SysEng 5212 /EE 5370
Introduction to Neural Networks and Applications

Lecture 10: Classier Performance Evaluation and Support Vector Machines

Cihan H Dagli, PhD

Professor of Engineering Management and Systems Engineering
Professor of Electrical and Computer Engineering
Founder and Director of Systems Engineering Graduate Program

dagli@mst..edu

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY
Rolla, Missouri, U.S.A.

Lecture outline

- **Classier Performance Evaluation**
  - Confusion Matrix
  - ROC Curves
- **Support Vector Machines**
  - Introduction
  - Linearly Separable Patterns
  - Nonlinearity Separable Patterns
  - SVM as a Kernel Machine

Project Status Review Presentations

April 24, 2018

Murat Aslan, Tatiana Cardona Sepulveda, Prince Codjoe, Xiongming Dai
Jeffrey Dierker, Venkata Sai Abhishek Dwivadula, Brian Guenther, Anthony Guertin
Timothy Guertin, Seth Kitchen

May 1, 2018

Gregory Leach, Yu Li, John Nganga, Igor Povarich, Jack Savage, William Symolon
Wayne Viers III, Tao Wang, Kari Ward, Julia White, Jun Xu

Project Status Presentation Template

- Project Title
- Description of engineering problem to be solved by Neural Network (one slide)
- What methods are currently being used? (one slide)
- What will be impact of problem solution success? (one slide)

Project Status Presentation Template

- Data structure to be used in solving the engineering problem (one slide)
- Input data representation and possible pre-processing (one slide)
- Out put data representation and possible post processing (one slide)
- Possible neural network architectures to be used and why they are selected? (one slide)

Project Status Presentation Template

- Schematic representation of the solution architecture proposed (one slide) (It is possible for you to use more than one neural network or collection of neural networks for solving your problem)
- Input and output data statistics (one slide)
  - Number of data points
  - Training data size
  - Testing data size
  - Validation data size
  - Type of data normalization

Project Status Presentation Template

- Progress you made so far (multiple slides)
- Problems that you are facing and how are you mitigate them? (one or two slides)

Two-Class Classifier Example

- Consider a classier for credit card fraud detection which determines whether transactions are legal or fraudulent
- The stated objective (hypothesis) of the classier is to detect fraud.
- Each transaction is classified into two classes: fraudulent(true) and legal(false)

There are four possible outcomes for each classification result.

- True Positive: A fraudulent transaction is classified as fraud
- False Positive: A legal transaction is classified as fraud
- True Negative: A legal transaction is classified as legal
- False Negative: A fraudulent transaction is classified as legal

Neural Network Output Classifications

- Consider a binary classification problem,

  Class 1 : 1

  Class 2 : 0

- Network is trained to classify inputs as class 1 (C1) or class 2(C2).
- How do we determine which class the neural network's output belongs to?

Neural Network Output Threshold

Let β be the classification threshold or tolerance and γ be the midpoint of the output interval.

For output interval $[0, 1]$, $\gamma = 0.5$.

