Kernel Machines

Rui Kuang

Department of Computer Science and Engineering
University of Minnesota
kuang@umn.edu

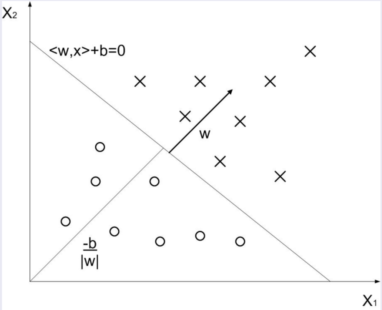
Spring 2020

Linear Classification

Linear classifier

- Linear separation of the input space X with a hyperplane $< w, x > +b = 0$.
- Prediction function is the linear function $f(x) =< w, x > +b$.
- Prediction is made with $sign(f(x))$.

Two dimensional illustration



Perceptron: A Linear Learning Algorithm

Algorithm (Rosenblatt, 1957)

- Learn a w such that $< w, x > +b = 0$ separates two classes.
- Ignore b as an additional dimension of w by adding a 1 to the xs.
- Initialize $k = 0$ and $w(0) = \overrightarrow{0}$
- Until converge (no mistake on a certain number of data points)

$$\begin{aligned}
&\text{for each example } (x^t, y^t): \\
&\qquad\qquad \text{if} \quad (< w(k), x^t >) * y^t \leq 0 \\
&\qquad\qquad\qquad\quad w(k+1) \leftarrow w(k) + y^t x^t \\
&\qquad\qquad\qquad\quad k \leftarrow k + 1
\end{aligned}$$

- Solution is a linear combination of training data $w = \sum \alpha^t y^t x^t$, $\alpha^t \geq 0$.

Perceptron: A Linear Learning Algorithm

Kernel Algorithm

- Learn a α such that $f(x) =< w, x > +b$, where $w = \sum_t \alpha^t y^t x^t$ separates two classes.
- Ignore b as an additional dimension of w by adding a 1 to the xs.
- Initialize $\alpha = \overrightarrow{0}$
- Until converge (no mistake on a certain number of data points)

  for each example (x^t, y^t) :

$$\text{if} \quad (\sum_s \alpha^s y^s < x^s, x^t >) * y^t \leq 0$$

$$\alpha^t = \alpha^t + 1$$

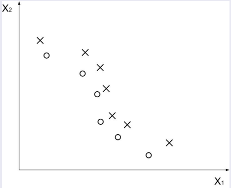
- The dual classifier is $f(x) = \sum_t \alpha^t y^t < x^t, x > +b$.

Duality and Non-linear Mapping

Limitations

- Hard datasets are often in very high dimensional feature space. Very inefficient to learn w.
- Most real-world datasets are non-linearly separable.

non-linearly separable data



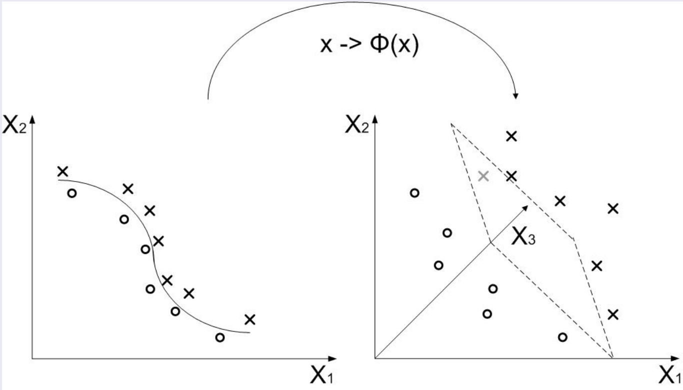
Duality

- The linear function can be rewritten in a dual representation: $f(x) =< w, x > +b = \sum \alpha^t y^t < x^t, x > +b$.
- Instead of relying on high dimensional w, we only need to know α^ts, which only depends on the dot products between x^ts.
- Duality allows us to introduce non-linear mapping $x \rightarrow \phi(x)$ from the old feature space to a new feature space.

Duality and Non-linear Mapping

Non-linear mapping for linearly non-separable data

Duality allows us to introduce non-linear mapping to a new feature space. Data points are linearly separable in the new space.



