Regression Part II

Note: Several slides taken from tutorial by Bernard Schölkopf

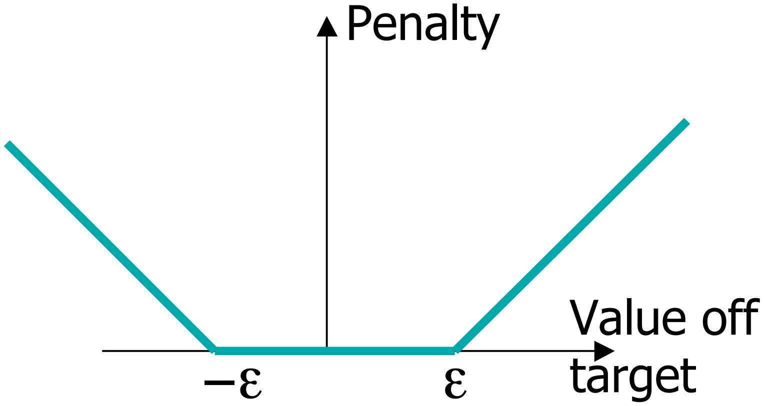
Multi-class Classification

- SVM is basically a two-class classifier
- One can change the QP formulation to allow multi-class classification
- More commonly, the data set is divided into two parts “intelligently” in different ways and a separate SVM is trained for each way of division
- Multi-class classification is done by combining the output of all the SVM classifiers
  - Majority rule
  - Error correcting code
  - Directed acyclic graph

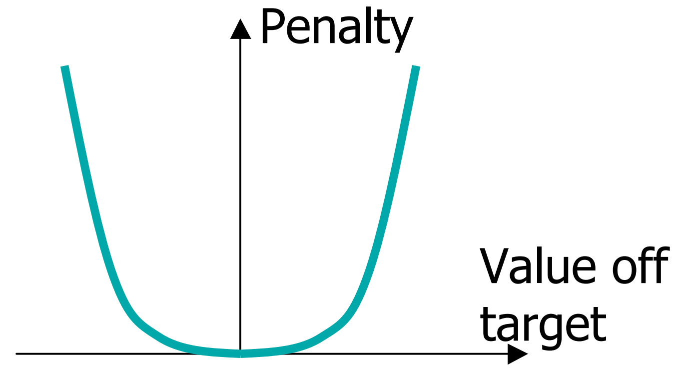
Epsilon Support Vector Regression (ε-SVR)

- Linear regression in feature space
- Unlike in least square regression, the error function is ε -insensitive loss function
  - Intuitively, mistake less than ε is ignored
  - This leads to sparsity similar to SVM

ε-insensitive loss function



Square loss function



Epsilon Support Vector Regression (ε-SVR)

- Given: a data set $\{x_1, ..., x_n\}$ with target values $\{u_1, ..., u_n\}$, we want to do ε-SVR
- The optimization problem is

$$\text{Min } \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*)$$

$$\text{subject to } \begin{cases} u_i - \mathbf{w}^T\mathbf{x}_i - b \le \epsilon + \xi_i \\ \mathbf{w}^T\mathbf{x}_i + b - u_i \le \epsilon + \xi_i^* \\ \xi_i \ge 0, \xi_i^* \ge 0 \end{cases}$$

- Similar to SVM, this can be solved as a quadratic programming problem

Epsilon Support Vector Regression (ε-SVR)

- C is a parameter to control the amount of influence of the error
- The $\frac{1}{2}||w||^2$ term serves as controlling the complexity of the regression function
  - This is similar to ridge regression
- After training (solving the QP), we get values of α_i and α_i^*, which are both zero if $\mathbf{x}_i$ does not contribute to the error function
- For a new data **z**,

$$f(\mathbf{z}) = \sum_{j=1}^{s} (\alpha_{t_j} - \alpha_{t_j}^*) K(\mathbf{x}_{t_j}, \mathbf{z}) + b$$

SV Regression: ε-Insensitive Loss

Goal: generalize SV pattern recognition to regression, preserving the following properties:

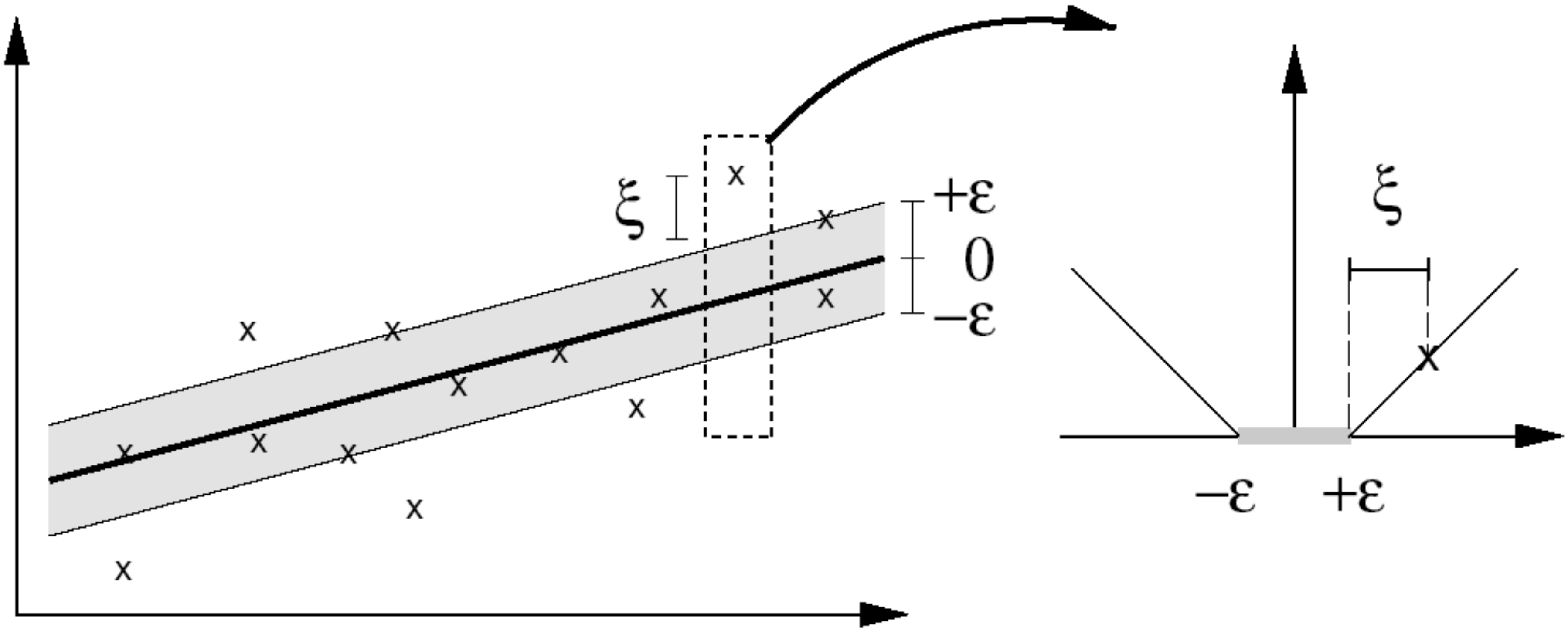
- formulate the algorithm for the linear case, and then use kernel trick
- sparse representation of the solution in terms of SVs

ε-Insensitive Loss:

$$|y - f(\mathbf{x})|_\varepsilon := \max\{0, |y - f(\mathbf{x})| - \varepsilon\}$$

Estimate a linear regression $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$ by minimizing

$$\frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{m}\sum_{i=1}^{m} |y_i - f(\mathbf{x}_i)|_\varepsilon.$$



ξ
+ε
0
−ε
ξ
−ε
+ε

Formulation as an Optimization Problem

Estimate a linear regression

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$$

with precision ε by minimizing

$$\begin{aligned}
\text{minimize} \quad & \tau(\mathbf{w}, \boldsymbol{\xi}, \boldsymbol{\xi}^*) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}(\xi_i + \xi_i^*) \\
\text{subject to} \quad & (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) - y_i \leq \varepsilon + \xi_i \\
& y_i - (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \leq \varepsilon + \xi_i^* \\
& \xi_i, \xi_i^* \geq 0
\end{aligned}$$

for all $i = 1, \ldots, m$.

