \cdot \boldsymbol{x}_i \rangle + b$ with $\boldsymbol{w} \in \mathbb{R}^s$ and $b \in \mathbb{R}$. In order to avoid generalisation errors (over-fitting of the regression surface) we will require the function f to be as flat as possible so we want to minimise $\boldsymbol{w}$. We define a *convex quadratic* optimisation:

$$\begin{aligned} &\min \tfrac{1}{2}\boldsymbol{w}\boldsymbol{w}^T \\ &\text{subject to:} \quad y_i - \langle \boldsymbol{w} \cdot \boldsymbol{x}_i \rangle - b \leq \epsilon \\ &\qquad\qquad\quad\;\; \langle \boldsymbol{w} \cdot \boldsymbol{x}_i \rangle + b - y_i \leq \epsilon \end{aligned} \tag{2}$$

Feasibility

We assumed the existence of a function f satisfying the constraints. It is possible that for a given ϵ no function satisfying the constraint $|f(\boldsymbol{x}_i) - y_i| \leq \epsilon$ exists. So we define slack varibles $\psi_i > 0$ and $\phi_i > 0$ and re-write the optimization as:

$$\text{minimise } \tfrac{1}{2}\boldsymbol{w}\boldsymbol{w}^T + \zeta \textstyle\sum_i^n (\psi_i + \phi_i)$$

subject to:

$$\begin{aligned} & y_i - \langle \boldsymbol{w} \cdot \boldsymbol{x}_i \rangle - b \leq \epsilon + \psi_i \\ & \langle \boldsymbol{w} \cdot \boldsymbol{x}_i \rangle + b - y_i \leq \epsilon + \phi_i \\ & \psi_i \geq 0, \phi_i \geq 0, \forall n \end{aligned} \tag{3}$$

The constant ζ maintains the trade-off between how much deviation greater than ϵ is permitted versus the generalisation or in this case the flatness of the regression function.

We define the Lagrangian; the objective function plus a linear combination of the constraints:

$$\begin{aligned} L(\boldsymbol{w}, b, \boldsymbol{\phi}, \boldsymbol{\psi} | \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\eta}, \boldsymbol{\eta}^*) &= \frac{1}{2}\|w\|^2 + \zeta \sum_i^n (\psi_i + \phi_i) \\ &- \sum_{i=1}^n \alpha_i(\epsilon + \psi_i - y_i + \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \\ &- \sum_{i=1}^n \beta_i(\epsilon + \phi_i + y_i - \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle - b) \\ &- \sum_{i=1}^n (\eta_i \psi_i + \eta_i^* \phi_i) \qquad (4) \end{aligned}$$

where α_i, β_i, η_i and η_i^* are non-negative dual variables or lagrange multipliers.

Lagrangian Dual Problem

The dual of the lagrangian is defined as:

$$L_{dual}(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\eta}, \boldsymbol{\eta}^*) = \inf_{w,b,\phi,\psi} L(\boldsymbol{w}, b, \boldsymbol{\phi}, \boldsymbol{\psi} | \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\eta}, \boldsymbol{\eta}^*) \tag{5}$$

The Lagrangian Dual Problem is then:

$$\begin{aligned} \max L_{dual}(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\eta}, \boldsymbol{\eta}^*) \\ \text{subject to: } \boldsymbol{\alpha} \geq 0 \text{ and } \boldsymbol{\beta} \geq 0 \text{ and } \boldsymbol{\eta} \geq 0 \text{ and } \boldsymbol{\eta}^* \geq 0 \end{aligned} \tag{6}$$

Lagrangian Weak Duality Theorem

The primal objective function f and its dual L_{dual} satisfy

$$f(\boldsymbol{w}, b, \boldsymbol{\psi}, \boldsymbol{\phi}) \geq L_{dual}(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\eta}, \boldsymbol{\eta}^*) \tag{7}$$

Proof:

$$\begin{aligned} L_{dual}(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\eta}, \boldsymbol{\eta}^*) &= \inf_{w,b,\phi,\psi} L(\boldsymbol{w}, b, \boldsymbol{\phi}, \boldsymbol{\psi} | \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\eta}, \boldsymbol{\eta}^*) \\ &\leq L(\boldsymbol{w}, b, \boldsymbol{\phi}, \boldsymbol{\psi} | \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\eta}, \boldsymbol{\eta}^*) \\ &\leq f(\boldsymbol{w}, b, \boldsymbol{\psi}, \boldsymbol{\phi}) \end{aligned} \tag{8}$$