Kernels

Dual representation

$f(x) =< w, x > +b = \sum \alpha^t y^t < x^t, x > +b$

Dual representation with new mapping ϕ

$f_\phi(x) = \sum \alpha^t y^t < \phi(x^t), \phi(x) > +b$

Introduce kernel function K

$K(x^s, x^t) =< \phi(x^s), \phi(x^t) >$

Dual representation with kernel function K

$f_\phi(x) = \sum \alpha^t y^t < \phi(x^t), \phi(x) > +b = \sum \alpha^t y^t K(x^t, x) + b$

Kernel Methods

Implicit feature mapping with kernel function

- Training the classifier only depends on $K(x^t, x^s)$ of all pairs of examples.
- No need to explicitly define a ϕ, if K is a valid kernel function.
- Kernels can be defined to handle discrete and structured data such as graphs, strings and any other objects.

Kernel methods

- Kernel methods employ kernel functions to handle high dimensional or discrete and structured data.
- Kernel algorithms have dual representation in optimization problems.

Kernel Matrix (Gram Matrix)

Gram matrix

$K(x^1, x^1)$	$K(x^1, x^2)$	$K(x^1, x^3)$	...	$K(x^1, x^n)$
$K(x^2, x^1)$	$K(x^2, x^2)$	$K(x^2, x^3)$	...	$K(x^2, x^n)$
...	...	...	...	...
$K(x^n, x^1)$	$K(x^n, x^2)$	$K(x^n, x^3)$	...	$K(x^n, x^n)$

Gram matrix

- Kernel function computes pairwise similarity between examples.
- There exists a ϕ s.t. $K(x, z) =< \phi(x), \phi(z) > <=> K$ is positive semidefinite and symmetric.
- Eigenvalue expansion, $K(x, z) = \sum_t \lambda_t \phi_t(x)\phi_t(z)$ that is $\phi(x) = (\sqrt{\lambda_1}\phi_1(x), \sqrt{\lambda_2}\phi_2(x)...\sqrt{\lambda_3}\phi_n(x))$
- For any c, $c^T Kc = \sum_{s,t} c^t c^s (x^t)^T x^s = ||\sum_t c^t x^t||^2 \geq 0$

Kernels: Positive Definite and Symmetric

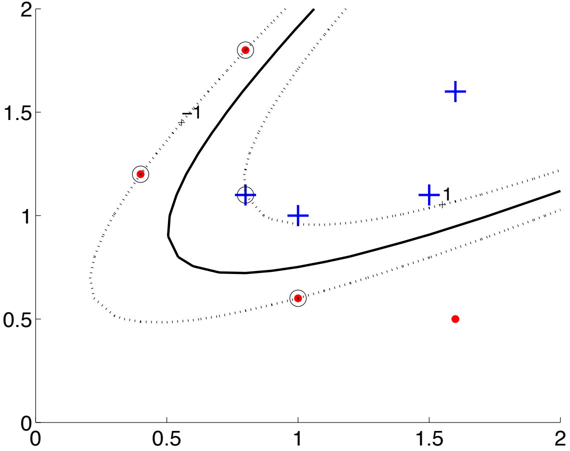
Examples

- Polynomial kernel: $K(x,z) =< x, z >^p + c$
- RBF kernel: $K(x,z) = e^{-\|x-z\|^2/2\sigma}$
- Sigmoid kernel: $K(x,z) = 1/(1 + e^{\kappa < x,z > -\delta})$
- Linear combination of kernels: $K(x,z) = \sum_t c^t K^t(x,z)$

