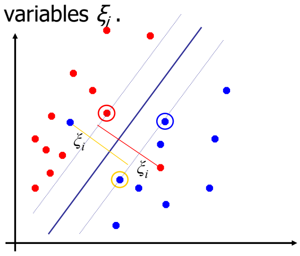
SVM: for non-linear separations

Akito Sakurai

Soft-margin classifiers

- If a training dataset is not linearly separable, we allow misclassifications for those that are difficult to classify or contaminated by noises by introducing slack variables ξ_i.



Soft margin formulation

- Hard margin formulation:

> Find $\mathbf{w}$ and b such that:
> Minimize: $\mathbf{\Phi}(\mathbf{w}) = ½\, \mathbf{w}^T\mathbf{w}$;
> Subject to: $y_i\,(\mathbf{w}^T\mathbf{x_i} + b) \geq 1$ for all $\{(\mathbf{x_i}, y_i)\}$

- Soft margin formulation with slack variables:

> Find $\mathbf{w}$ and b such that:
> Minimize: $\mathbf{\Phi}(\mathbf{w}) = ½\, \mathbf{w}^T\mathbf{w} + C\Sigma\xi_i$;
> Subject to : $y_i\,(\mathbf{w}^T\mathbf{x_i} + b) \geq 1 - \xi_i$ for all $\{(\mathbf{x_i}, y_i)\}$, and
> $\xi_i \geq 0$ for all i

- Parameter C is thought to be a overtraining controller

Soft margin solution

- Dual for the soft margin formulation:

> Find $\alpha_1 \ldots \alpha_N$ such that:
> Maximize: $\mathbf{Q}(\boldsymbol{\alpha}) = \Sigma\alpha_i - ½\Sigma\Sigma\alpha_i\alpha_j y_i y_j \mathbf{x_i}^T\mathbf{x_j}$ subject to
> (1) $\Sigma\alpha_i y_i = 0$, and
> (2) $0 \leq \alpha_i \leq C$ for all α_i

- Note that no slack variables ξ_i nor Lagrange multipliers appear in the dual problem.
- Note also that again for non-zero α_i , $\mathbf{x_i}$ is a support vector.
- A solution of the dual problem is:

> $\mathbf{w} = \Sigma\alpha_i y_i \mathbf{x_i}$
> $b = y_k(1 - \xi_k) - \mathbf{w}^T\mathbf{x_k}$ where $k = \underset{k}{\operatorname{argmax}}\ \alpha_k$

Classification can be done without explicit appearance of $\mathbf{w}$.

> $f(\mathbf{x}) = \Sigma\alpha_i y_i \mathbf{x_i}^T\mathbf{x} + b$

Lagrangian of the primal problem is:

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\mathbf{w}\cdot\mathbf{w} + C\sum_i \xi_i - \sum_i \alpha_i\big(y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - (1-\xi_i)\big) - \sum_i \nu_i\xi_i$$

Because

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial \mathbf{w}} = \mathbf{w} - \sum_i \alpha_i y_i \mathbf{x}_i \qquad \frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial b} = \sum_i \alpha_i y_i \qquad \frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial \xi_i} = C - \alpha_i - \nu_i$$

the stationary points are

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i \qquad 0 = \sum_i \alpha_i y_i \qquad 0 = C - \alpha_i - \nu_i$$

If we substitute them back to the primal problem:

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

Where the conditions must be met.

$$0 = \sum_i \alpha_i y_i \qquad 0 \leq \alpha_i \leq C \ \ (\text{for all } i)$$

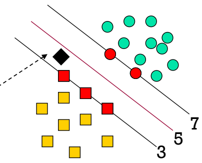
Classification by SVM

- For a given unseen point ($x1$,$x2$), measure the height to a hyper-plane (call it score):
  - For a 2-d case: score = $w_1x_1 + w_2x_2 + b$.
  - therefore: score = $wx + b = \Sigma\alpha_i y_i \mathbf{x_i}^T\mathbf{x} + b$
  - Let us define condfidence bound t.

score > t : yes

score < -t : no

otherwise: don't know



Any forced classification is granted

linear SVM: a symmarys

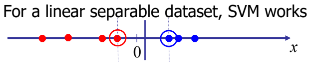
- Classifier itself is a hyperplane: *separating hyperplane.*
- The most important training samples are the support vectors because they define the discriminant function.
- A solution of the quadratic programming let us know which $\mathbf{x_i}$ is a support vector corresponding to a non-zero lagrangian multiple a_i .
- Note that the training samples appear only in the inner procuts in the dual problem and in a solution:

Find $\alpha_1 \dots \alpha_N$ such that:
Maximize: $\mathbf{Q}(\boldsymbol{\alpha}) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j \boxed{\mathbf{x_i}^T\mathbf{x_j}}$
subject to
(1) $\Sigma\alpha_i y_i = 0$, and
(2) $0 \leq \alpha_i \leq C$ for all α_i

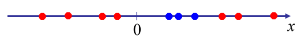
$f(\mathbf{x}) = \Sigma\alpha_i y_i \boxed{\mathbf{x_i}^T\mathbf{x}} + b$

Nonlinear SVM

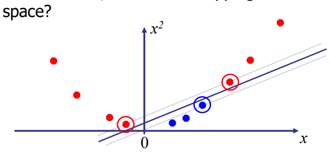
- For a linear separable dataset, SVM works even with small noise:



- Does it work for a dataset which is not linealy separable?

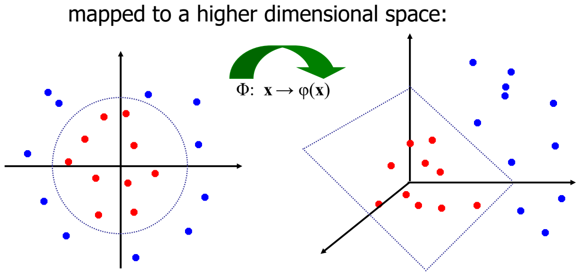


- In such cases, how about mapping them to a higher dimensional space?



Nonlinear SVM: a feature space

- A general idea: any non-linearly-separable dataset become linearly separable when its feature space is mapped to a higher dimensional space:



Higher dimensional space: a problem

- Computation time:
  - If a dataset with 1001 samples is mapped to a 1000 dimensional space with an appropriate non-linear function, the dataset becomes linearly separable.
  - But not only computation of the nonlinear function takes time, but also 1000 times computation is required for each sample, the total computation time is huge.

  ⇒ "kernel trick" is utilized to reduce the computation time.

- Generalization capability:
  - Any dataset with 1001 samples (in general position) can be separated linearly.
  - This means that any target label set can be implemented by a hyperplane. It is definitely overtrained.

  ⇒ A solution to this problem is a large margin classifier.

Nonlinear mapping

- Consider a nonlinear function $\phi(x)$ $\varphi: R^N \to F$ that maps a sample x to a high dimensional space F.

Primal Lagrangian:

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\mathbf{w}\cdot\mathbf{w} - \sum_i \alpha_i (y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1)$$

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\mathbf{w}\cdot\mathbf{w} - \sum_i \alpha_i (y_i(\mathbf{w}\cdot\varphi(\mathbf{x}_i) + b) - 1)$$

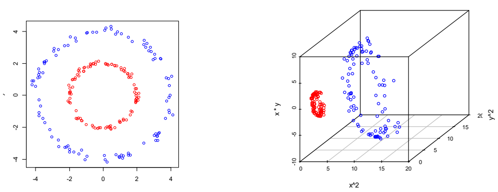
Dual Lagrangian:

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

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j \varphi(\mathbf{x}_i)\cdot\varphi(\mathbf{x}_j)$$

$$\sum_i \alpha_i y_i = 0 \text{ and } \forall i \ \alpha_i \geq 0$$

An example



$$(x_1, x_2) \to (x_1^2, x_2^2, x_1 x_2)$$

“Kernel Trick”

- Note that $\Phi(x)$ appears only in an inner product such as $\Phi(x)\cdot\Phi(y)$.

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}_j)$$

- Therefore if a simple function K exists such that $K(x,y)=\Phi(x)\cdot\Phi(y)$, then computational burden is reduced.
  - Moreover if $K(x,y)$ is a function of $x\cdot y$, much less computation is needed.