Lagrangian Strong Duality Theorem

1. A real valued function is said to be convex if $\forall \boldsymbol{w}, \boldsymbol{u} \in \mathbb{R}^s$ and for any $\theta \in (0, 1)$ we have: $f(\theta\boldsymbol{w} + (1-\theta)\boldsymbol{u}) \leq \theta f(\boldsymbol{w}) + (1-\theta)f(\boldsymbol{u})$
2. Every strictly convex constrained optimisation problem has a unique solution.
3. An affine function can be expressed in the form $f(\boldsymbol{w}) = \boldsymbol{A}\boldsymbol{w} + \boldsymbol{b}$.
4. Affine functions are convex.
5. A convex optimisation problem has a convex objective function and affine constraints.

Strong Duality Theorem: For a convex optimisation problem *the duality gap is zero* at primal optimality.

Karush-Kuhn-Tucker Complementary Conditions

Let $(\boldsymbol{\alpha}_o, \boldsymbol{\beta}_o, \boldsymbol{\eta}_o, \boldsymbol{\eta^*}_o)$ and $(\boldsymbol{w}_o, b, \boldsymbol{\psi}_o, \boldsymbol{\phi}_o)$ be optimal solutions of the dual and primal respectively, then $L_{dual}(\boldsymbol{\alpha}_o, \boldsymbol{\beta}_o, \boldsymbol{\eta}_o, \boldsymbol{\eta^*}_o) = f(\boldsymbol{w}_o, b, \boldsymbol{\psi}_o, \boldsymbol{\phi}_o)$. From (4) and (8) we then derive the KKT conditions:

$$\begin{aligned} \alpha_i(\epsilon + \psi_i - y_i + \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) &= 0 \\ \beta_i(\epsilon + \phi_i + y_i - \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle - b) &= 0 \\ \eta_i \psi_i &= 0 \\ \eta_i^* \phi_i &= 0 \\ i &= 1, \cdots, n \end{aligned} \tag{9}$$

A constraint $c_t(\boldsymbol{w})$ is said to be tight or active if the solution $\boldsymbol{w}_o$ satisfies $c_t(\boldsymbol{w}_o) = 0$ and is otherwise said to be inactive.

Lagrangian Saddlepoint Equivalence Theorem

When we have an optimal primal solution (so the duality gap is zero) it is a saddle point of the lagrangian function of the primal problem and so we have:

$$\begin{aligned}\frac{\partial L}{\partial b} &= \textstyle\sum_{i=1}^{n}(\beta_i - \alpha_i) &= 0 \qquad (10)\\ \frac{\partial L}{\partial \boldsymbol{w}} &= \boldsymbol{w}^* - \textstyle\sum_{i=1}^{n}(\alpha_i - \beta_i)\boldsymbol{x}_i &= 0 \qquad (11)\\ \frac{\partial L}{\partial \phi_i} &= \zeta - \beta_i - \eta_i^* &= 0 \qquad (12)\\ \frac{\partial L}{\partial \psi_i} &= \zeta - \alpha_i - \eta_i &= 0 \qquad (13)\end{aligned}$$

(11) can be written as $\boldsymbol{w} = \sum_i^n(\alpha_i - \beta_i)\boldsymbol{x}_i$ and so
$f(\boldsymbol{x}_a) = \langle \boldsymbol{w} \cdot \boldsymbol{x}_a \rangle + b = \sum_{i=1}^{n}(\alpha_i - \beta_i)\langle \boldsymbol{x}_i \cdot \boldsymbol{x}_a \rangle + b$

Dual Formulation

To remove the dependance on the primal variables we explicitly compute (5) by differentiating $L(\boldsymbol{w}, b, \boldsymbol{\phi}, \boldsymbol{\psi} | \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\eta}, \boldsymbol{\eta}^*)$ with respect to the primal variables which leaves us with (10) to (13) which we substitute into (4):

$$\begin{array}{lll} \text{maximise} & & -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\alpha_i - \beta_i)(\alpha_j - \beta_j)\langle \boldsymbol{x}_i \cdot \boldsymbol{x}_j \rangle \\ & & -\epsilon\sum_{i=1}^{n}(\alpha_i + \beta_i) + \sum_{i=1}^{n} y_i(\alpha_i - \beta_i) \\ \text{subject to:} & \sum_{i=1}^{n}(\alpha_i - \beta_i) & = 0 \\ & \alpha_i, \beta_i & \in [0, \zeta] \end{array} \tag{14}$$

So the dual of our quadratic program (3) is another quadratic program but with simpler constraints.