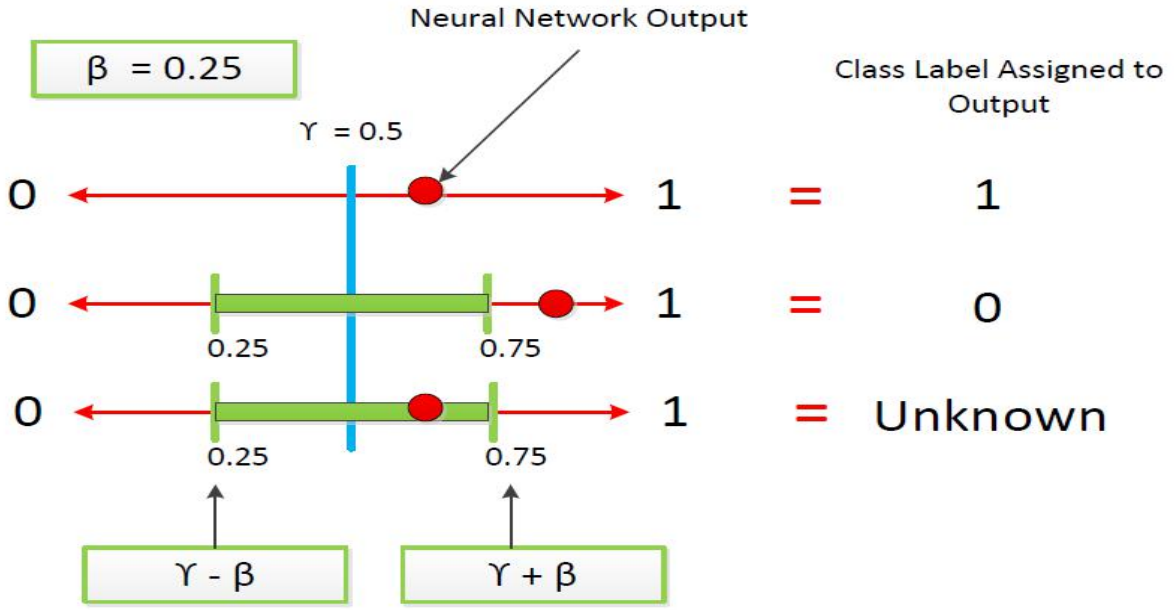
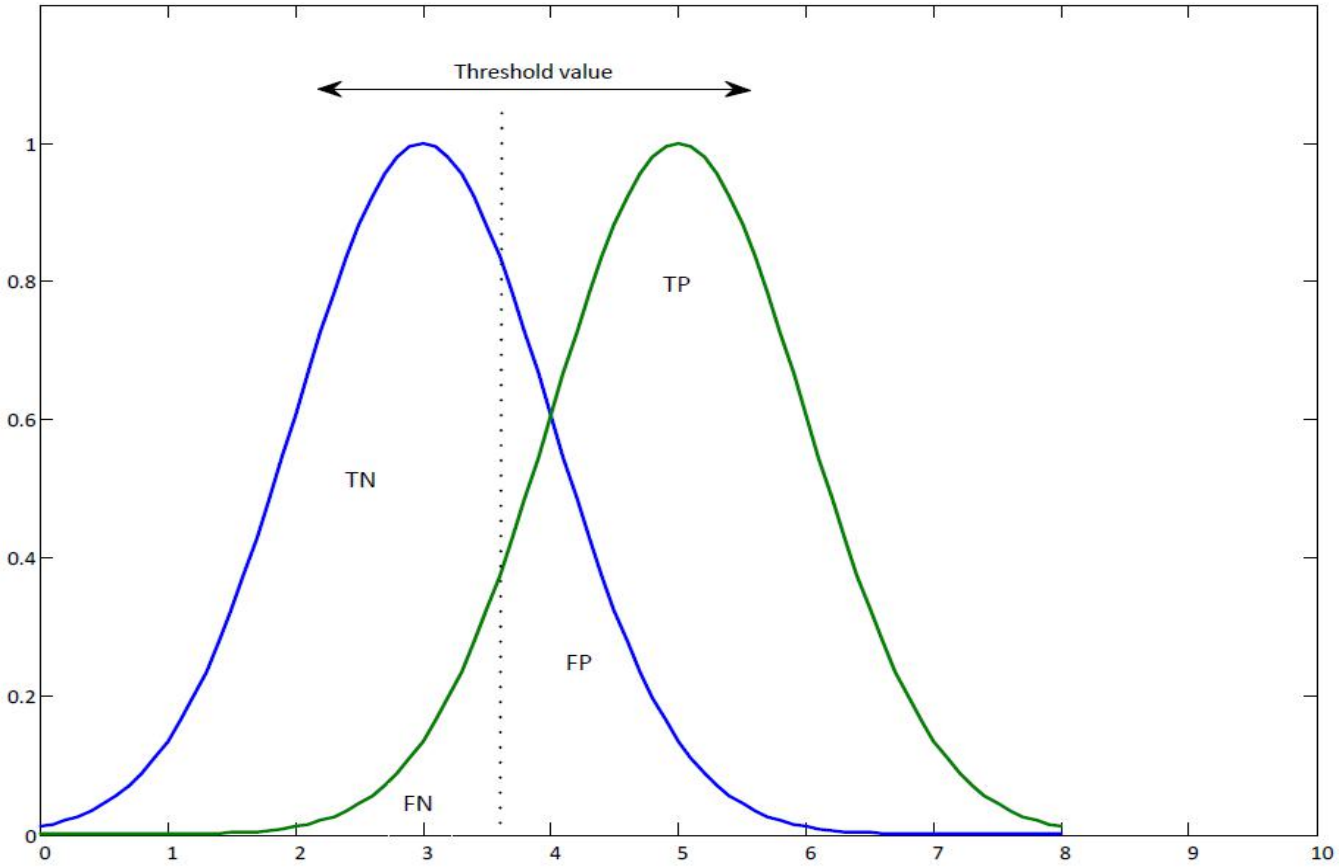


Illustration of Classier Results



Confusion Matrix

- The confusion matrix depicts the observed or actual classifications (columns) versus the predicted classification of the model (rows).
- The correct classifications, i.e., the true positives and true negatives lie along the diagonal of the confusion matrix.
- A perfect classier would have zero false positives and false negatives.
- Other model performance metrics can be derived from the elements of the confusion matrix.