Mercer’s theorem

- A function K is written in an inner product form:

$$K(x,y) = \sum_{i=1}^{\infty} \lambda_i \phi_i(x) \phi_i(y)$$

if and only if K is symmetric and positive semi-definite. i.e.,

$$K(x,y) = K(y,x)$$

$$\iint K(x,y) f(x) f(y) dx dy \geq 0 \quad \text{for any } f$$

where $\phi_i(x)$ is an eigen function of $K(x,y)$ i.e.,

$$\int K(x,y)\phi_i(x)dx = \lambda_i \phi_i(y)$$

Common kernel functions

- Linear kernel $K(x,y) = x^T y$
- polynomial $K(x,y) = (x^T y + 1)^p$ or $(x^T y)^p$
- RBF $K(x,y) = \exp\left(-\|x-y\|^2 / 2\sigma^2\right)$
- MLP $K(x,y) = \tanh\left(\beta_0 x^T y + \beta_1\right)$ ← Not positive semidefinite
- An example: For a 2-D vector $\mathbf{x}=[x_1 \ \ x_2]$, set $K(\mathbf{x}_i,\mathbf{x}_j)=(1 + \mathbf{x}_i^T\mathbf{x}_j)^2$,
  Then the following holds for $K(\mathbf{x}_i,\mathbf{x}_j)= \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j)$:
  $K(\mathbf{x}_i,\mathbf{x}_j)=(1 + \mathbf{x}_i^T\mathbf{x}_j)^2$,
  $= 1+ x_{i1}^2x_{j1}^2 + 2\, x_{i1}x_{j1}\, x_{i2}x_{j2} + x_{i2}^2x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2}$
  $= [1 \ \ x_{i1}^2 \ \ \sqrt{2}\, x_{i1}x_{i2} \ \ x_{i2}^2 \ \ \sqrt{2}x_{i1} \ \ \sqrt{2}x_{i2}]^T\, [1 \ \ x_{j1}^2 \ \ \sqrt{2}\, x_{j1}x_{j2} \ \ x_{j2}^2 \ \ \sqrt{2}x_{j1} \ \ \sqrt{2}x_{j2}]$
  $= \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j)$
  where $\varphi(\mathbf{x}) = [1 \ \ x_1^2 \ \ \sqrt{2}\, x_1x_2 \ \ x_2^2 \ \ \sqrt{2}x_1 \ \ \sqrt{2}x_2]$

SVM: generalization capability

- A classifier with high generalization capability is sought.
- How to get a better generalization performance?
  - A larger training dataset
  - Reduce errors for training dataset properly
  - Larger capacity/variance (the number of parameters and/or expressiveness of models)
- In SVM, an error bound for an unseen sample is given based on these values.

Risk bound by VC dimension

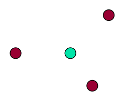
- Theoretical risk bound:

$$R(\alpha) \leq R_{emp}(\alpha) + \sqrt{\frac{h(\log(2l/h) + 1) - \log(\eta/4)}{l}}$$

- Risk = average error rate
- α – the model (parameters define it)
- R_{emp} – empirical risk, l – # of observations, h – VC dim.,
- The expression holds with probability (1-η)
  - VC (Vapnik-Chervonenkis) dimension: maximum # of points which can be shattered
  - A point set is shattered if any labeling of the points is realizable by a classfier.
- A very important theoretical property. But not often used.

Ex.: VC dim. Of a hyperplane

- Suppose that there are n points in a d dim. space, and they are labeled red or green. How large (as a function of d) n should be for us to be able to find an example where red and green points are not linearly separable?
- Ex. For d =2, $n \geq 4$.



A sketch: theoretical evidence of margin maximization

- Vapnik proved the following: *VC* dim. h of a linear classifier set with margin greater than ρ has an upper bound:

$$h \le \min\left\{\left\lceil \frac{D^2}{\rho^2} \right\rceil, m_0\right\} + 1$$

where D is the radius of a smallest hypersphere that encloses all the training examples, and m_0 is the dimension of the sample space.

- Intuitively, this shows that regardless of the dimension m_0 of the sample space, by maximizing the marge ρ, VC dimension is minimized.

- In this way, we can reduce the complexity of classifiers irrelevant to the dimension of the sample space..

Vapnik 1982: Estimation of Dependences Based on Empirical Data: Springer Series in Statistics (Springer Series in Statistics) Springer-Verlag New York, Inc.

Performance of SVM

- Many believe SVM has the highest performance.
- Sometimes statistically relevant, sometimes not. There are some that perform similar to SVM.
- Ex.: regularized logistic regression (Zhang & Oles)
  - Tong Zhang, Frank J. Oles: Text Categorization Based on Regularized Linear Classification Methods. Information Retrieval 4(1): 5-31 (2001)
- Comparison: Yang & Liu
  - Yiming Yang, Xin Liu: A re-examination of text categorization methods, 22nd Annual International SIGIR (1999).