Support Vectors

The Karush-Kuhn-Tucker complementary conditions state that only the active constraints will have non-zero dual variables.

$$
\begin{aligned}
\epsilon + \phi_i - y_i + f(\boldsymbol{x}_i) > 0 &\text{ implies } \alpha_i = 0 \\
\epsilon + \phi_i - y_i + f(\boldsymbol{x}_i) = 0 &\text{ implies } \alpha_i \neq 0 \\
\epsilon + \psi_i + y_i - f(\boldsymbol{x}_i) > 0 &\text{ implies } \beta_i = 0 \\
\epsilon + \psi_i + y_i - f(\boldsymbol{x}_i) = 0 &\text{ implies } \beta_i \neq 0 \\
\forall i : 1 \leq i \leq n &
\end{aligned} \tag{15}
$$

The $\boldsymbol{x}_i$ with non-zero α_i **or** β_i are the *support vectors*; if we were to train the SVM on only these $\boldsymbol{x}_i$, ignoring all the examples for which $\alpha_i = 0$ **and** $\beta_i = 0$, *we would still induce the same regression surface.*

Sparse Support Vector Expansion

We see that the vector $\boldsymbol{w}$ can be written as a linear combination of the input training data points $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_n, y_n)\} \subseteq \mathbb{R}^s \times \mathbb{R}$ and so:

$$f(\boldsymbol{x}_a) = \langle \boldsymbol{w} \cdot \boldsymbol{x}_a \rangle + b = \sum_{i=1}^{n} (\alpha_i - \beta_i) \langle \boldsymbol{x}_i \cdot \boldsymbol{x}_a \rangle + b \tag{16}$$

Let $\Lambda \subseteq \{1, 2, \cdots, n\}$ such that $\forall i \in \Lambda$ we have *both* $\alpha_i \neq 0$ and $\beta_i \neq 0$. Then we can rewrite our regression function as:

$$f(\boldsymbol{x}_a) = \sum_{i \in \Lambda} (\alpha_i - \beta_i) \langle \boldsymbol{x}_i \cdot \boldsymbol{x}_a \rangle + b \tag{17}$$

We have a sparse expansion of $f(\boldsymbol{x}_a)$.

Non-linear SVM Regression - $\Phi : \mathbb{R}^s \rightarrow \mathscr{F}$

Our machine is linear. Our data might be non-linear.

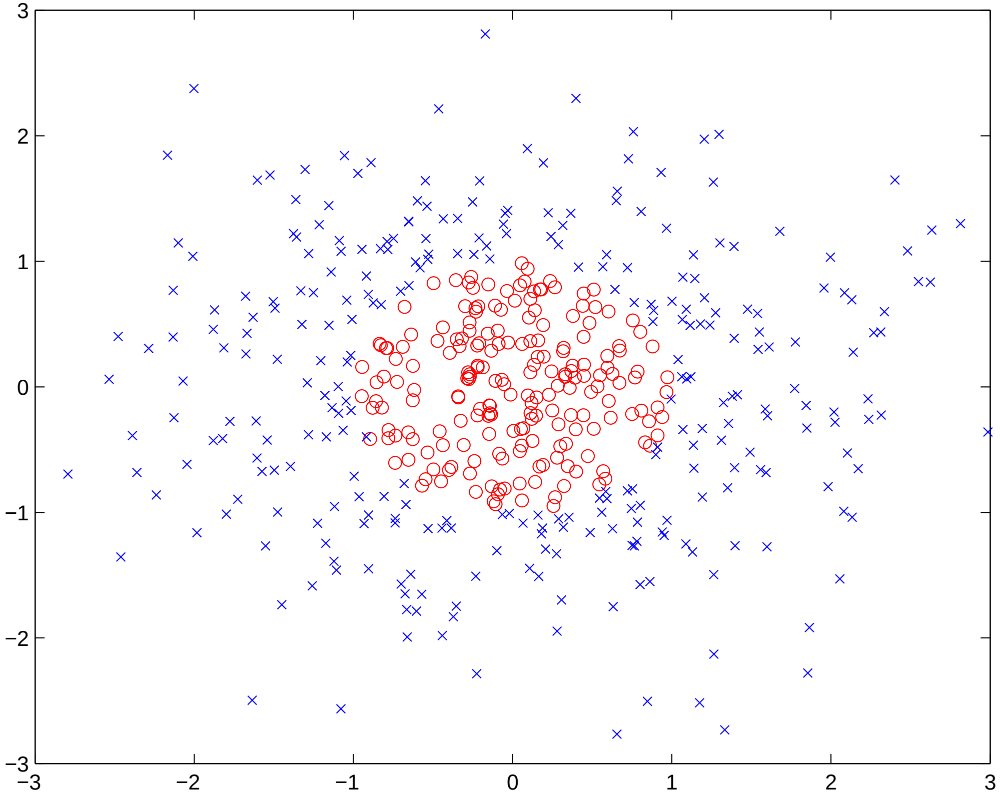
We will apply a mapping function Φ to our input data, essentially projecting it into a higher dimensional feature space $\mathscr{F}$.