Confusion Matrix (Contd.)

Two-Class Problem:

		Prediction		
		c_1	c_2	Total
Actual	c_1	TP	FP	N^+
	c_2	FN	TN	N^-
	Total	$\hat{N}^1$	$\hat{N}^2$	N

Confusion Matrix (Contd.)

Several-Class Problem

		Prediction					
		c_1	c_2	c_3	...	c_n	Total
Actual	c_1	TP_1	FN_{12}	FN_{13}	...	FN_{1n}	N_1
	c_2	FN_{21}	TP_2	FN_{23}	...	FN_{2n}	N_2
	c_3	FN_{31}	FN_{32}	TP_3	...	FN_{3n}	N_3
	...	...	...	...	...	...	...
	c_n	FN_{n1}	FN_{n2}	FN_{n3}	...	TP_n	N_n
	Total	$\hat{N}_1$	$\hat{N}_2$	$\hat{N}_3$	...	$\hat{N}_n$	$\hat{N}$

Evaluation Measures Derived from the Confusion Matrix

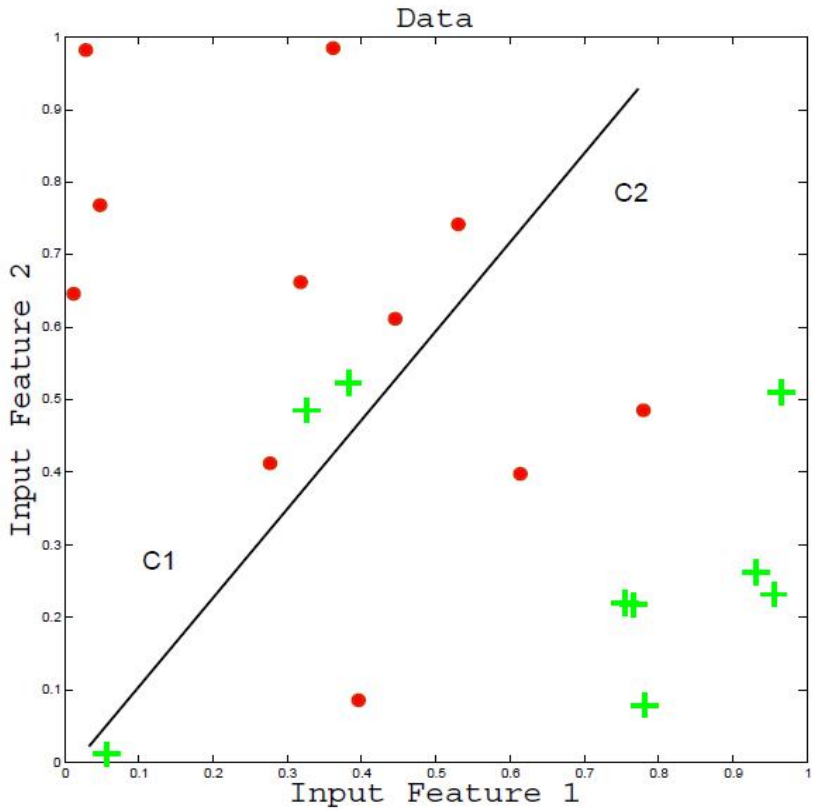
Based on Confusion Matrix:

- Accuracy/Classification Error
- Recall/Sensitivity
- Specificity
- Precision
- F-score

Based on Receiver Operator Characteristic (ROC):

- Area under the ROC curve (AUC)