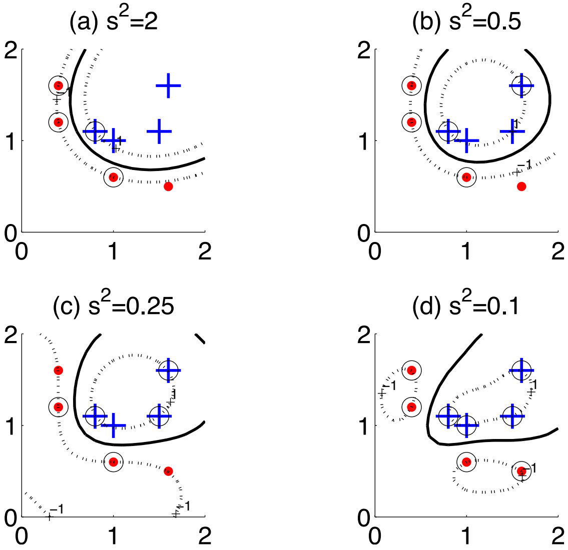
Proof: Show a function is a kernel

- Mapping:
  $< x, y >^2 =< (x_1, x_2), (y_1, y_2) >^2 = x_1^2 y_1^2 + x_2^2 y_2^2 + 2x_1 y_1 x_2 y_2$
  $=< (x_1^2, x_2^2, \sqrt{2}x_1 x_2), (y_1^2, y_2^2, \sqrt{2}y_1 y_2) >$
  $=< \phi(x), \phi(y) >$, where $\phi((x_1, x_2)) = (x_1^2, x_2^2, \sqrt{2}x_1 x_2)$.
- Mercer's theorem (Vapnik, 1995):
  $\int K(x,y)g(x)g(y)dxdy \geq 0$ for any $g(x)$ with finite $\int g(x)^2 dx$.

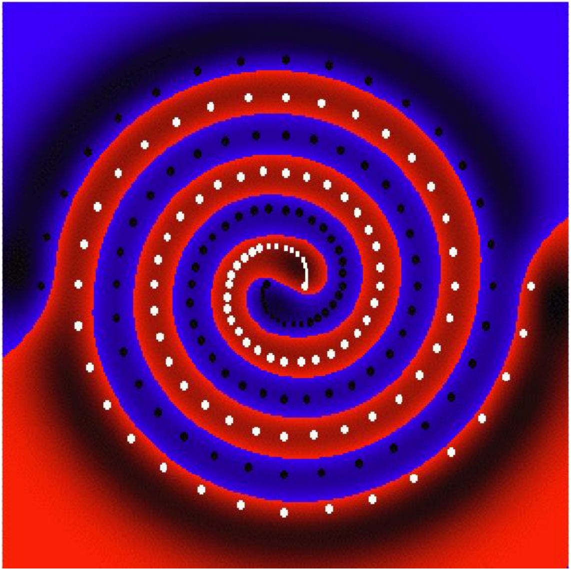
Non-linear Decision Boundary of Polynomial Kernel



Non-linear Decision Boundary of Gaussian Kernel

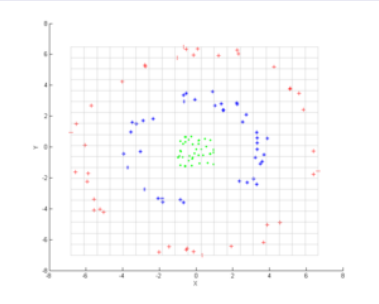


Separate Two Spirals with Gaussian Kernel

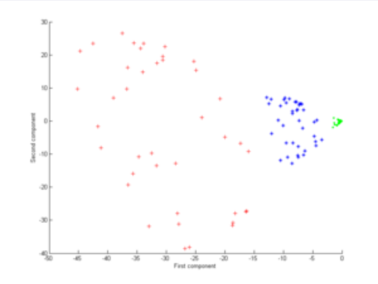


Kernel PCA

Original Data



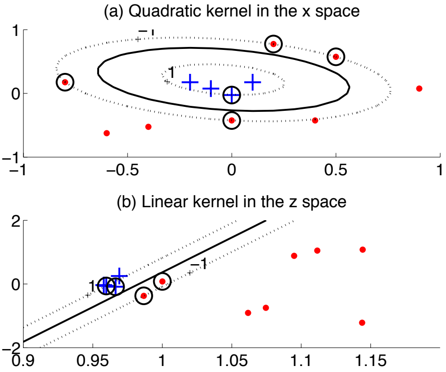
polynomial kernel



Kernel PCA

Formulation

- Introduce a mapping $\phi(x)$ and $C = \frac{1}{N}\sum_{t=1}^{N} \phi(x^t)\phi(x^t)^T$
- Find eigen-direction w, where $w = \sum_{t=1}^{N} \alpha^t \phi(x^t)$
- Solve
  $\lambda w = Cw \Rightarrow \lambda \sum_{t=1}^{N} \alpha^t \phi(x^t) = \frac{1}{N}\sum_{t=1,s=1}^{N} \alpha^t \phi(x^s)(\phi(x^s)^T \phi(x^t))$
- $\lambda \sum_{t=1}^{N} \alpha^t (\phi(x^l)^T \phi(x^t)) =$
  $\frac{1}{N}\sum_{t=1,s=1}^{N} \alpha^t (\phi(x_l)^T \phi(x^s))(\phi(x^s)^T \phi(x^t)), \forall l$
- Let $K_{ts} =< \phi(x^t), \phi(x^s) >$
- $n\lambda K\alpha = K^2\alpha \Leftrightarrow n\lambda\alpha = K\alpha$
- Eigen-direction: $w = \sum_{t=1}^{N} \alpha^t \phi(x^t)$
- Projection $w^T * \phi(x) = \sum_{t=1}^{N} \alpha^t \phi(x^t)^T * \phi(x) = \sum_{t=1}^{N} \alpha^t K(x^t, x)$