Dual Problem, In Terms of Kernels

For $C > 0, \varepsilon \geq 0$ chosen a priori,

$$\text{maximize} \quad W(\boldsymbol{\alpha}, \boldsymbol{\alpha}^*) = -\varepsilon \sum_{i=1}^{m} (\alpha_i^* + \alpha_i) + \sum_{i=1}^{m} (\alpha_i^* - \alpha_i) y_i - \frac{1}{2} \sum_{i,j=1}^{m} (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) k(\mathbf{x}_i, \mathbf{x}_j)$$

$$\text{subject to} \quad 0 \leq \alpha_i, \alpha_i^* \leq C, \; i = 1, \ldots, m, \; \text{and} \; \sum_{i=1}^{m} (\alpha_i - \alpha_i^*) = 0.$$

The regression estimate takes the form

$$f(\mathbf{x}) = \sum\nolimits_{i=1}^{m} (\alpha_i^* - \alpha_i) k(\mathbf{x}_i, \mathbf{x}) + b,$$

ν-SV Regression

We want to estimate the noise as well -

Introduce a parameter that bounds the noise and minimize

Primal problem: for $0 \leq \nu \leq 1$, minimize

$$\tau(\mathbf{w}, \varepsilon) = \frac{1}{2}\|\mathbf{w}\|^2 + C\left(\nu\varepsilon + 1/m\sum_{i=1}^{m}|y_i - f(\mathbf{x}_i)|_\varepsilon\right)$$

Duals, Using Kernels

C-SVM dual: maximize

$$W(\boldsymbol{\alpha}) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j)$$

subject to $0 \le \alpha_i \le C, \quad \sum_i \alpha_i y_i = 0.$

ν-SVM dual: maximize

$$W(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{ij} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j)$$

subject to $0 \le \alpha_i \le \frac{1}{m}, \quad \sum_i \alpha_i y_i = 0, \quad \sum_i \alpha_i \ge \nu$

In both cases: *decision function:*

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{m} \alpha_i y_i k(\mathbf{x}, \mathbf{x}_i) + b\right)$$

Soft Margin SVMs

C-SVM [15]: for $C > 0$, minimize

$$\tau(\mathbf{w}, \boldsymbol{\xi}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i$$

subject to $y_i \cdot (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$ (margin $2/\|\mathbf{w}\|$)

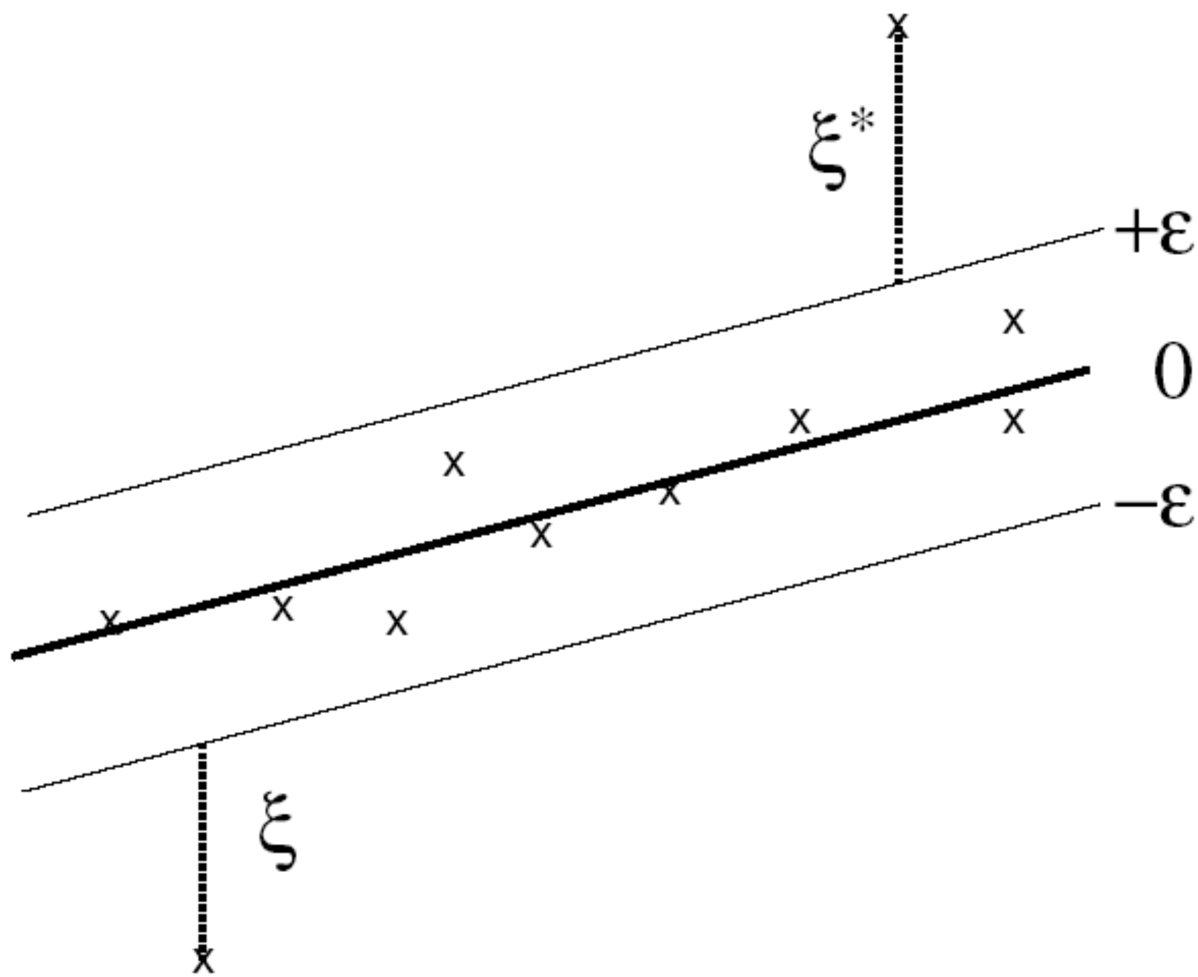
ν-SVM [55]: for $0 \leq \nu < 1$, minimize