$$f(\boldsymbol{x}_i) = \langle \boldsymbol{w} \cdot \Phi(\boldsymbol{x}_i) \rangle + b \tag{18}$$

Then apply our linear machinery to find a linear regression in this new feature space. The corresponding regression surface in the input space will be non-linear, specifically $\Phi^{-1}(f(x))$.

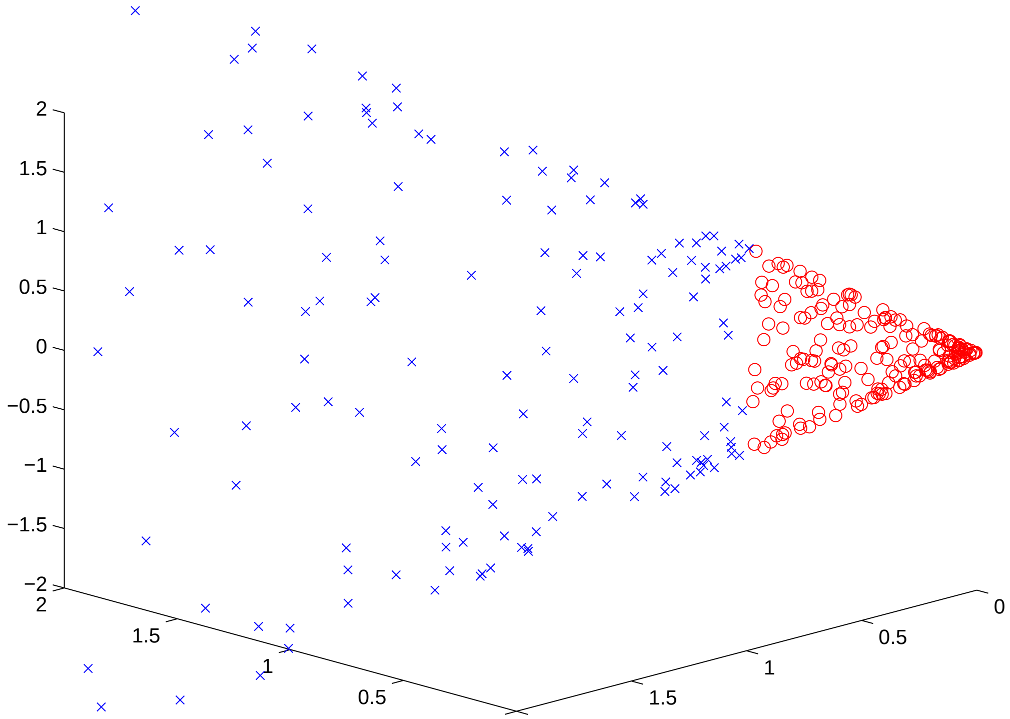
Example: Quadratic Map

$$(u_1, u_2) \to \phi(u_1, u_2) = (u_1^2, u_2^2, \sqrt{2}u_1u_2) \qquad (19)$$



Input Space: circular data in $\mathbb{R}^2$

Input data mapped into Feature Space



Data is linearly separable in feature space

Computational Feasibility

*W*hat computations are we performing in the feature space?

Since we are projecting only our input data $\boldsymbol{x}_i$ - the dual form (14) has *slightly* changed; instead of computing the dot product between training examples in the input space $\langle \boldsymbol{x}_i \cdot \boldsymbol{x}_j \rangle$, we compute a a dot product in the feature space $\mathscr{F}$:

$$\langle \Phi(\boldsymbol{x}_i) \cdot \Phi(\boldsymbol{x}_j) \rangle \tag{20}$$

If the dimension of $\mathscr{F}$ is large then our problem might be computationally infeasible. We need to avoid computing in the feature space and hence projecting data points into the feature space.