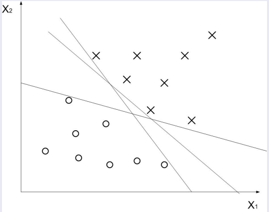


Comments:

- Kernel PCA introduce nonlinear mapping of the data.
- The same assumptions in the mapped feature space: linear rotation, mean and covariance, etc.
- Doesn't require explicit feature mapping.

From Perceptron to Support Vector Machines

Large margin based learning

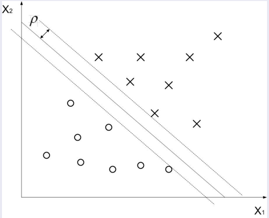


Generalized Learning

- There are many hyperplanes that can separate the data. Arbitrary choose may overfit the data.
- Need theoretical principle to choose one that generalize to the test data best.
- Support Vector Machines find the hyperplane with the largest margin.
- The large margin principle gives a bound on true error.

From Perceptron to Support Vector Machines

Large margin based learning



Generalized Learning

- There are many hyperplanes that can separate the data. Arbitrary choice may overfit the data.
- Need theoretical principle to choose one that generalize the data distribution best.
- Support Vector Machines find the hyperplane with the largest margin.
- The large margin principle gives a bound on true error.

Consistency of Empirical Risk Minimization

Definition

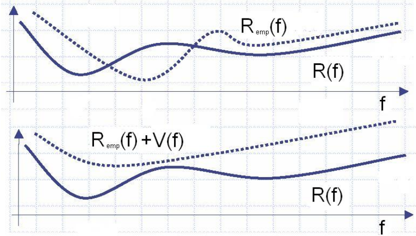
- Loss function $L(x, y, f(x))$ measures error.
- True Risk: $R(f) = \int L(x, y, f)P(x, y)dxdy$.
- Empirical Risk: $R_{emp}(f) = \frac{1}{N}\sum_t L(y^t, x^t, f)$.

Empirical Risk Minimization

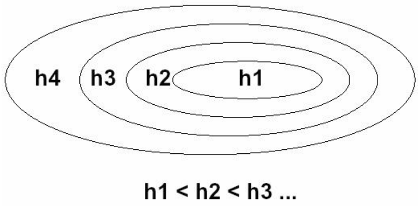
- True distribution $P(x, y)$ is not available.
- Consistency and convergence rate (not rigorous): $\lim_{N\to\infty} P\{\sup_f(R(f) - R_{emp}(f)) > \epsilon\} = 0$.
- Only have finite number N of observations; minimizing $R_{emp}(f)$ alone doesn't guarantee an approximation of minimizing $R(f)$.
- In general, minimizing empirical risk is inconsistent and can easily overfit data.

Structural Risk Minimization

- Introduce a regularizer $V(f)$ such that $R(f) \leq R_{emp}(f) + V(f)$.
- The VC dimension of f is not contiguous; to find the optimal f, introduce a "structure" on the function class {f}.
- Minimizing $R_{emp}(f) + V(f)$ in nested sets of functions is called Structural Risk Minimization.



Bound on real risk



Structural Risk Minimization

VC Dimension

Theorem: VC bound

For the 0/1 loss function, given the number of data points N, with probability $1 - \eta$,

$$R(f) \leq R_{emp}(f) + \sqrt{\frac{v(log(2N/v) + 1) - log(\frac{\eta}{4})}{N}},$$

where v is the Vapnik-Chervonenkis (VC) dimension of $\{f\}$ (1970).

Properties

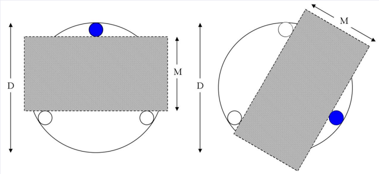
- This bound is a general bound independent of any $P(x, y)$.
- The higher the VC dimension is, the larger the VC confidence is.
- Structural risk minimization finds a balance between the empirical errors and function complexity.