$$\tau(\mathbf{w}, \boldsymbol{\xi}, \rho) = \frac{1}{2}\|\mathbf{w}\|^2 - \nu\rho + \frac{1}{m}\sum_{i}\xi_i$$

subject to $y_i \cdot (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq \rho - \xi_i, \quad \xi_i \geq 0$ (margin $2\rho/\|\mathbf{w}\|$)

B. Schölkopf, Canberra, February 2002

Illustration



Cost function: $\frac{1}{2C}\|\mathbf{w}\|^2 + \nu\varepsilon + \frac{1}{m}\sum_{i=1}^{m}(\xi_i + \xi_i^*)$

The ν-Property

Proposition 3 *Assume $\varepsilon > 0$. The following statements hold:*

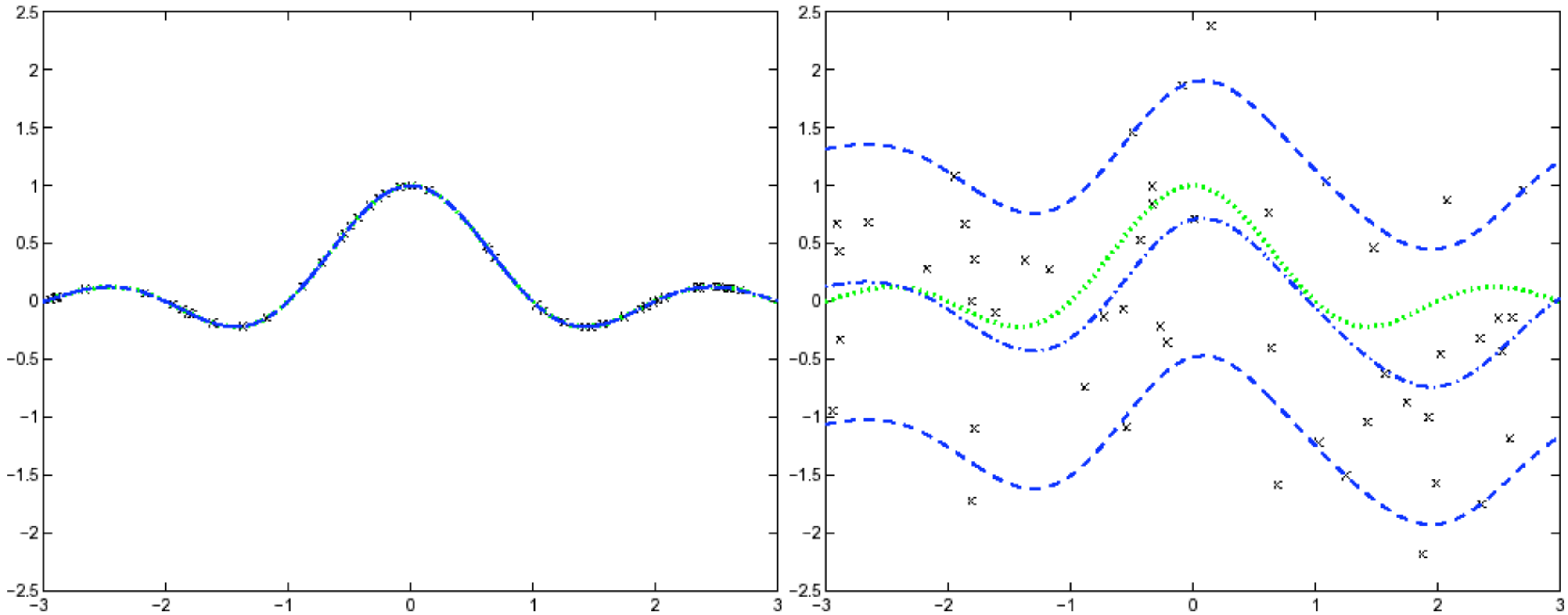
(i) ν is an upper bound on the fraction of errors.