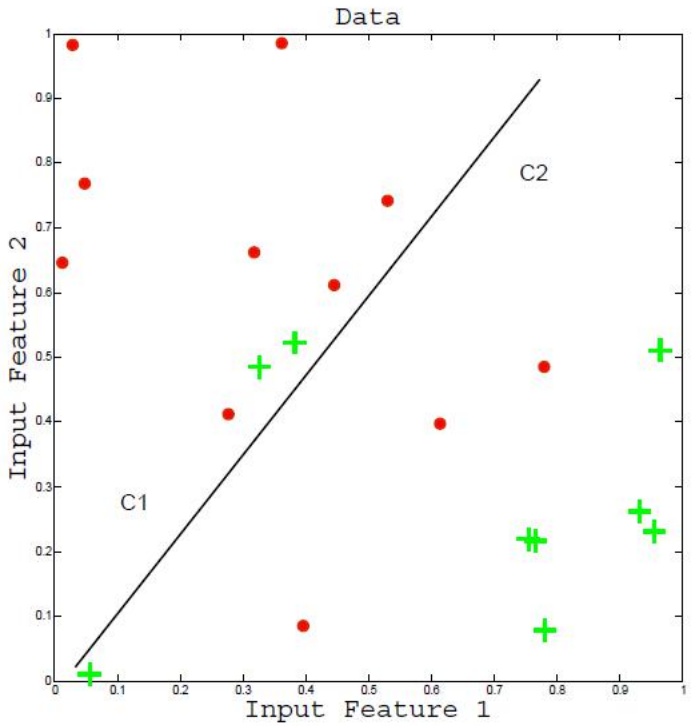
Two-Class Data



Data:

X_1	X_2	C
0.29	0.42	c_1
0.39	0.86	c_1
0.79	0.79	c_2
0.34	0.47	c_2
...	...	...

Two-Class Confusion Matrix

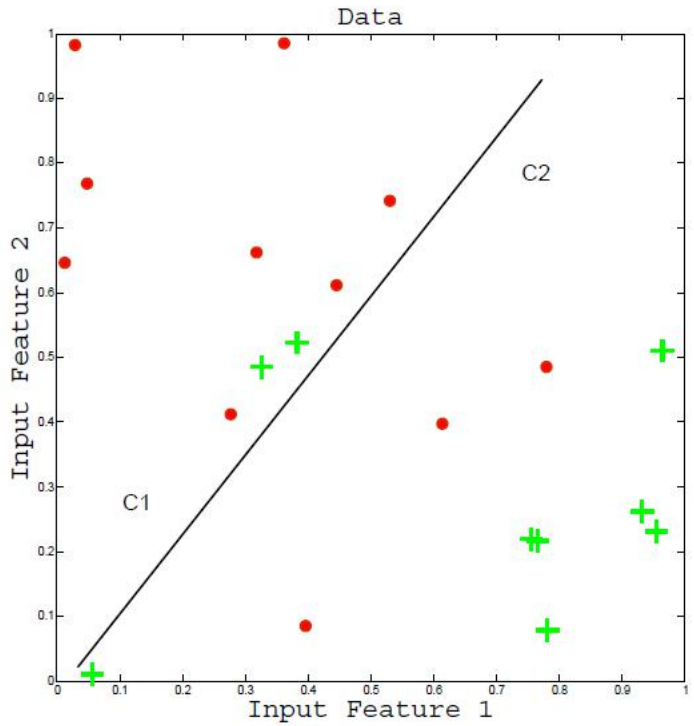


Confusion matrix:

		Prediction		
		c_1	c_2	Total
Actual	c_1	8	3	11
	c_2	2	7	9
	Total	10	10	20

Accuracy/Classification Error

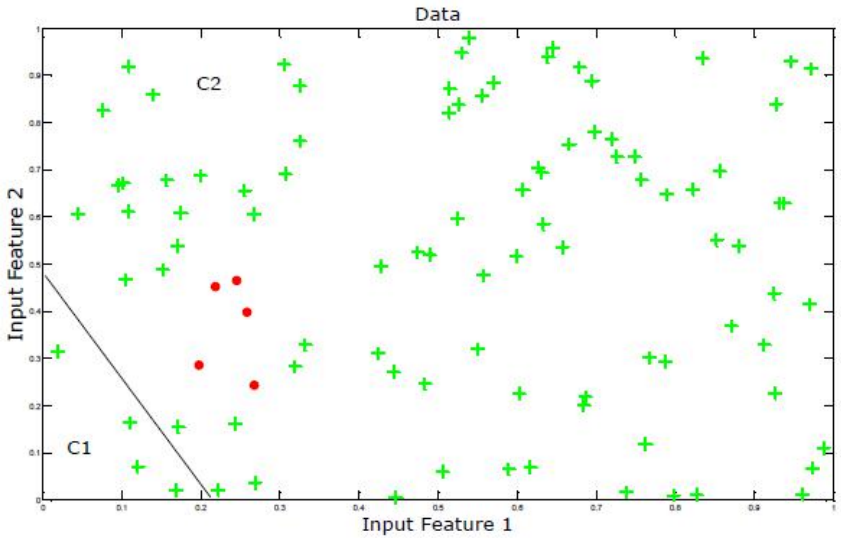
Definition: Data samples classified correctly/incorrectly.



