Advanced Machine Learning
Lecture 18

Support Vector Regression & Co.

16.01.2013

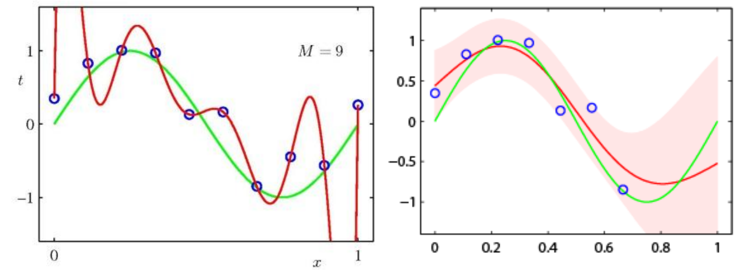
Bastian Leibe
RWTH Aachen
http://www.vision.rwth-aachen.de/

leibe@vision.rwth-aachen.de

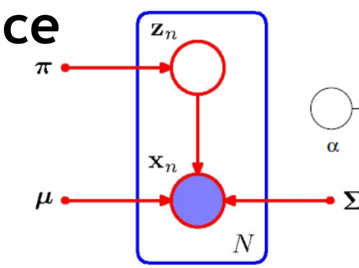
This Lecture: *Advanced Machine Learning*

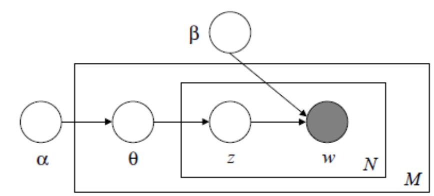
- Regression Approaches
  - Linear Regression
  - Regularization (Ridge, Lasso)
  - Kernels (Kernel Ridge Regression)
  - Gaussian Processes

$$f : \mathcal{X} \to \mathbb{R}$$



- Bayesian Estimation & Bayesian Non-Parametrics
  - Prob. Distributions, Approx. Inference
  - Mixture Models & EM
  - Dirichlet Processes
  - Latent Factor Models
  - Beta Processes





- SVMs and Structured Output Learning
  - SVMs, SVDD, SV Regression
  - Large-margin Learning

$$f : \mathcal{X} \to \mathcal{Y}$$

Topics of This Lecture

- Recap: Support Vector Machines
  - Discussion & Analysis
- Other Kernel Methods
  - Kernel PCA
  - Kernel k-Means Clustering
- Support Vector Data Description (1-class SVMs)
  - Motivation
  - Definition
  - Applications
- Support Vector Regression
  - Error function
  - Primal form
  - Dual form

Recap: SVM - Analysis

- **Traditional soft-margin formulation**

$$\min_{\mathbf{w}\in\mathbb{R}^D,\, \xi_n\in\mathbb{R}^+} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\xi_n$$

"Maximize the margin"

subject to the constraints

$$t_n y(\mathbf{x}_n) \geq 1 - \xi_n$$

"Most points should be on the correct side of the margin"

- **Different way of looking at it**
  - We can reformulate the constraints into the objective function.

$$\min_{\mathbf{w}\in\mathbb{R}^D} \underbrace{\frac{1}{2}\|\mathbf{w}\|^2}_{L_2 \text{ regularizer}} + \underbrace{C\sum_{n=1}^{N}[1 - t_n y(\mathbf{x}_n)]_+}_{\text{"Hinge loss"}}$$

where $[x]_+ := \max\{0,x\}$.

Slide adapted from Christoph Lampert

Recap: SVM – Discussion

- SVM optimization function

$$\min_{\mathbf{w}\in\mathbb{R}^D} \underbrace{\frac{1}{2}\|\mathbf{w}\|^2}_{\text{L}_2 \text{ regularizer}} + \underbrace{C\sum_{n=1}^{N}[1 - t_n y(\mathbf{x}_n)]_+}_{\text{Hinge loss}}$$

- Hinge loss enforces sparsity
  - Only a subset of training data points actually influences the decision boundary.
  - This is different from sparsity obtained through a regularizer! There, only a subset of input dimensions are used.
  - Unconstrained optimization, but non-differentiable function.
  - Solve, e.g. by *subgradient descent*
  - Currently most efficient: *stochastic gradient descent*

Outline of the Remaining Lectures

- *We will generalize the SVM idea in several directions...*

- Other Kernel methods
  - Kernel PCA
  - Kernel k-Means

- Other Large-Margin Learning formulations
  - Support Vector Data Description (one-class SVMs)
  - Support Vector Regression

- Structured Output Learning
  - General loss functions
  - General structured outputs
  - Structured Output SVM
  - Example: Multiclass SVM

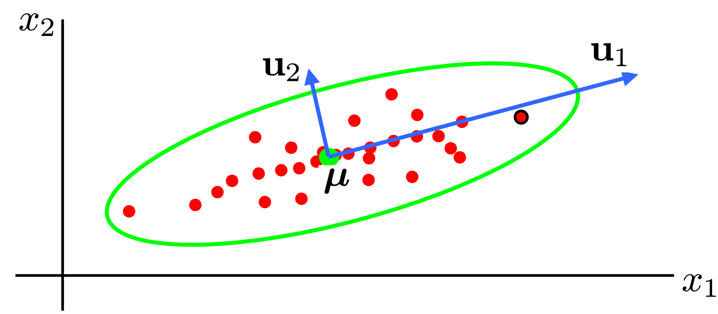
Topics of This Lecture

- Recap: Support Vector Machines
  - Discussion & Analysis
- Other Kernel Methods
  - Kernel PCA
  - Kernel k-Means Clustering
- Support Vector Data Description (1-class SVMs)
  - Motivation
  - Definition
  - Applications
- Support Vector Regression
  - Error function
  - Primal form
  - Dual form

Recap: PCA

- PCA procedure
  - Given samples $\mathbf{x}_n \in \mathbb{R}^d$, PCA finds the directions of maximal covariance. Without loss of generality assume that $\sum_n \mathbf{x}_n = \mathbf{0}$.
  - The PCA directions $\mathbf{e}_1,\ldots,\mathbf{e}_d$ are the **eigenvectors of the covariance matrix**

$$C = \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}_n \mathbf{x}_n^T$$



sorted by their eigenvalue.