(ii) ν is a lower bound on the fraction of SVs.

(iii) Suppose the data were generated iid from a 'well-behaved distribution $\mathrm{P}(\mathbf{x}, y)$. With probability 1, asymptotically, ν equals both the fraction of SVs and the fraction of errors.*

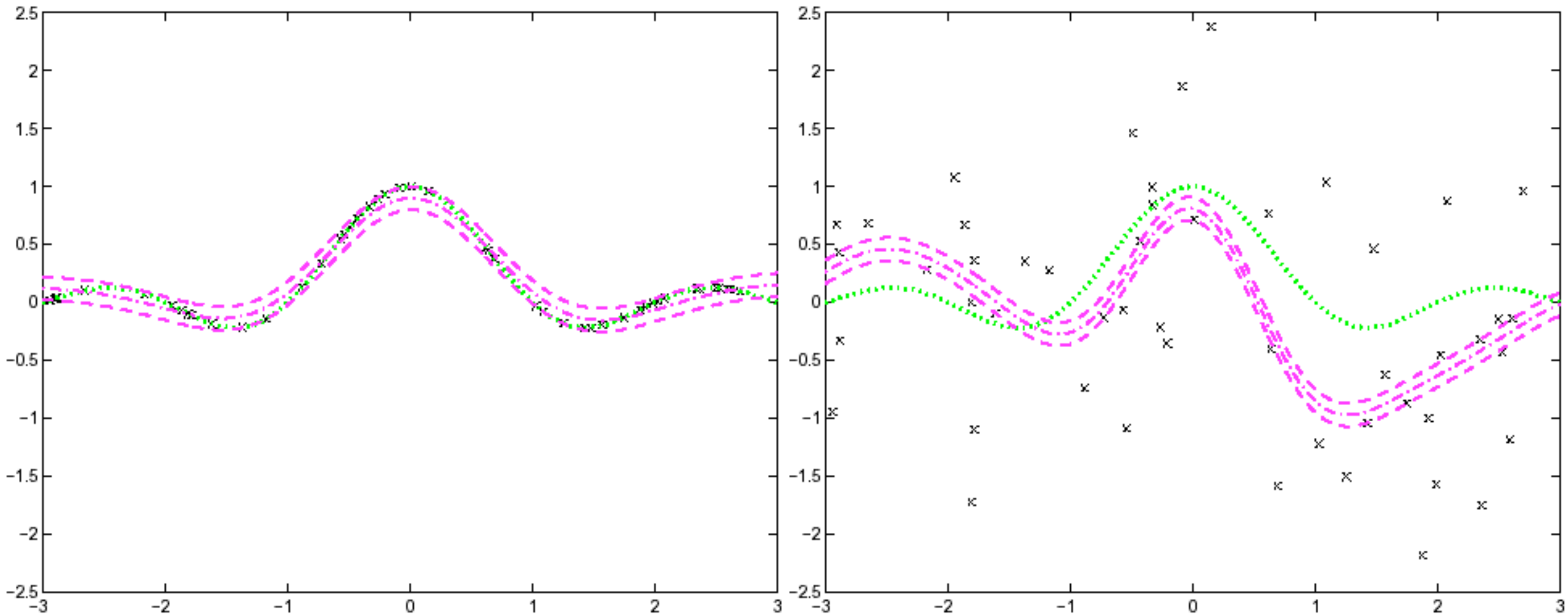
* Essentially, $\mathrm{P}(\mathbf{x}, y) = \mathrm{P}(\mathbf{x})\mathrm{P}(y|\mathbf{x})$ with $\mathrm{P}(y|\mathbf{x})$ continuous (some details omitted).

ν-SV-Regression: Automatic Tube Tuning



Identical machine parameters ($\nu = 0.2$), but different amounts of noise in the data.