But we still have to compute the dot product of the features efficiently.

Kernel Functions

What is a kernel: a function $\boldsymbol{K}$ such that for all $\boldsymbol{x}, \boldsymbol{z}$ in some input space:

$$\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{z}) = \langle \boldsymbol{\phi}(\boldsymbol{x}) \cdot \boldsymbol{\phi}(\boldsymbol{z}) \rangle \tag{21}$$

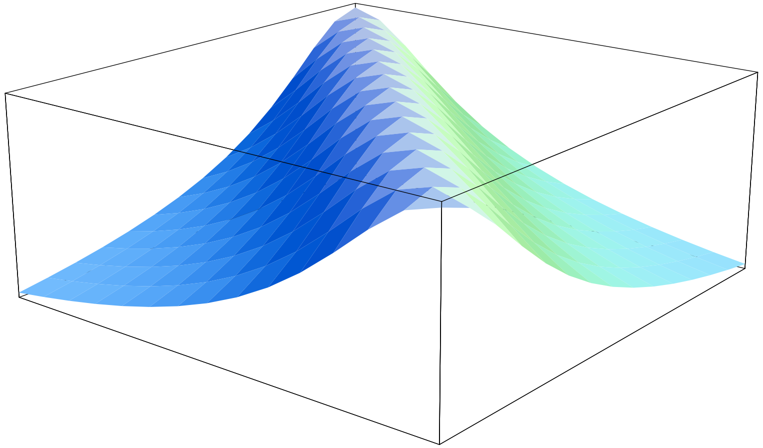
where ϕ is a mapping from the input space to a feature space.

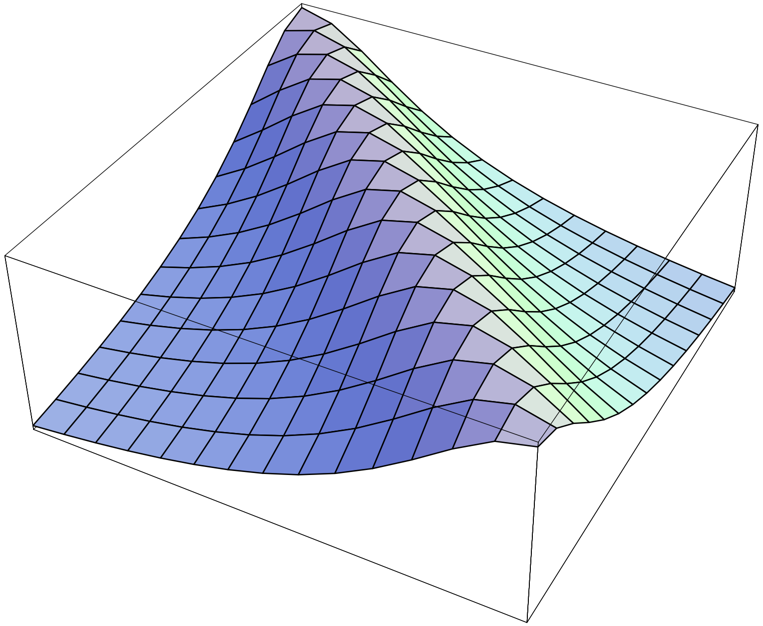
Mercer's Theorem: characterises what constitutes a valid kernel and how they can be built.

So the dimension of the feature space does *not* affect the computational complexity!

Inverse Multiquadric Kernel

$$k(\mathbf{x},\mathbf{y}) = \frac{\mathbf{1}}{\sqrt{\|\mathbf{x}-\mathbf{y}\|^{\mathbf{2}}+\mathbf{c}^{\mathbf{2}}}} \tag{22}$$





Quadratic Optimization using a Kernel Function

$$\begin{aligned} \text{maximise} \quad & -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\alpha_i - \beta_i)(\alpha_j - \beta_j)\boldsymbol{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ & -\epsilon\sum_{i=1}^{n}(\alpha_i + \beta_i) + \sum_{i=1}^{n} y_i(\alpha_i - \beta_i) \\ \text{subject to:} \quad \textstyle\sum_{i=1}^{n}(\alpha_i - \beta_i) \quad & = 0 \\ \alpha_i, \beta_i \quad & \in [0, \zeta] \end{aligned} \tag{23}$$

So using the dual representation helps us in two ways:

1. we can operate in high dimensional spaces with the help of kernel functions which replace the dot product in the dual

2. lets us make use of optimisation algortihms designed for the dual form