VC Dimension and SVM

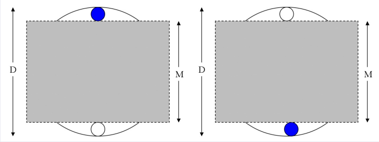
Large margin

- Pick arbitrary hyperplane is not helpful in the structural risk analysis.
- Instead, consider gap-tolerant classifiers, hypertubes with margin M.
- Data are considered to be in a sphere of diameter D.
- When we increase M, the VC dimension of the gap-tolerant classifiers drops.
- VC dimension of GT classifier: $v \leq min[ceil[\frac{D^2}{M^2}], d] + 1$.

Small M can shatter 3 points



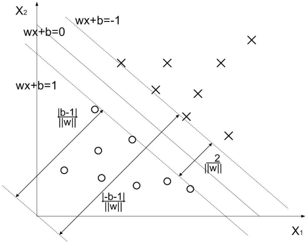
Large M can only shatter 2



The Separable Case: Primal Problem

Definition

Define, without loss of generality, hyperplane $< w, x > +b = 0$ and two margin hyperplanes
$< w, x > +b = 1$,
$< w, x > +b = -1$
(w and b can be scaled).



Optimization formulation

- SRM of SVM minimizes classification error and maximizes margin.
- Margin: $M = \frac{|b-1|}{\|w\|} + \frac{|-b-1|}{\|w\|} = \frac{2}{\|w\|}$.
- Empirical error: $< w, x^t > +b \geq 1$ if $y^t = 1$; $< w, x^t > +b \leq -1$ if $y^t = -1$.
- Optimization: $\min \frac{1}{2} \parallel w \parallel^2$, subject to $y^t(< w, x^t > +b) \geq 1$.

The Separable Case: Dual Problem

- Prime optimization is convex:

$$L_p = \min \frac{1}{2} \parallel w \parallel^2, \text{ subject to } y^t(< w, x^t > +b) \geq 1.$$

- Introduce Lagrange α:

$$L_p = \min \frac{1}{2} \parallel w \parallel^2 - \sum_t \alpha^t (y^t(< w, x^t > +b) - 1).$$

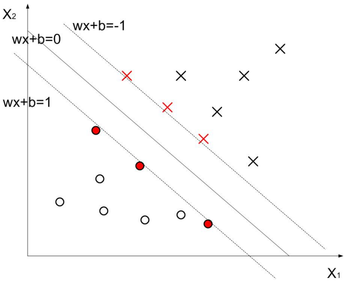
- Take partial derivatives:

$$\begin{aligned} \frac{\partial L_p}{\partial w} &= w - \sum_t \alpha^t y^t x^t = 0 \Longrightarrow w = \sum_t \alpha^t y^t x^t. \\ \frac{\partial L_p}{\partial b} &= -\sum_t \alpha^t y^t = 0. \end{aligned}$$

The Separable Case: Dual Problem

The dual form is also convex; can be solved to get αs:

$$\sum_t \alpha^t - \frac{1}{2}\sum_{t,s} \alpha^t \alpha^s y^t y^s < x^t, x^s >, \text{ subject to } \sum_t \alpha^t y^t = 0 \text{ and } \alpha^t \geq 0.$$

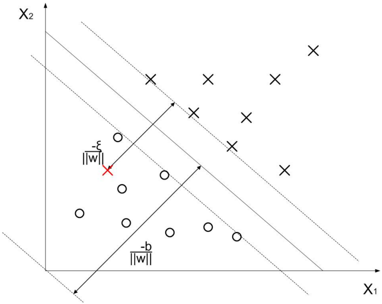


- Support vectors: data points on the margin hyperplanes with non-zero αs; they are the ones that really matter. Sparsity in solution—very few nonzero αs.
- $w = \sum_t \alpha^t y^t x^t$.
- $b = average(y^t - < w, x^t >)$ for each support vectors, since they need to satisfy $< w, x^t > + b = y^t$.