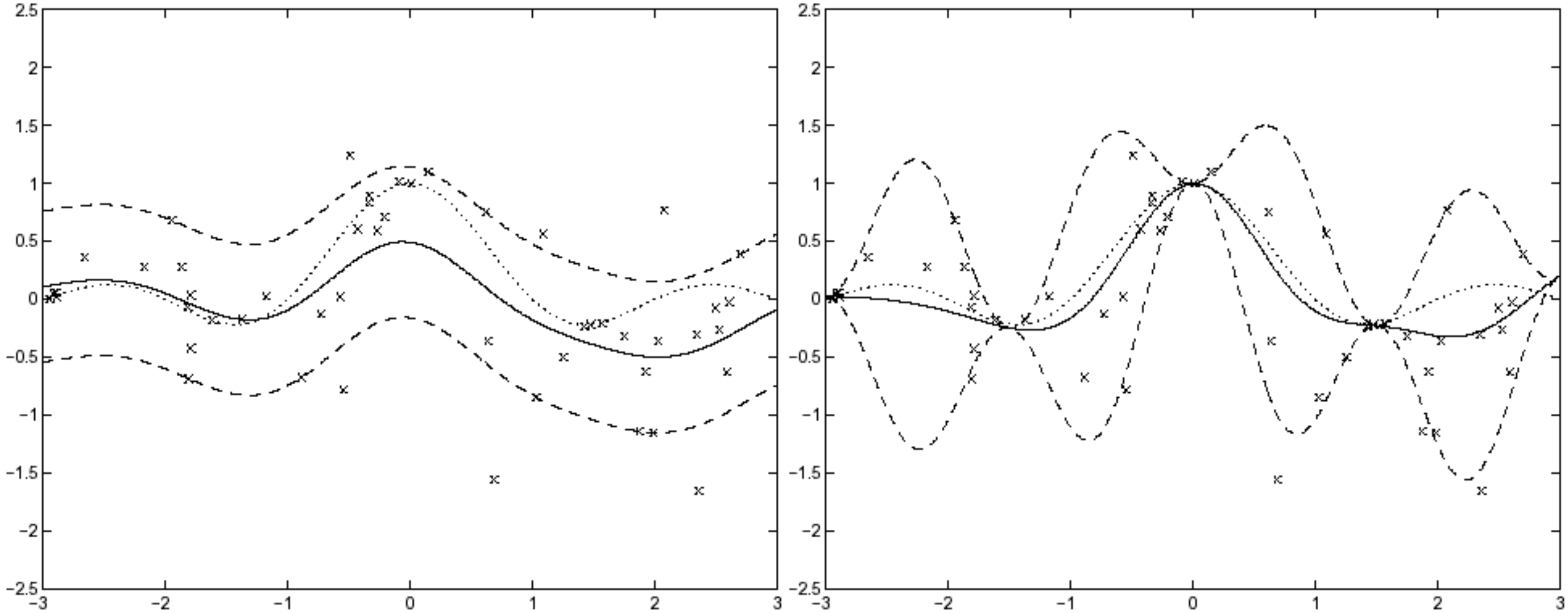
ε-SV-Regression, Run on the Same Data



Identical machine parameters ($\varepsilon = 0.2$), but different amounts of noise in the data.

Handling Heteroscedasticity

Assumption: we have prior knowledge indicating that the noise is modulated by $\zeta(x) = \sin^2((2\pi/3)x)$.



constant-radius tube | parametric model using $\zeta(x)$

Robustness of SV Regression

Proposition. Using SVR with $|.|_\varepsilon$, local movements of target values of points outside the tube do not change the estimated regression.

Proof.

1. Shift y_i locally $\longrightarrow$ $(\mathbf{x}_i, y_i)$ still outside the tube $\longrightarrow$ original dual solution $\boldsymbol{\alpha}^{(*)}$ still feasible ($\alpha_i^{(*)} = C$, since *all* points outside the tube are at the upper bound).
2. The primal solution, with ξ_i transformed according to the movement, is also feasible.
3. The KKT conditions are still satisfied, as still $\alpha_i^{(*)} = C$. Thus [5, e.g.], $\boldsymbol{\alpha}^{(*)}$ is still the optimal solution.