		Prediction		
		c_1	c_2	Total
Actual	c_1	8	3	11
	c_1	2	7	9
	Total	10	10	20

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{8 + 7}{20} = 75\%$$

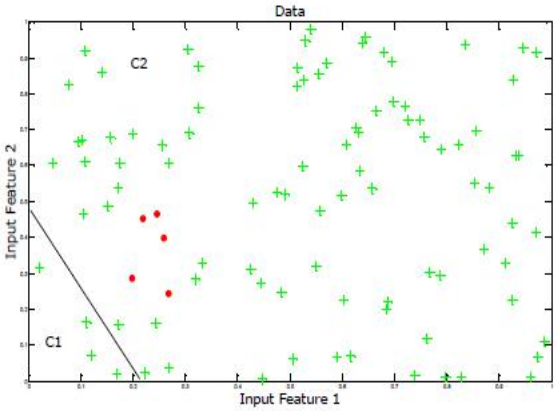
Two Class Problem with Skewed Data



X_1	X_2	C
0.12	0.3	c_2
0.12	0.6	c_2
0.16	0.27	c_2
0.22	0.3	c_2
0.28	0.24	c_1
0.26	0.24	c_1
...	...	...

The distribution of samples in the two classes is unbalanced. Class 2 has many more samples that class 1.

Skewed Data - Classification Accuracy



		Prediction		
		c_1	c_2	Total
Actual	c_1	0	4	4
	c_2	5	91	96
	Total	5	95	100

$$\epsilon = \frac{TP + TN}{TP + TN + FP + FN}$$
$$= \frac{0 + 91}{100} = 91\% \text{ Misleading!}$$

Recall

- Fraction of positive class samples correctly classified
- Also called True Positive Rate (TPR) or Sensitivity

		Prediction		
		c_1	c_2	Total
Actual	c_1	0	4	4
	c_2	5	91	96
	Total	5	95	100

$$r(\phi) = \frac{TP}{TP + FN} = \frac{TP}{P}$$
$$= \frac{0}{0 + 5} = 0 \text{ Very poor recall!}$$

Precision

Definition:

- Fraction of data samples classified as c_1 which are actually c_1
- Equivalent to Positive Predictive Value (PPV)

		Prediction		
		c_1	c_2	Total
Actual	c_1	0	4	4
	c_2	5	91	96
	Total	5	95	100

$$pr(\phi) = \frac{TP}{TP + FP}$$
$$= \frac{0}{0+4} = 0 \text{ Very poor precision!}$$

Specificity

Definition:

- Fraction of negative class samples correctly classified
- Specificity = 1 - False Positive Rate

		Prediction		
		c_1	c_2	Total
Actual	c_1	0	4	4
	c_2	5	91	96
	Total	5	95	100

$$FPR(\phi) = \frac{FP}{TN + FP} = \frac{FP}{N} = \frac{4}{95} = 0.0421$$

$$sp(\phi) = \frac{TN}{TN + FP} = \frac{TN}{N} = \frac{91}{95} = 0.9579$$

Positive and Negative Predictive Values

- Positive Predictive Value (PPV): Fraction of data samples classified as c_1 which are actually c_1
- Negative Predictive Value (NPV): Fraction of data samples classified as c_2 which are actually c_2

$$pr(\phi) = \frac{TP}{TP + FP} = \frac{0}{0 + 4} = 0$$

$$pr(\phi) = \frac{TN}{TN + FN} = \frac{91}{91 + 5} = 0.9479$$

Predictive value for c_1 is very poor but the predictive value for c_2 is quite good. Can be attributed to the larger amount of training samples available for class 2.

Balanced Scores

- Balanced accuracy rate

$$acc_{bal.} = \frac{1}{2}\left(\frac{TP}{P} + \frac{TN}{N}\right) = \frac{recall + specificity}{2}$$

- Balanced error rate

$$\epsilon_{bal} = \frac{1}{2}\left(\frac{FP}{P} + \frac{FN}{N}\right)$$

- Balanced scores for the skewed data

		Prediction		
		c_1	c_2	Total
Actual	c_1	0	4	4
	c_2	5	91	96
	Total	5	95	100

$$Bal.acc. = \frac{1}{2}\left(\frac{0}{5} + \frac{91}{95}\right) = 0.4789$$

$$Bal.\epsilon = \frac{1}{2}\left(\frac{4}{5} + \frac{5}{95}\right) = 0.426$$

Final Outcomes for Skewed Data Example

		Prediction		Total	
		c_1	c_2		
Actual	c_1	0	4	4	$PPV = \frac{0}{0+4} = 0$
	c_2	5	91	96	$NPV = \frac{91}{5+91} = 0.9479$
	Total	5	95	100	
		Sensitivity $= \frac{0}{0+5} = 0$	**Specificity** $= \frac{91}{4+91} = 0.9579$		