SVM: The Non-Separable Case

- Real datasets are often noisy—non-separable even with the right choice of kernel.
- Relax the constrains by introducing slack variables ξs to tolerant error.

$$< w, x^t > +b \geq 1 - \xi^t, \forall y^t = +1$$
$$< w, x^t > +b \leq -1 + \xi^t, \forall y^t = -1$$



The Non-Separable Case: Primal Problem

- New optimization penalizes the slack:

$$L_p: \quad \min \frac{1}{2} \parallel w \parallel^2 + C \sum^{t} \xi^t$$
$$\text{subject to } y^t(< w, x^t > +b) - 1 + \xi^t \geq 0 \text{ and } \xi^t \geq 0$$

- Introduce Lagrange αs and βs into L_p:

$$\min \frac{1}{2} \parallel w \parallel^2 + C \sum_t \xi^t - \sum_t \alpha^t (y^t(< w, x > +b) - 1 + \xi^t) - \sum_t \beta^t \xi^t$$

- Take partial derivatives (as before):

$$\frac{\partial L_p}{\partial w} = w - \sum_t \alpha^t y^t x^t = 0 \text{ and } \frac{\partial L_p}{\partial b} = -\sum_t \alpha^t y^t = 0.$$
$$\frac{\partial L_p}{\partial \xi^t} = C - \alpha^t - \beta^t = 0 \Longrightarrow \alpha^t = C - \beta^t \Longrightarrow \alpha^t \in [0, C].$$

The Non-Separable Case: Dual Problem

- The dual problem of non-separable case: The same as before but αs cannot be larger than C.

$$L_D : \sum_t \alpha^t - \frac{1}{2} \sum_{s,t} \alpha^t \alpha^s y^t y^s < x^t, x^s >,$$

$$\text{subject to } \sum_t \alpha^t y^t = 0 \text{ and } \alpha^t \in [0, C].$$

- Support vectors have their $\alpha \in (0, C]$; optimization gives up on those non-separable points and assigns $\alpha = C$.
- Compute $w = \sum_t \alpha^t y^t x^t$ with $\forall \alpha^t \in (0, C]$.
- Solve b with $\forall \alpha^t \in (0, C)$, since their ξ is 0.

SVMs and Kernels

- As in the perceptron algorithm, SVMs also allow dual representation in both the separable and non-separable cases.
- Separable case:

$$\sum_t \alpha^t - \frac{1}{2}\sum_{s,t} \alpha^t \alpha^s y^t y^s K(x^t, x^s), \text{ sbj } \sum_t \alpha^t y^t = 0 \text{ and } \alpha^t \geq 0.$$

- Non-Separable case:

$$\sum_t \alpha^t - \frac{1}{2}\sum_{t,s} \alpha^t \alpha^s y^t y^s K(x^t, x^s), \text{ sbj } \sum_t \alpha^t y^t = 0 \text{ and } \alpha^t \in [0, C].$$

- SVM classifiers with kernels are proved the most effective classification algorithms in many empirical problems in bioinformatics, text categorization, natural language processing, computer vision...

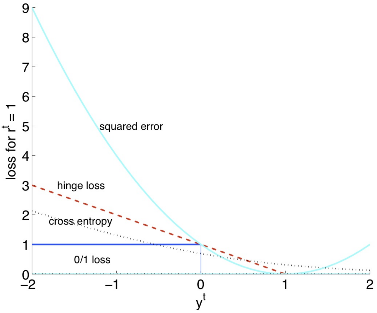
SVM and Logistic Regression

- Let $y^t =< w, x^t > +b$;
- Logistic Regression: $E_{LR}(y^t) = -logp(r^t|y^t) = log(1 + exp(-y^t))$
- Regularized Logistic Regression: $\sum_t E_{LR}(y^t) + \lambda||w||^2$. $||w||^2$ is a regularizer.
- SVMs:

$$\min \frac{1}{2} \parallel w \parallel^2 + C \sum_t \xi^t; \text{subject to } r^t(< w, x^t > +b) - 1 + \xi^t \geq 0$$

- Hinge loss $E_{SV}(y^t) = [1 - y^t * r^t]_+$
- Take the form $\sum_t E(y^t) + \lambda||w||^2$.

SVM and Logistic Regression



- Squared error: $(1 - y^t)^2$;
- Hinge loss: $[1 - y^t * r^t]_+$;
- Cross entropy: $-log \frac{1}{1+exp(-y^t)}$;
- 0/1 loss: 0 or 1.

- Hinge loss also penalizes small margin even if correctly classified.
- Hinge loss penalize linear error instead of squared error.
- Cross entropy is an approximation of hinge loss.

Multiclass SVM

- 1-vs-all
- Pairwise separation
- Error-Correcting Output Codes
- Multiclass SVM: the margin of the correct class is larger than any other class by a margin 2.