Ex.: A classic dataset Reuters

- A dataset used quite often
- 21578 documents
- 9603 training, 3299 test articles (ModApte split)
- 118 categories
  - One article could belong to more than one category
  - 118 binary classes
- Average number of categories per 1 document
  - 1.24
- 10 categories are significant（among 118 categories）

Significant categories (#train, #test)

- Earn (2877, 1087)
- Acquisitions (1650, 179)
- Money-fx (538, 179)
- Grain (433, 149)
- Crude (389, 189)
- Trade (369,119)
- Interest (347, 131)
- Ship (197, 89)
- Wheat (212, 71)
- Corn (182, 56)

Reuters Text Categorization data set (**Reuters-21578**) document

<REUTERS TOPICS="YES" LEWISSPLIT="TRAIN" CGISPLIT="TRAINING-SET" OLDID="12981" NEWID="798">

<DATE> 2-MAR-1987 16:51:43.42</DATE>

<TOPICS><D>livestock</D><D>hog</D></TOPICS>

<TITLE>AMERICAN PORK CONGRESS KICKS OFF TOMORROW</TITLE>

<DATELINE> CHICAGO, March 2 - </DATELINE><BODY>The American Pork Congress kicks off tomorrow, March 3, in Indianapolis with 160 of the nations pork producers from 44 member states determining industry positions on a number of issues, according to the National Pork Producers Council, NPPC.

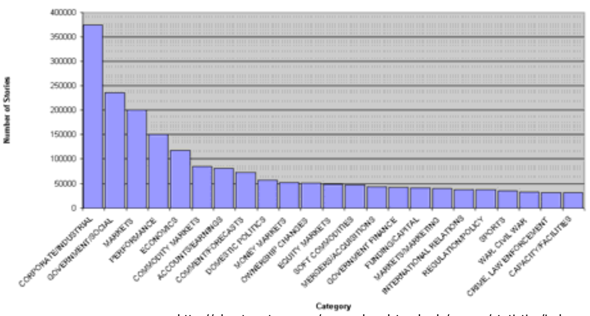
Delegates to the three day Congress will be considering 26 resolutions concerning various issues, including the future direction of farm policy and the tax law as it applies to the agriculture sector. The delegates will also debate whether to endorse concepts of a national PRV (pseudorabies virus) control and eradication program, the NPPC said.

A large trade show, in conjunction with the congress, will feature the latest in technology in all areas of the industry, the NPPC added. Reuter

&#3;</BODY></TEXT></REUTERS>

New Reuters: RCV1: 810,000 docs.

- Frequent categories in Reuters RCV1



http://about.reuters.com/researchandstandards/corpus/statistics/index.asp

Dumais et al. 1998: Reuters – Break-Even Performance

	Rocchio	NBayes	Trees	LinearSVM
earn	92.9%	95.9%	97.8%	98.2%
acq	64.7%	87.8%	89.7%	92.8%
money-fx	46.7%	56.6%	66.2%	74.0%
grain	67.5%	78.8%	85.0%	92.4%
crude	70.1%	79.5%	85.0%	88.3%
trade	65.1%	63.9%	72.5%	73.5%
interest	63.4%	64.9%	67.1%	76.3%
ship	49.2%	85.4%	74.2%	78.0%
wheat	68.9%	69.7%	92.5%	89.7%
corn	48.2%	65.3%	91.8%	91.1%
Avg Top 10	64.6%	81.5%	88.4%	91.4%
Avg All Cat	61.7%	75.2%	na	86.4%

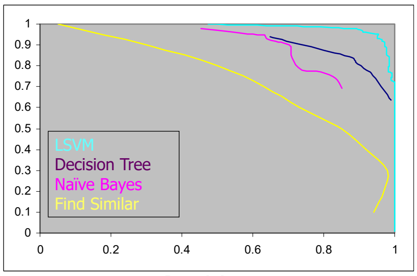
Break Even: Recall = Precision

Recall: = TP/(TP+TN);

Precision: = TP/(TP+FP);

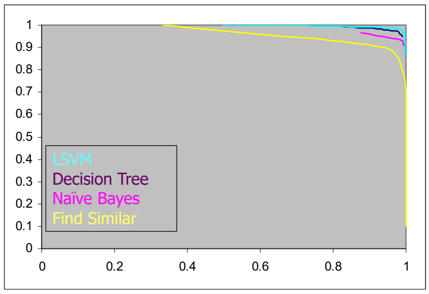
S. T. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In CIKM-98: Proceedings of the Seventh International Conference on Information and Knowledge Management, 1998.