  - We can express $\mathbf{x}_n$ in PCA space by $F(\mathbf{x}_n) = \sum_{k=1}^{K} \langle \mathbf{x}_n, \mathbf{e}_k \rangle \mathbf{e}_k$
  - Lower-dim. coordinate mapping: $\mathbf{x}_n \mapsto \begin{pmatrix} \langle \mathbf{x}_n, \mathbf{e}_1 \rangle \\ \langle \mathbf{x}_n, \mathbf{e}_2 \rangle \\ \ldots \\ \langle \mathbf{x}_n, \mathbf{e}_K \rangle \end{pmatrix} \in \mathbb{R}^K$

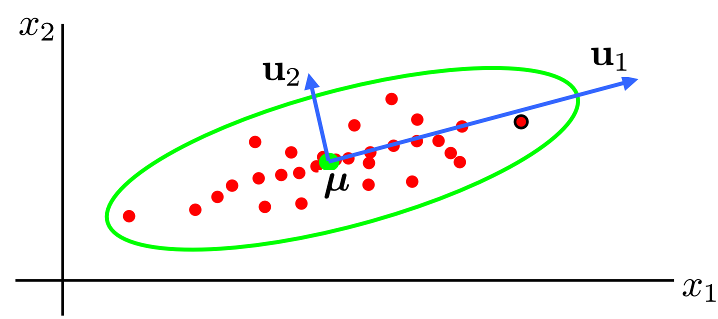
Slide credit: Christoph Lampert

Kernel-PCA

- **Kernel-PCA procedure**
  - Given samples $\mathbf{x}_n \in \mathcal{X}$, kernel $\mathcal{X} \times \mathcal{X} \to \mathbb{R}$ with an implicit feature map ϕ: $\mathcal{X} \to \mathcal{H}$. Perform PCA in the Hilbert space $\mathcal{H}$.
  - The kernel-PCA directions $\mathbf{e}_1,\ldots,\mathbf{e}_d$ are the eigenvectors of the covariance operator

$$C = \frac{1}{N}\sum_{n=1}^{N} \phi(\mathbf{x}_n)\phi(\mathbf{x}_n)^T$$

sorted by their eigenvalue.

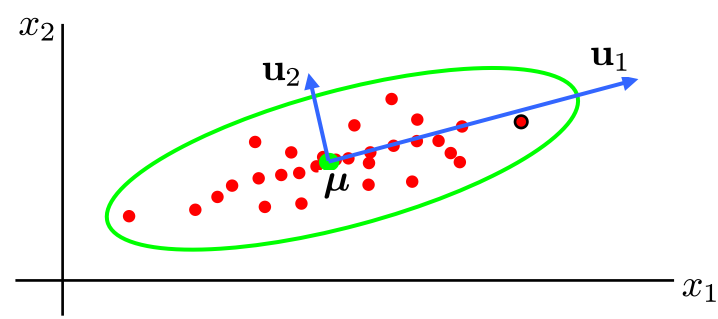


  - Lower-dim. coordinate mapping: $\mathbf{x}_n \mapsto \begin{pmatrix} \langle \phi(\mathbf{x}_n), \mathbf{e}_1 \rangle \\ \langle \phi(\mathbf{x}_n), \mathbf{e}_2 \rangle \\ \ldots \\ \langle \phi(\mathbf{x}_n), \mathbf{e}_K \rangle \end{pmatrix} \in \mathbb{R}^K$

Slide credit: Christoph Lampert

Kernel-PCA

- **Kernel-PCA procedure**
  - **Given samples $\mathbf{x}_n \in \mathcal{X}$, kernel $\mathcal{X} \times \mathcal{X} \to \mathbb{R}$ with an implicit feature map ϕ: $\mathcal{X} \to \mathcal{H}$. Perform PCA in the Hilbert space $\mathcal{H}$.**
  - **Equivalently, we can use the eigenvectors $\mathbf{e}'_k$ and eigenvalues λ_k of the kernel matrix**