The Representer Theorem

Theorem 4 *Given: a p.d. kernel k on $\mathcal{X} \times \mathcal{X}$, a training set $(x_1, y_1), \ldots, (x_m, y_m) \in \mathcal{X} \times \mathbb{R}$, a strictly monotonic increasing real-valued function Ω on $[0, \infty[$, and an arbitrary cost function $c : (\mathcal{X} \times \mathbb{R}^2)^m \to \mathbb{R} \cup \{\infty\}$*

Any $f \in \mathcal{F}$ minimizing the regularized risk functional

$$c\left((x_1, y_1, f(x_1)), \ldots, (x_m, y_m, f(x_m))\right) + \Omega\left(\|f\|\right) \qquad (3)$$

admits a representation of the form

$$f(.) = \sum_{i=1}^{m} \alpha_i k(x_i, .).$$

More on Kernels

Mercer's Theorem

If k is a continuous kernel of a positive definite integral operator on $L_2(\mathcal{X})$ (where $\mathcal{X}$ is some compact space),

$$\int_{\mathcal{X}} k(x, x') f(x) f(x') \, dx \, dx' \geq 0,$$

it can be expanded as

$$k(x, x') = \sum_{i=1}^{\infty} \lambda_i \psi_i(x) \psi_i(x')$$

using eigenfunctions ψ_i and eigenvalues $\lambda_i \geq 0$ [41].

The Mercer Feature Map

In that case

$$\Phi(x) := \begin{pmatrix} \sqrt{\lambda_1}\psi_1(x) \\ \sqrt{\lambda_2}\psi_2(x) \\ \vdots \end{pmatrix}$$

satisfies $\langle \Phi(x), \Phi(x') \rangle = k(x, x')$.

Proof:

$$\langle \Phi(x), \Phi(x') \rangle = \left\langle \begin{pmatrix} \sqrt{\lambda_1}\psi_1(x) \\ \sqrt{\lambda_2}\psi_2(x) \\ \vdots \end{pmatrix}, \begin{pmatrix} \sqrt{\lambda_1}\psi_1(x') \\ \sqrt{\lambda_2}\psi_2(x') \\ \vdots \end{pmatrix} \right\rangle$$

$$= \sum_{i=1}^{\infty} \lambda_i \psi_i(x) \psi_i(x') = k(x, x')$$

Positive Definite Kernels

It can be shown that (modulo some details) the admissible class of kernels coincides with the one of positive definite (pd) kernels: kernels which are symmetric, and for

- any set of training points $x_1, \dots, x_m \in \mathcal{X}$ and
- any $a_1, \dots, a_m \in \mathbb{R}$

satisfy

$$\sum_{i,j} a_i a_j K_{ij} \geq 0, \quad \text{where } K_{ij} := k(x_i, x_j).$$

Elementary Properties of PD Kernels

Kernels from Feature Maps.
If Φ maps $\mathcal{X}$ into a dot product space $\mathcal{H}$, then $\langle \Phi(x), \Phi(x') \rangle$ is a pd kernel on $\mathcal{X} \times \mathcal{X}$.

Positivity on the Diagonal.
$k(x,x) \geq 0$ for all $x \in \mathcal{X}$

Cauchy-Schwarz Inequality.
$k(x,x')^2 \leq k(x,x)k(x',x')$ (Hint: compute the determinant of the Gram matrix)

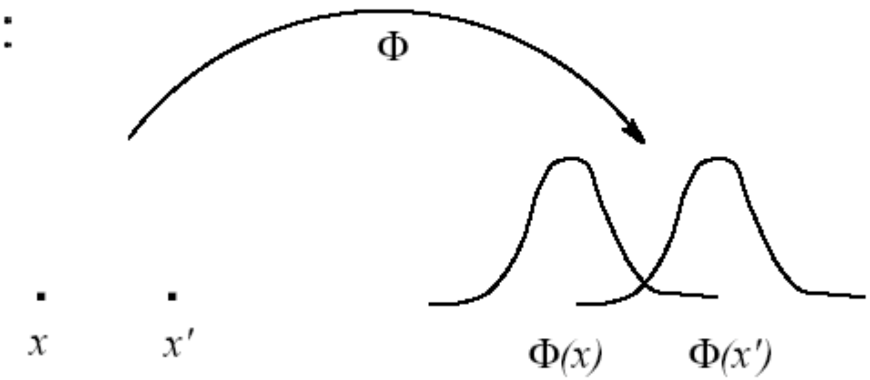
Vanishing Diagonals.
$k(x,x) = 0$ for all $x \in \mathcal{X} \Longrightarrow k(x,x') = 0$ for all $x, x' \in \mathcal{X}$

The Feature Space for PD Kernels [4, 1, 48]

- define a feature map

$$\Phi : \mathcal{X} \to \mathbb{R}^{\mathcal{X}}$$
$$x \mapsto k(.,x).$$

E.g., for the Gaussian kernel:



Next steps:

- turn $\Phi(\mathcal{X})$ into a linear space
- endow it with a dot product satisfying $\langle k(.,x_i), k(.,x_j) \rangle = k(x_i, x_j)$
- complete the space to get a *reproducing kernel Hilbert space*

Endow it With a Dot Product

$$\langle f, g \rangle := \sum_{i=1}^{m} \sum_{j=1}^{m'} \alpha_i \beta_j k(x_i, x'_j)$$

$$= \sum_{i=1}^{m} \alpha_i g(x_i) = \sum_{j=1}^{m'} \beta_j f(x'_j)$$

- This is well-defined, symmetric, and bilinear.
- It can be shown that it is also strictly positive definite (hence it is a dot product).
- Complete the space in the corresponding norm to get a Hilbert space $\mathcal{H}_k$.

The Reproducing Kernel Property

Two special cases:

- Assume

$$f(.) = k(., x).$$

  In this case, we have

$$\langle k(., x), g \rangle = g(x).$$

- If moreover

$$g(.) = k(., x'),$$

  we have the kernel trick

$$\langle k(., x), k(., x') \rangle = k(x, x').$$

k is called a *reproducing kernel* for $\mathcal{H}_k$.

Turn it Into a Linear Space

Form linear combinations

$$f(.) = \sum_{i=1}^{m} \alpha_i k(., x_i),$$

$$g(.) = \sum_{j=1}^{m'} \beta_j k(., x'_j)$$

$(m, m' \in \mathbb{N},\ \alpha_i, \beta_j \in \mathbb{R},\ x_i, x'_j \in \mathcal{X})$.

The Reproducing Kernel Property

Two special cases:

- Assume

$$f(.) = k(., x).$$

  In this case, we have

$$\langle k(., x), g \rangle = g(x).$$

- If moreover

$$g(.) = k(., x'),$$

  we have the kernel trick

$$\langle k(., x), k(., x') \rangle = k(x, x').$$

k is called a *reproducing kernel* for $\mathcal{H}_k$.

Kernels

Recall that the dot product has to satisfy

$$\langle k(x,.), k(x',.)\rangle = k(x,x').$$

For a Mercer kernel

$$k(x,x') = \sum_{j=1}^{N_F} \lambda_j \psi_j(x)\psi_j(x')$$

(with $\lambda_i > 0$ for all i, $N_F \in \mathbb{N} \cup \{\infty\}$, and $\langle \psi_i, \psi_j \rangle_{L_2(\mathcal{X})} = \delta_{ij}$), this can be achieved by choosing $\langle .,. \rangle$ such that

$$\langle \psi_i, \psi_j \rangle = \delta_{ij}/\lambda_i.$$

ctd.

To see this, compute

$$\begin{aligned}
\langle k(x,.), k(x',.)\rangle &= \left\langle \sum_i \lambda_i \psi_i(x)\psi_i, \sum_j \lambda_j \psi_j(x')\psi_j \right\rangle \\
&= \sum_{i,j} \lambda_i \lambda_j \psi_i(x)\psi_j(x')\langle \psi_i, \psi_j\rangle \\
&= \sum_{i,j} \lambda_i \lambda_j \psi_i(x)\psi_j(x')\delta_{ij}/\lambda_i \\
&= \sum_i \lambda_i \psi_i(x)\psi_i(x') \\
&= k(x,x').
\end{aligned}$$

Some Properties of Kernels [53]

If $k_1, k_2, \ldots$ are pd kernels, then so are

- αk_1, provided $\alpha \geq 0$
- $k_1 + k_2$
- $k_1 \cdot k_2$
- $k(x, x') := \lim_{n\to\infty} k_n(x, x')$, provided it exists
- $k(A, B) := \sum_{x\in A, x'\in B} k_1(x, x')$, where A, B are finite subsets of $\mathcal{X}$
  (using the feature map $\tilde{\Phi}(A) := \sum_{x\in A} \Phi(x)$)

Further operations to construct kernels from kernels: tensor products, direct sums, convolutions [28].

Computing Distances in Feature Spaces

Clearly, if k is positive definite, then there exists a map Φ such that

$$\|\Phi(x) - \Phi(x')\|^2 = k(x,x) + k(x',x') - 2k(x,x')$$

(it is the usual feature map).

This embedding is referred to as a *Hilbert space representation* as a distance. It turns out that this works for a larger class of kernels, called *conditionally positive definite*.

In fact, all algorithms that are translationally invariant (i.e. independent of the choice of the origin) in $\mathcal{H}$ work with cpd kernels [53].