Precision vs. Recall - Category “Grain”

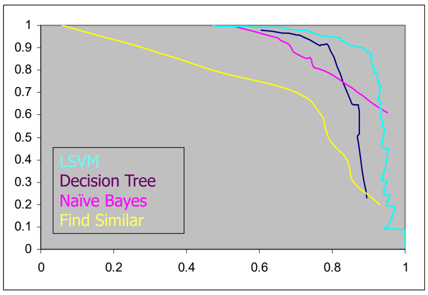


Recall: = TP/(TP+TN);
Precision: = TP/(TP+FP);

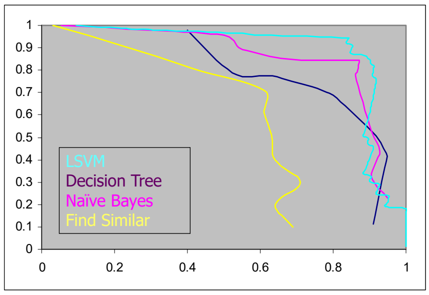
Precision vs. Recall - Category “Earn”



Precision vs. Recall - Category “Crude”



Precision vs. Recall - Category “Ship”



Kernel difference (Joachims)

					SVM (poly) degree $d =$					SVM (rbf) width $\gamma =$			
	Bayes	Rocchio	C4.5	k-NN	1	2	3	4	5	0.6	0.8	1.0	1.2
earn	95.9	96.1	96.1	97.3	98.2	98.4	**98.5**	98.4	98.3	**98.5**	98.5	98.4	98.3
acq	91.5	92.1	85.3	92.0	92.6	94.6	**95.2**	95.2	95.3	95.0	95.3	95.3	**95.4**
money-fx	62.9	67.6	69.4	78.2	66.9	72.5	75.4	74.9	**76.2**	74.0	75.4	**76.3**	75.9
grain	72.5	79.5	89.1	82.2	91.3	93.1	**92.4**	91.3	89.9	**93.1**	91.9	91.9	90.6
crude	81.0	81.5	75.5	85.7	86.0	87.3	88.6	**88.9**	87.8	**88.9**	89.0	88.9	88.2
trade	50.0	77.4	59.2	77.4	69.2	75.5	76.6	77.3	**77.1**	76.9	78.0	**77.8**	76.8
interest	58.0	72.5	49.1	74.0	69.8	63.3	67.9	73.1	**76.2**	74.4	75.0	**76.2**	76.1
ship	78.7	83.1	80.9	79.2	82.0	85.4	86.0	**86.5**	86.0	**85.4**	86.5	87.6	87.1
wheat	60.6	79.4	85.5	76.6	83.1	84.5	85.2	**85.9**	83.8	**85.2**	85.9	85.9	85.9
corn	47.3	62.2	87.7	77.9	86.0	86.5	85.3	**85.7**	83.9	**85.1**	85.7	85.7	84.5
microavg.	**72.0**	**79.9**	**79.4**	**82.3**	84.2	85.1	85.9	86.2	85.9	86.4	86.5	86.3	86.2
					combined: **86.0**					combined: **86.4**			

Fig. 2. Precision/recall-breakeven point on the ten most frequent Reuters categories and microaveraged performance over all Reuters categories, k-NN, Rocchio, and C4.5 achieve highest performance at 1000 features (with k = 30 for k-NN and β = 1.0 for Rocchio). Naive Bayes performs best using all featurcs.

T. Joachims, *Text Categorization with Support Vector Machines: Learning with Many Relevant Features*. Proceedings of the European Conference on Machine Learning (ECML), Springer, 1998

Yang&Liu: SVM vs others

Table 1: Performance summary of classifiers

method	miR	miP	miF1	maF1	error
SVM	.8120	.9137	.8599	.5251	.00365
KNN	.8339	.8807	.8567	.5242	.00385
LSF	.8507	.8489	.8498	.5008	.00414
NNet	.7842	.8785	.8287	.3765	.00447
NB	.7688	.8245	.7956	.3886	.00544

miR = micro-avg recall; miP = micro-avg prec.;
miF1 = micro-avg F1; maF1 = macro-avg F1.