Classification Cost

- All misclassifications cannot be weighed equally.
- Some misclassifications may be more desirable than other
- Credit card fraud detection: Some inconvenience to customer is more desirable versus allowing some fraudulent transactions to go undetected.
- Diagnosing a healthy patient as sick is preferable to diagnosing a sick patient as healthy.
- Some spam in the inbox is preferable to tagging legitimate emails as spam.
- In these cases minimizing overall cost of misclassification is preferred over minimizing classification error.

Classification Cost Computation

- A cost, utility or penalty value is associated with each class.
- Costs for misclassification for different classes will depend on the relative importance of that class.
- The cost matrix shows the costs associated with each type of classification error.

		Prediction	
		c_1	c_2
Actual	c_1	$COST_{TP}$	$COST_{FN}$
	c_2	$COST_{FP}$	$COST_{TN}$

- The most common type of cost function is the 0/1 function. A penalty of 0 is assessed for correct classification and 1 for misclassifications.

		Prediction	
		c_1	c_2
Actual	c_1	0	1
	c_2	1	0

Generalization to Multi-Class Problems

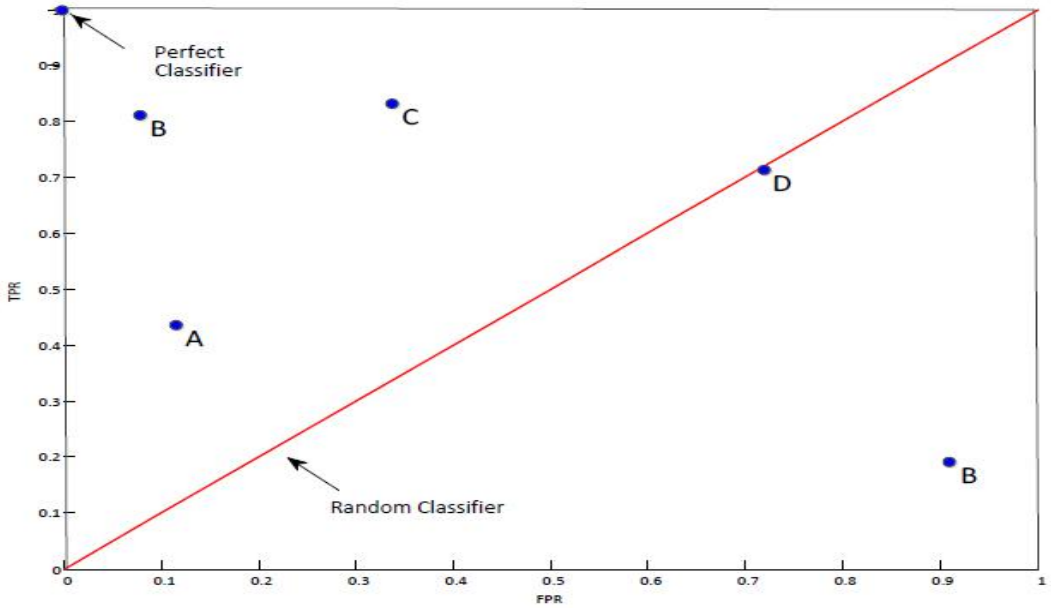
- All evaluation measures obtained from the confusion matrix can be generalized to multi-class problems.

		Prediction					
		c_1	c_2	c_3	...	c_n	Total
Actual	c_1	TP_1	FN_{12}	FN_{13}	...	FN_{1n}	N_1
	c_2	FN_{21}	TP_2	FN_{23}	...	FN_{2n}	N_2
	c_2	FN_{31}	FN_{32}	TP_3	...	FN_{3n}	N_3
	...	...	...	...	...	...	...
	c_n	FN_{n1}	FN_{n2}	FN_{n3}	...	TP_n	N_n
	Total	$\hat{N}_1$	$\hat{N}_2$	$\hat{N}_3$	...	$\hat{N}_n$	$\hat{N}$

Receiver Operating Characteristics (ROC)

- Performance metrics derived from the confusion matrix are sensitive to data anomalies such as class skew.
- ROC curves are a graphical tool for evaluating classier performance.
- They convey the information from a confusion matrix in a more intuitive fashion.
- ROC curves are plots of the true positive rate (sensitivity) versus the false positive rate (1-specificity) for different values of a threshold parameter.
- The Area Under the ROC Curve (AUC) is a measure of how well a model can distinguish between two classes.
- A valid AUC estimate can be interpreted as the probability that the classier will assign a higher score to a randomly chosen positive sample than to a randomly chosen negative sample.