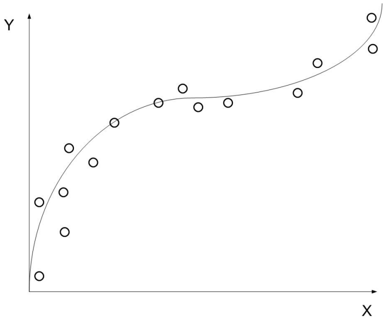
$$\min \frac{1}{2}\sum_{i=1}^{K}||w_i||^2 + C\sum_t\sum_t \xi_i^t,$$

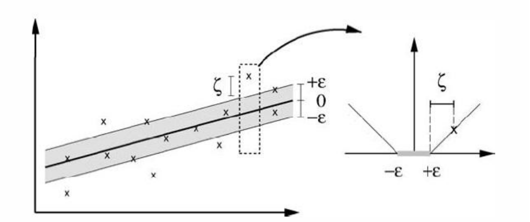
$$\text{subject to } w_{z^t}^T x^t + w_{z^t0} \geq w_i^T x^t + w_{i0} + 2 - \xi_i^t, \forall i \neq z^t, \xi^t \geq 0.$$

Notation: Regression Problem

Notation

- Input: $x \in X$, where X is a vector space $\mathbb{R}^n$
- Output: $y \in \mathbb{R}$
- Training data: $D = \{(x^1, y^1), ..., (x^t, y^t), ...\}$
- Find a function $f(x)$ to fit the data, i.e. $f(x^t) = y^t$





- Allow at most ε deviation
- As in the classification case, minimize empirical error plus the function complexity:

$$\begin{aligned} \min \quad & \frac{1}{2} \parallel w \parallel^2 + C \sum^{t} (\xi^t + \hat{\xi}^t), \\ \text{subject to} \quad & < w, x^t > + b - y^t \leq \varepsilon + \xi^t \\ & y^t - < w, x^t > - b \leq \varepsilon + \hat{\xi}^t \end{aligned}$$

SVM Regression

- Introduce Lagrange αs and $\hat{\alpha}$s:

$$L_p = \min \frac{1}{2} \parallel w \parallel^2 + C \sum^{t} (\xi^t + \hat{\xi}^t)$$
$$+ \sum^{t} \alpha^t (< w, x^t > + b - y^t - \varepsilon - \xi^t)$$
$$+ \sum^{t} \hat{\alpha}^t (y^t - < w, x^t > - b - \varepsilon - \hat{\xi}^t) - \sum^{t} (\mu^t \xi^t + \hat{\mu}^t \hat{\xi}^t)$$

- Take partial derivatives:

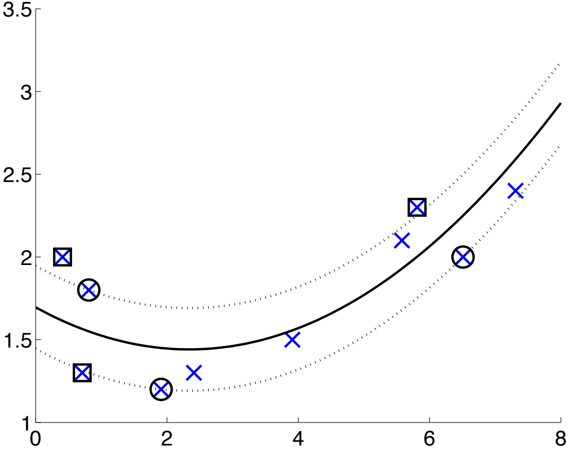
$$\frac{\partial L_p}{\partial w} = w - \sum^{t} (\hat{\alpha}^t - \alpha^t) x^t = 0 \Longrightarrow w = \sum^{t} (\hat{\alpha}^t - \alpha^t) x^t.$$
$$\frac{\partial L_p}{\partial (\xi^t, \hat{\xi}^t)} \Longrightarrow C - \alpha^t - \mu^t = 0, C - \hat{\alpha}^t - \hat{\mu}^t = 0.$$