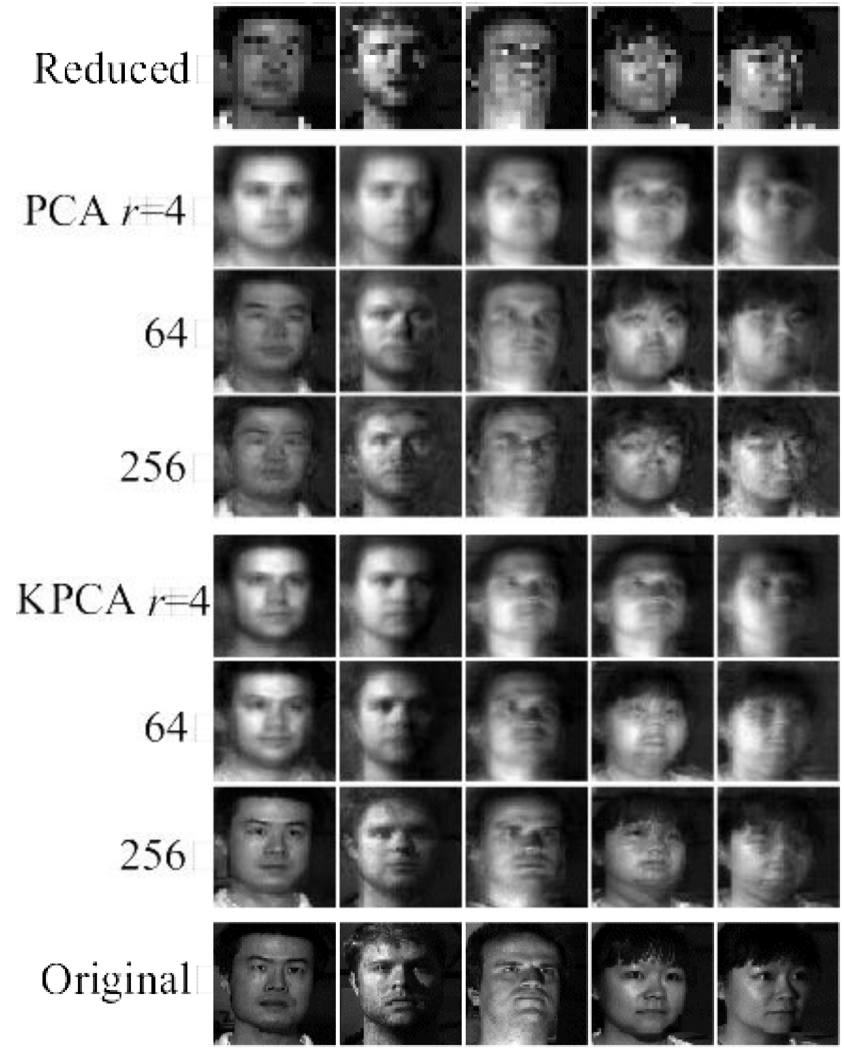
$$K = (\langle \phi(\mathbf{x}_m), \phi(\mathbf{x}_n) \rangle)_{m,n=1,\ldots,N}$$
$$= (k(\mathbf{x}_m, \mathbf{x}_n))_{m,n=1,\ldots,N}$$

  - **Coordinate mapping:**

$$\mathbf{x}_n \mapsto (\sqrt{\lambda_1}\mathbf{e}_1^{'}, \ldots, \sqrt{\lambda_K}\mathbf{e}_K^{'})$$

Example: Image Superresolution

- Training procedure
  - Collect high-res face images
  - Use KPCA with RBF-kernel to learn non-linear subspaces
- For new low-res image:
  - Scale to target high resolution
  - Project to closest point in face subspace



Reconstruction in r dimensions

Kim, Franz, Schölkopf, Iterative Kernel Principal Component Analysis for Image Modelling, IEEE Trans. PAMI, Vol. 27(9), 2005.

Kernel k-Means Clustering

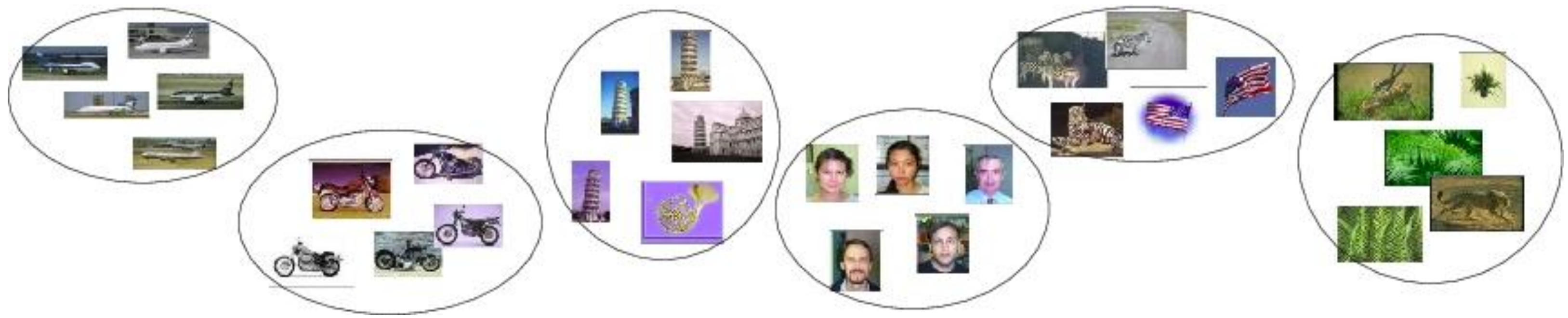
- Kernel PCA is more than just non-linear versions of PCA
  - PCA maps $\mathbb{R}^d$ to $\mathbb{R}^{d'}$, e.g., to remove noise dimensions.
  - Kernel-PCA maps $\mathcal{X} \rightarrow \mathbb{R}^{d'}$, so it provides a vectorial representation of non-vectorial data.

  $\Rightarrow$ We can apply algorithms that only work in vector spaces to data that is not in a vector representation.

- Example: k-Means clustering
  - Given $\mathbf{x}_1,\ldots,\mathbf{x}_n \in \mathcal{X}$.
  - Choose a kernel function $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$.
  - Apply kernel-PCA to obtain vectorial $\mathbf{v}_1,\ldots,\mathbf{v}_n \in \mathbb{R}^{d'}$.
  - Cluster $\mathbf{v}_1,\ldots,\mathbf{v}_n \in \mathbb{R}^{d'}$ using *K*-Means.

  $\Rightarrow$ $\mathbf{x}_1,\ldots,\mathbf{x}_n$ are clustered based on the similarity defined by k.