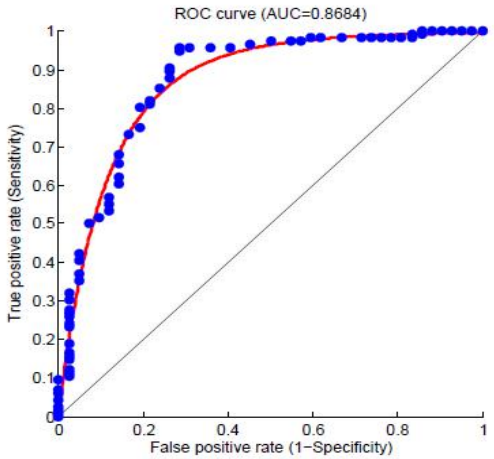
ROC Plot



- Chance Classifier: equal number of false positives and true positives.
- Perfect Classifier: no false positives
- Below red line: worse than random performance

ROC Example

The current standard method of generating ROC curves for NNs is to vary the output node threshold for classification.



The best model is the one with the smallest distance from the perfect classifier.

Image generated using roc.m, Cardillo G. (2008) ROC curve: compute a Receiver Operating Characteristics curve. http://www.mathworks.com/matlabcentral/fileexchange/19950

Steps for Plotting an ROC Curve for NN

- Statistical classifiers have underlying parameters that can be varied to generate ROC curves.
- For nonparametric models like ANN, ROC curves can be generated by varying the output threshold.
- The NN model of interest must be trained before the ROC curve can be generated.
- Output node thresholding:
  - The threshold for the output units of the NN are varied in the interval [0, 1].
  - For each value of the threshold, inputs that produce outputs greater than the threshold are classified into the true class, and others into the false class.
  - A set of operating points (TPR ad FPR) are obtained for varying thresholds and plotted as an ROC curve.

Model Selection

- Given two or more models, how can we pick one to deploy based on ROC curves?
- The model with the curve closest to the perfect classier (top left corner of the plot) has superior classification performance - commits fewer false positives
- Selection of the best model depends on the tradeoff between the `costs' of false positives versus false negatives.
  - Ensuring zero fraud even if it a small percentage of legal transactions get flagged as fraudulent -- better fraud prevention
  - Allowing a small percentage of spam in the inbox, as long as no legitimate emails are flagged as spam --better customer experience

References

[1] Han and Kamber, Data Mining: Concepts and Techniques, 2nd Ed., Morgan Kaufmann, 2006.

[2] Simon Haykin, Neural Networks and Learning Machines, 3rd Ed., 2008.

[3] Lozano et. al., Classier performance evaluation and comparison, ICMLA 2010,

http://www.icmla-conference.org/icmla10/CFP Tutorial les/jose.pdf.

[4] Bishop, Neural Networks for Pattern Recognition, Clarendon Press, Oxford, 1995.

All content and images adapted from reference 2 unless specified otherwise.

Introduction

- Support Vector Machine (SVM) is a binary learning machine with some highly elegant properties
  - has sound mathematical basis
  - very successful real world applications
  - most popularly used algorithm for text processing
- SVMs have some inherent advantages over MLPs trained with backpropagation
  - Search for the optimal solution; not a greedy search
  - Do not get stuck in local minima
  - Do not suffer from the curse of dimensionality