SVM Regression Dual Form

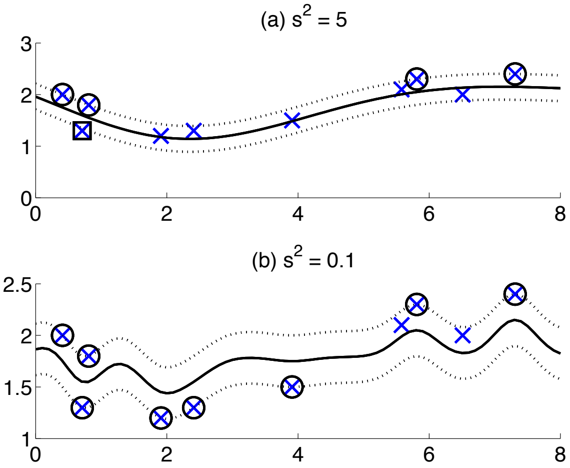
- Dual form:

$$
\begin{aligned}
\text{min} \quad & -\frac{1}{2}\sum_{i,j}(\hat{\alpha}^t - \alpha^t)(\hat{\alpha}^s - \alpha^s)(< x^t, x^s >) + \\
& \sum^t(\hat{\alpha}^t - \alpha^t)y^t - \sum^t(\hat{\alpha}^t + \alpha^t)\varepsilon \\
\text{subject to} \quad & \sum_t(\hat{\alpha}^t - \alpha^t) = 0 \\
& C \geq \alpha^t \geq 0, C \geq \hat{\alpha}^t \geq 0.
\end{aligned}
$$

SVM Regression with Non-linear Kernel



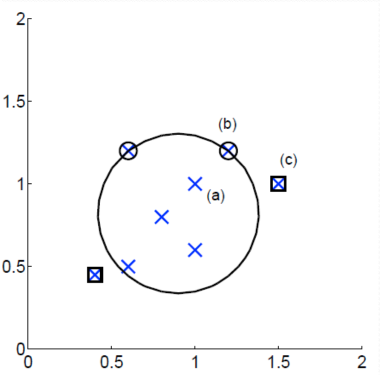
SVM Regression with Non-linear Kernel



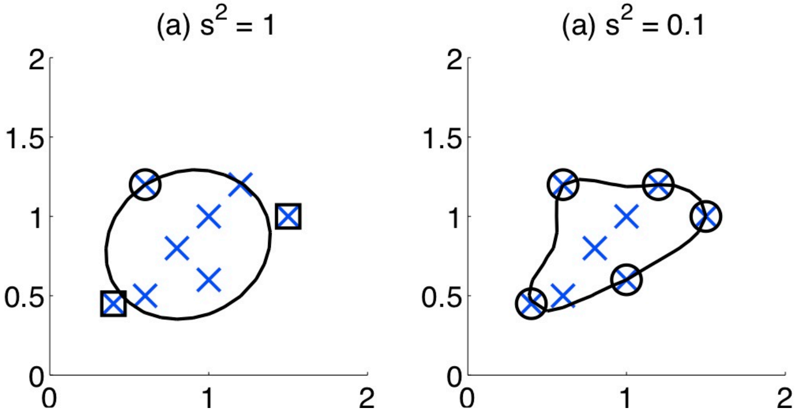
One Class SVM

$$L_p = \min_{R,a} R^2 + C \sum^{t} \xi^t$$

subject to $\xi^t \geq 0, ||x^t - a||^2 \leq R^2 + \xi^t$



One Class SVM



Summary

- Kernels are positive definite and symmetric functions for measuring similarity between two inputs.
- Kernel methods utilize kernels to deal with high dimensional or discrete data.
- SVM classifier is a linear classifier with maximum margin.
- SVM training minimizes structural risk which is a upper bound on the true error.
- The dual form of SVM optimization allows to plug-in kernel.
- The solution of SVM optimization only depends on support vectors, and often support vectors are sparse.

Other Topics about Kernel Methods and SVMs

- Fast training with Reduced Working Set algorithms or Sequential Minimal Optimization algorithms.
- SVM application in other learning problems: regression, clustering, structured output learning, feature selection...
- Optimal combination of kernels: kernel alignment and semi-definite programming.
- Design effective and fast kernels for object data such as strings, graphs, 3-D structures, images, videos, documents...

References

- *A Tutorial on Support Vector Machines for Pattern Recognition*
  Christopher J.C. Burges
  Data Mining and Knowledge Discovery, 1998
- *The Nature of Statistical Learning Theory*
  Vladimir N. Vapnik
  Springer, 1995
- *An Introduction to Support Vector Machines*
  Nello Cristianini and John Shawe-Taylor
  Cambridge University Press, 2000
- *Understanding Machine Learning: From Theory to Algorithms*
  Shai Shalev-Shwartz and Shai Ben-David
  Cambridge University Press, 2014