Example: Unsupervised Object Categorization



- **Automatically group images that show similar objects**
  - Represent images by bag-of-word histograms
  - Perform Kernel k-Means Clustering

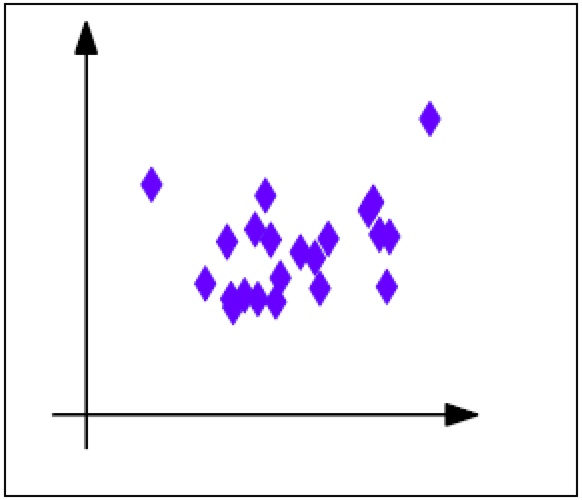
  ⇒ Observation: Clusters get better if we use a good image kernel (e.g., χ^2) instead of plain k-Means (linear kernel).

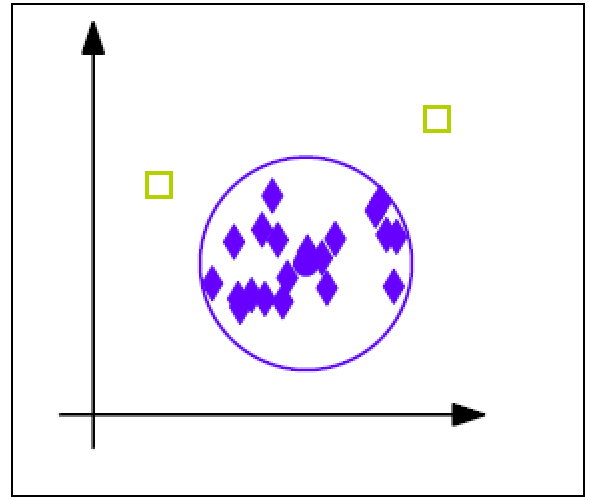
T. Tuytelaars, C. Lampert, M. Blaschko, W. Buntine, Unsupervised object discovery: a comparison, IJCV, 2009.]

Topics of This Lecture

- Recap: Support Vector Machines
  - Discussion & Analysis
- Other Kernel Methods
  - Kernel PCA
  - Kernel k-Means Clustering
- **Support Vector Data Description (1-class SVMs)**
  - **Motivation**
  - **Definition**
  - **Applications**
- Support Vector Regression
  - Error function
  - Primal form
  - Dual form

One-Class SVMs





- **Motivation**
  - For unlabeled data, we are interested in detecting outliers, i.e. samples that lie far away from most of the other samples.
- **Problem statement**
  - For samples $\mathbf{x}_1,\ldots,\mathbf{x}_N$, find the smallest ball (center $\mathbf{c}$, radius R) that contains "most" of the samples.
  - "Most" again means that we allow some points to have slack.

One-Class SVMs

- **Formalization**
  - **Solve**

$$\min_{R\in\mathbb{R},\, \mathbf{c}\in\mathbb{R}^D,\, \xi_n\in\mathbb{R}^+} R + \frac{1}{\nu N}\sum_{n=1}^{N}\xi_n$$

subject to

$$\|\mathbf{x}_n - \mathbf{c}\|^2 \le R^2 + \xi_n \qquad \textbf{for} \quad n = 1, ..., N$$

where $\nu \in (0,1)$ **upper bounds the number of outliers.**

One-Class SVMs

- Again apply the kernel trick
  - Use a kernel $k\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ with an implicit feature map $\phi\colon \mathcal{X} \to \mathcal{H}$.
  - Do outlier detection for $\phi(\mathbf{x}_1),\ldots,\phi(\mathbf{x}_N)$:
  - Find the smallest ball (center $\mathbf{c} \in \mathcal{H}$, radius R) that contains "most" of the samples.
  - Solve

$$\min_{R\in\mathbb{R},\, \mathbf{c}\in\mathcal{H},\, \xi_n\in\mathbb{R}^+} R + \frac{1}{\nu N}\sum_{n=1}^{N}\xi_n$$

subject to

$$\|\phi(\mathbf{x}_n) - \mathbf{c}\|^2 \leq R^2 + \xi_n \qquad \text{for} \quad n = 1, \ldots, N$$

One-Class SVM

- **Solution**
  - The **representer theorem** states that we can write the solution only in terms of the kernel $k(\mathbf{x}_n,\mathbf{x}_m)$ as

$$\mathbf{c} = \sum_{n=1}^{N} a_n \phi(\mathbf{x}_n)$$

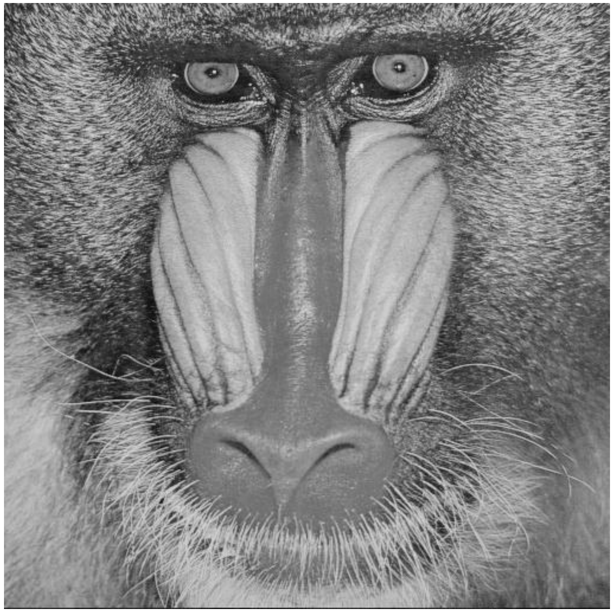
  - where again we know from the KKT conditions that for each point $\mathbf{x}_n$, either the constraint is active (i.e., the point is on the circle R) or the Lagrange multiplier $a_n = 0$.

⇒ Sparse solution, depends only on few data points, the support vectors.