LLSF: Linear Least Square Fit

Nnet: 64 hidden units
SVM kernel: linear (better than others)

```
# of features
NNet: 1000
NB:   2000
kNN:  2415
LLSF: 2415
SVM: 10000
```

Summary

- Support Vector Machine (SVM) is
  - Defines a hyperplane utilizing support vectors
    - Support vector = critical samples close to decision boundary
  - linear SVM is a linear classifier.
  - Kernel: maps samples to a higher dim. space where inner products are calculated easily
  - Risk (expected error rate on test data) upper bound
  - A best classifier when irrelevant features exist
    - For 1000 features, svm is robust
  - Popularized after free SVMlight
    - Runs fast and free (for research purpose)
    - A few others: TinySVM, libsvm, ….
  - Still very common

SVR: support vector regression

Akito Sakurai

SVR: Support Vector Regression

SVM is

a classification method which uses a linear function:

$$y(x) = w^T \varphi(x) + b$$

Let us use it for regression

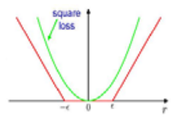
A view: cost function = error + regularization

By a linear regression, minimize the following error func.

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

The squared error is replace by an ε–insensitive errir:

$$C\sum_{n=1}^{N}E_\varepsilon(y(x_n) - t_n) + \frac{1}{2}\|w\|^2$$



Ex. ε–insensitive error function:

$$L_\varepsilon(y) = \begin{cases} 0 & \text{for} \quad |f(x) - y| < \varepsilon \\ |f(x) - y| - \varepsilon & \text{otherwise} \end{cases}$$

Gunn. Support Vector Machines for Classification and Regression

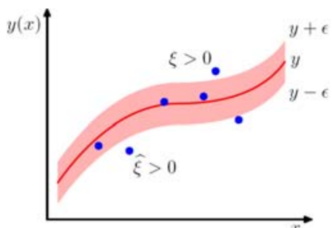
Introduction of slack variables

For target values are in this ε-tube:

$$y_n - \varepsilon \le t_n \le y_n + \varepsilon$$

To allow outer samples to be outside of ε-tube:

$$t_n \le y(x_n) + \varepsilon + \xi_n$$
$$t_n \ge y(x_n) - \varepsilon - \xi_n^-$$



Oprimization problem for SVR

Minimize:

$$C\sum_{n=1}^{N}(\xi_n + \xi_n^-) + \frac{1}{2}\|w\|^2$$

Subject to:

$\xi_n \ge 0$, $\xi_n^- \ge 0$ and $t_n \le y(x_n) + \varepsilon + \xi_n$, $t_n \ge y(x_n) - \varepsilon$

Lagrangian

Primal:

$$L = C\sum_{n=1}^{N}(\xi_n + \xi_n^-) + \frac{1}{2}\|w\|^2 - \sum_{n=1}^{N}(\mu_n\xi_n + \mu_n^-\xi_n^-) - \sum_{n=1}^{N}a_n(\varepsilon + \xi_n + y_n - t_n) - \sum_{n=1}^{N}a_n^-(\varepsilon + \xi_n^- - y_n + t_n)$$

$$\frac{\partial L}{\partial w} = 0 \Rightarrow w = \sum_{n=1}^{N}(a_n - a_n^-)\phi(x_n)$$

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

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

$$\frac{\partial L}{\partial \xi_n^-} = 0 \Rightarrow a_n^- + \mu_n^- = C$$

Dual

Maximize:

$$W(a, a^-) = -\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}(a_n - a_n^-)(a_m - a_m^-)k(x_n, x_m) - \varepsilon\sum_{n=1}^{N}(a_n + a_n^-) + \sum_{n=1}^{N}(a_n - a_n^-)t_n$$

Subject to:

$$0 \le a_n \le C$$
$$0 \le a_n^- \le C$$

prediction:

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

The constant term b

Karush-Kuhn-Tucker (KKT) condition:

$$a_n(\varepsilon + \xi_n + y_n - t_n) = 0$$
$$a_n^-(\varepsilon + \xi_n^- - y_n + t_n) = 0$$
$$(C - a_n)\xi_n = 0$$
$$(C - a_n^-)\xi_n^- = 0$$
$$b = t_n - \varepsilon - w^T\varphi(x_n)$$
$$= t_n - \varepsilon - \sum_{m=1}^{N}(a_m - a_m^-)k(x_n, x_m)$$