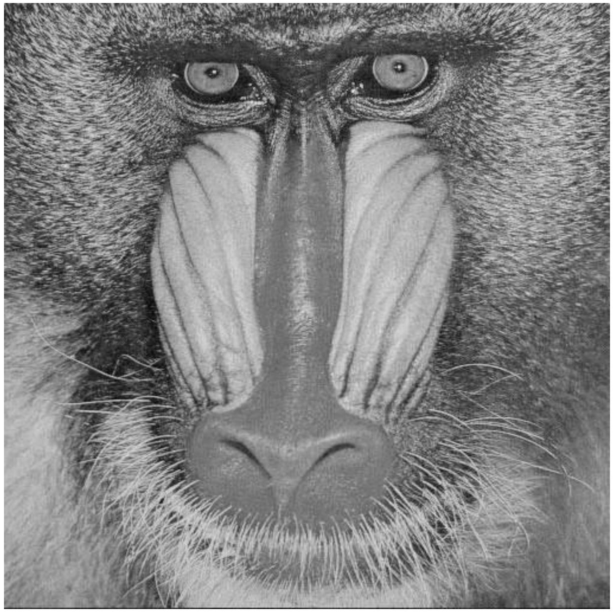
  - Because of this, the formulation is called **Support Vector Data Description** (SVDD) or one-class SVM.

⇒ Often used for outlier/anomaly detection.

Example: Steganalysis



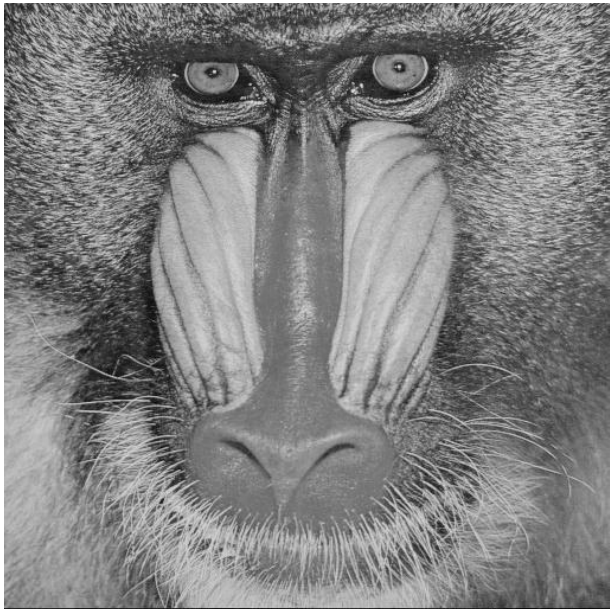
Original



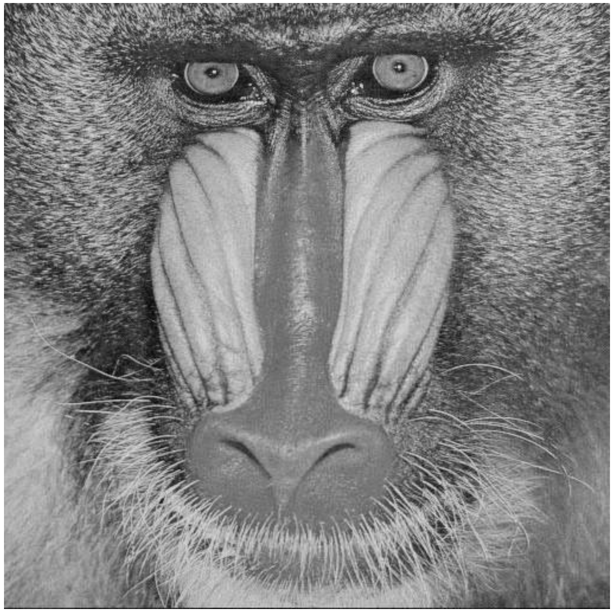
With 23’300 hidden bits

- Steganography
  - Hide data in other data (e.g. in images)
  - E.g., flip some least significant bits
- Steganalysis
  - Given any data, find out if some data is hidden

Example: Steganalysis



Original



With 23'300 hidden bits

- Possible procedure
  - Compute image statistics (color wavelet coefficients)
  - Train SVDD with RBF-kernel
  - Identified outlier images are suspicious candidates

S. Lyu, H. Farid. Steganalysis using color wavelet statistics and one-class support vector machines, SPIE EI, 2004

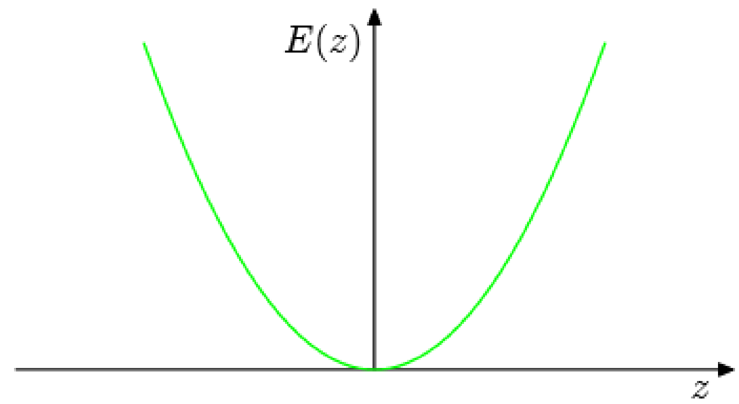
Topics of This Lecture

- Recap: Support Vector Machines
  - Discussion & Analysis
- Other Kernel Methods
  - Kernel PCA
  - Kernel k-Means Clustering
- Support Vector Data Description (1-class SVMs)
  - Motivation
  - Definition
  - Applications
- **Support Vector Regression**
  - **Error function**
  - **Primal form**
  - **Dual form**

SVMs for Regression

- **Linear regression**
  - Minimize a regularized quadratic error function

$$\frac{1}{2}\sum_{n=1}^{N}\{y_n - t_n\}^2 + \frac{\lambda}{2}||\mathbf{w}||^2$$



- **Problem**
  - Sensitive to outliers, because the quadratic error function penalizes large residues.
  - This is the case even for (Kernel) Ridge Regression, although regularization helps.

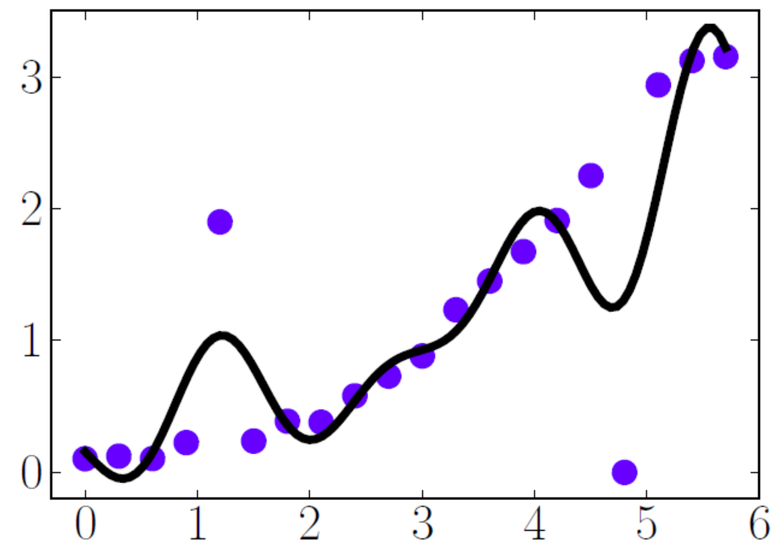
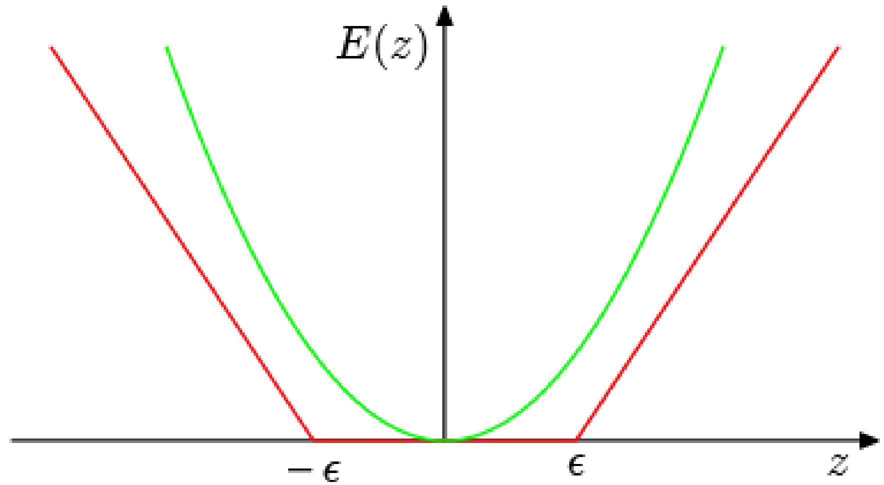


Image source: C.M. Bishop, C. Lampert

SVMs for Regression



- **Obtaining sparse solutions**
  - **Define an ϵ-insensitive error function**

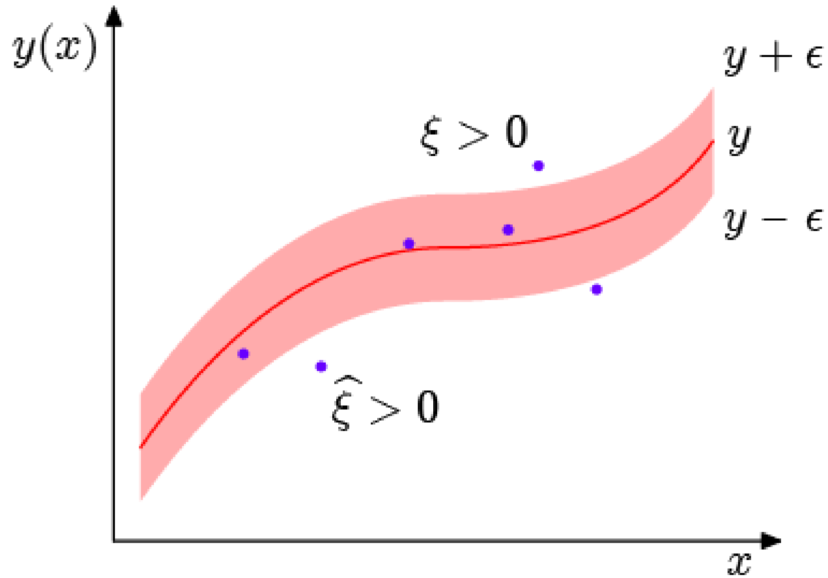
$$E_\epsilon(y(\mathbf{x}) - t) = \begin{cases} 0, & \text{if } |y(\mathbf{x}) - t| < \epsilon \\ |y(\mathbf{x} - t| - \epsilon, & \text{otherwise} \end{cases}$$

  - **and minimize the following regularized function**

$$C \sum_{n=1}^{N} E_\epsilon(y_n - t_n) + \frac{1}{2}||\mathbf{w}||^2$$

Image source: C.M. Bishop

Dealing with Noise and Outliers



- **Introduce slack variables**
  - **We now need two slack variables $\xi_n \geq 0$ and $\widehat{\xi}_n \geq 0$.**
  - **A target point lies in the ϵ-tube if $y_n - \epsilon \leq t_n \leq y_n + \epsilon$.**
  - **The corresponding conditions are**

$$t_n \leq y(\mathbf{x}_n) + \epsilon + \xi_n$$
$$t_n \geq y(\mathbf{x}_n) - \epsilon - \widehat{\xi}_n$$

Image source: C.M. Bishop

Dealing with Noise and Outliers

- **Optimization with slack variables**
  - The error function can then be rewritten as

$$C \sum_{n=1}^{N} [|y(\mathbf{x}_n) - t_n| - \epsilon]_+ + \frac{1}{2}||\mathbf{w}||^2$$

  - Using the conditions for the slack variables, we obtain

$$\begin{aligned} t_n &\leq y(\mathbf{x}_n) + \epsilon + \xi_n \\ t_n &\geq y(\mathbf{x}_n) - \epsilon - \widehat{\xi}_n \end{aligned} \quad \Rightarrow \quad \begin{aligned} \xi_n &\geq -(y(\mathbf{x}_n) - t_n) - \epsilon \\ \widehat{\xi}_n &\geq (y(\mathbf{x}_n) - t_n) - \epsilon \end{aligned}$$

  - And thus

$$C \sum_{n=1}^{N} (\xi_n + \widehat{\xi}_n) + \frac{1}{2}||\mathbf{w}||^2 \qquad \begin{aligned} \xi_n &\geq 0 \\ \widehat{\xi}_n &\geq 0 \end{aligned}$$

Support Vector Regression - Primal Form

- **Lagrangian primal form**

$$L_p = C\sum_{n=1}^{N}(\xi_n + \widehat{\xi}_n) + \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N}(\mu_n\xi_n + \widehat{\mu}_n\widehat{\xi}_n)$$

$$- \sum_{n=1}^{N} a_n\,(\epsilon + \xi_n + y_n - t_n) - \sum_{n=1}^{N}\widehat{a}_n\left(\epsilon + \widehat{\xi}_n - y_n + t_n\right)$$

- **Solving for the variables**

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \;\Rightarrow\; \boxed{\mathbf{w} = \sum_{n=1}^{N}(a_n - \widehat{a}_n)\phi(\mathbf{x}_n)}$$

$$\frac{\partial L}{\partial \xi_n} = 0 \;\Rightarrow\; \boxed{a_n + \mu_n = C}$$

$$\frac{\partial L}{\partial b} = 0 \;\Rightarrow\; \boxed{\sum_{n=1}^{N}(a_n - \widehat{a}_n) = 0}$$

$$\frac{\partial L}{\partial \widehat{\xi}_n} = 0 \;\Rightarrow\; \boxed{\widehat{a}_n + \widehat{\mu}_n = C}$$

Support Vector Regression - Dual Form

- From this, we can derive the dual form
  - Maximize

$$L_d(\mathbf{a}, \widehat{\mathbf{a}}) = -\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}(a_n - \widehat{a}_n)(a_m - \widehat{a}_m)k(\mathbf{x}_n, \mathbf{x}_m)$$

$$-\epsilon\sum_{n=1}^{N}(a_n + \widehat{a}_n) + \sum_{n=1}^{N}(a_n - \widehat{a}_n)t_n$$

  - under the conditions

$$0 \le a_n \le C$$

$$0 \le \widehat{a}_n \le C$$

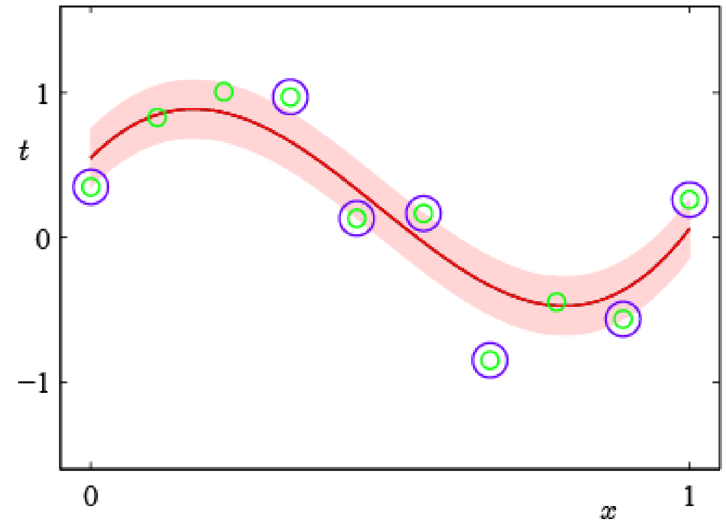
  - Predictions for new inputs are then made using

$$y(\mathbf{x}) = \sum_{n=1}^{N}(a_n - \widehat{a}_n)k(\mathbf{x}, \mathbf{x}_n) + b$$

KKT Conditions

- **KKT conditions**

$$
\begin{aligned}
a_n(\epsilon + \xi_n + y(\mathbf{x}_n) - t_n) &= 0 \\
\widehat{a}_n(\epsilon + \widehat{\xi}_n - y(\mathbf{x}_n) + t_n) &= 0 \\
(C - a_n)\xi_n &= 0 \\
(C - \widehat{a}_n)\widehat{\xi}_n &= 0
\end{aligned}
$$

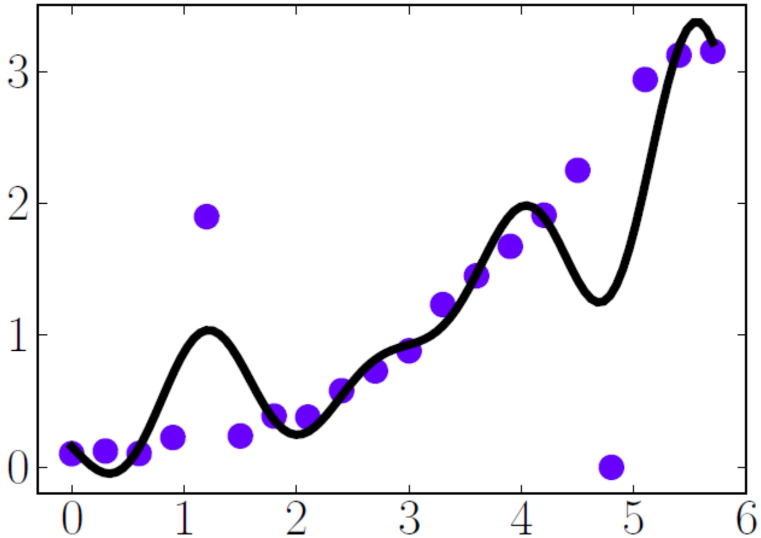


- **Observations**
  - A coefficient a_n can only be non-zero if the first constraint is active, i.e., if a point lies either on or above the ϵ-tube.
  - Similarly, a non-zero coefficient $\widehat{a}_n$ must be on/below the ϵ-tube
  - The first two constraints cannot both be active at the same time

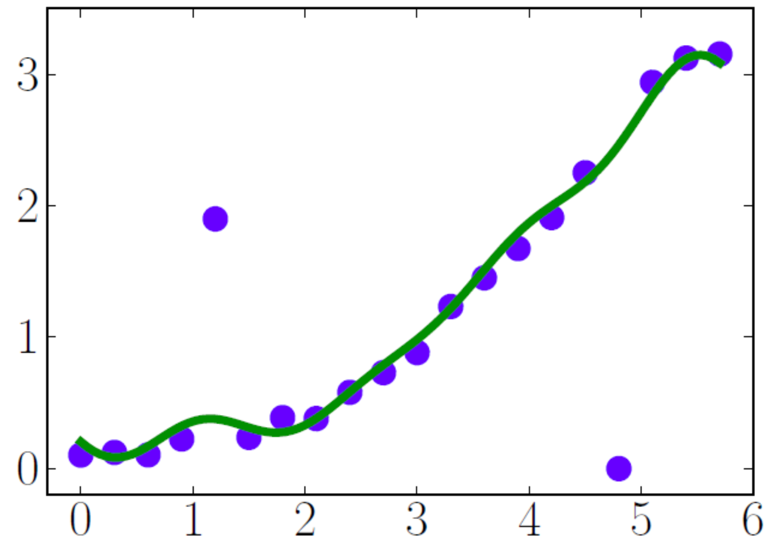
  $\Rightarrow$ Either a_n or $\widehat{a}_n$ or both must be zero.
  - The support vectors are those points for which $a_n \neq 0$ or $\widehat{a}_n \neq 0$, i.e., the points on the boundary of or outside the ϵ-tube.

Image source: C.M. Bishop

Discussion



Least-squares regression



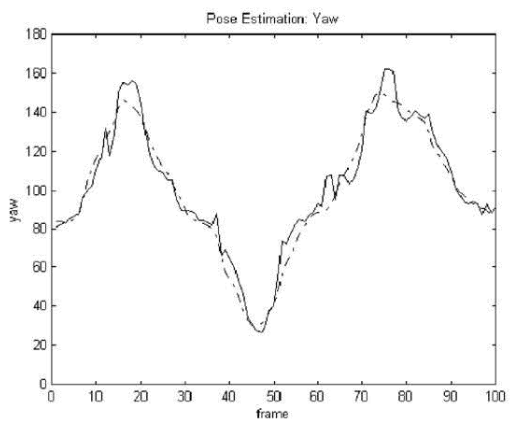
Support vector regression

- **Slightly different interpretation**
  - For SVMs, classification function depends only on SVs.
  - For SVR, support vectors mark outlier points. SVR tries to limit the effect of those outliers on the regression function.
  - Nevertheless, the prediction $y(\mathbf{x})$ only depends on the support vectors.

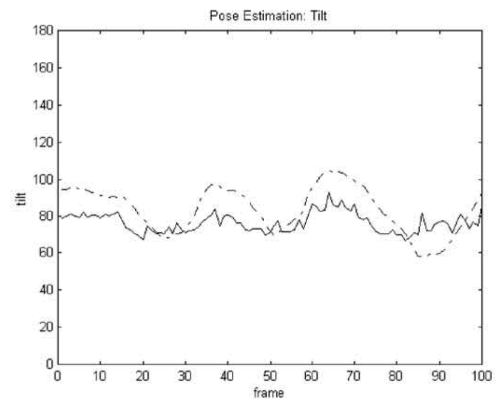
Image source: Christoph Lampert

Example: Head Pose Estimation

(a) Sample frames of a test sequence



(b) Yaw estimation



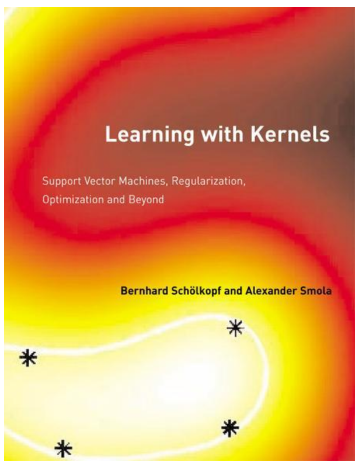
(c) Tilt estimation

- **Procedure**
  - **Detect faces in image**
  - **Compute gradient representation of face region**
  - **Train support vector regression for yaw, tilt (separately)**

Y. Li, S. Gong, J. Sherra, H. Liddell, Support vector machine based multi-view face detection and recognition, Image & Vision Computing, 2004.

References and Further Reading

- **More information on Kernel PCA can be found in Chapter 12.3 of Bishop's book. Support Vector Regression is described in Chapter 7.1. You can also look at Schölkopf & Smola (some chapters available online).**



Christopher M. Bishop
Pattern Recognition and Machine Learning
Springer, 2006

B. Schölkopf, A. Smola
Learning with Kernels
MIT Press, 2002
http://www.learning-with-kernels.org/