Key Ideas Behind SVMs

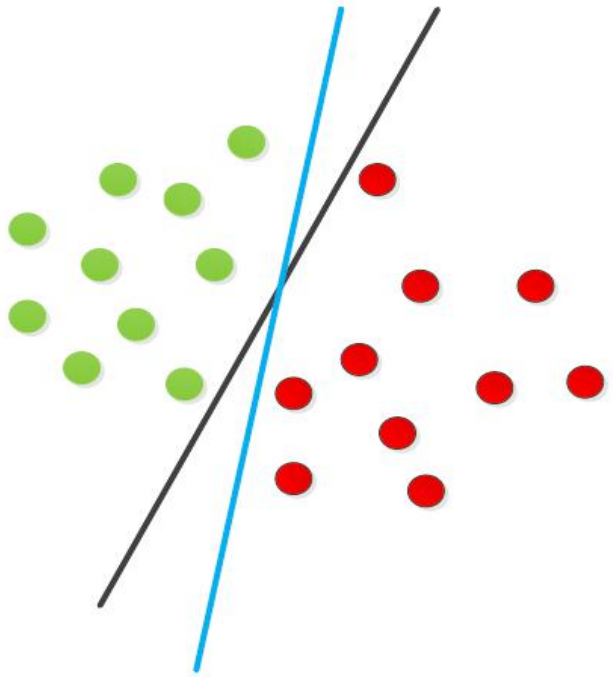
Key Idea

Given a linearly separable training sample, the SVM constructs a separating hyperplane as a decision surface where the margin of separation between positive and negative examples is maximized.

- This basic idea can be extended to non-separable patterns by mapping the original data into a new space.
- Support vectors are data points extracted by the learning algorithm from the training sample itself.

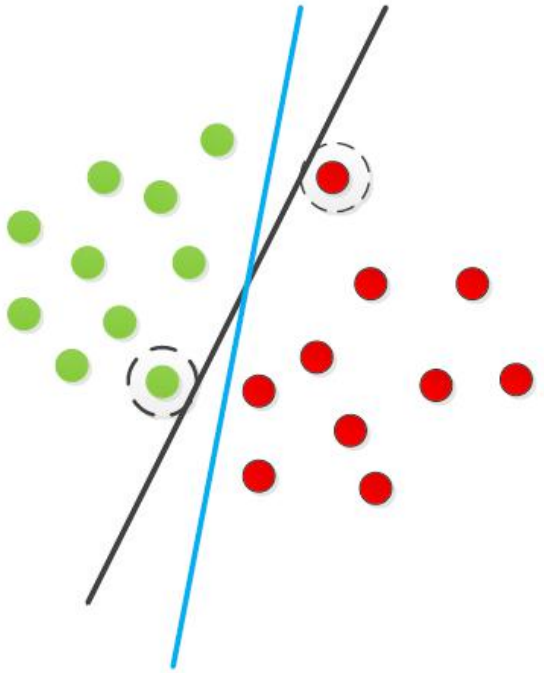
Recap: Separating Hyperplane

- Hyperplane: decision boundary separating data into classes
  - Separating hyperplane in 2-dimensions is a line
  - Multidimensional surface in higher dimensions
- Perceptron boundary is given by $w_1 x + w_2 y = b$
- Solution is not unique.
  - Lots of possible solutions for w_1, w_2 and b.
- Which hyperplane to choose?



Optimal Hyperplane

- SVMs find the optimal separating hyperplane
- How should optimality be determined?
- Can use distance from the data points as a measure
- Which points should affect optimality?
  - All points in a class: Linear regression, Naive Bayes
  - Only the points closest to the decision boundary which are hardest to classify: Support vector machines



Optimal Hyperplane for Linearly Separable Patterns

Consider a set of linearly separable patterns $(\mathbf{x}_i, d_i)_{i=1}^{N}$ (with $d_i \in +1, -1$):

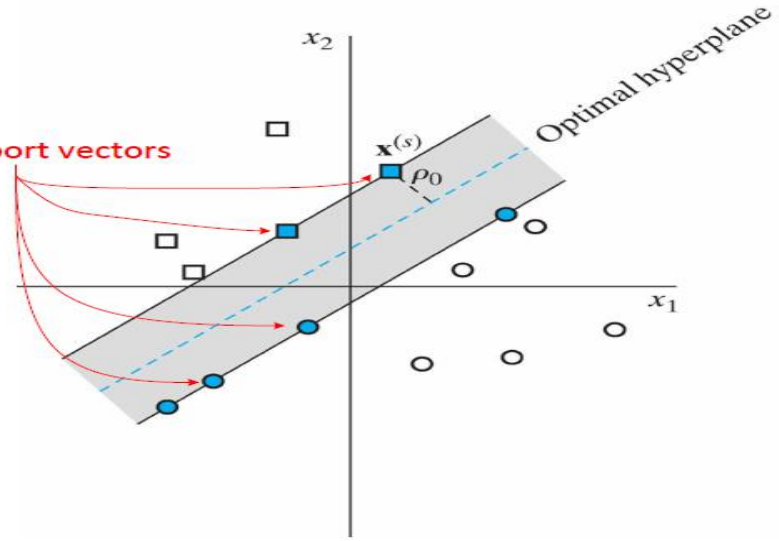
- The separating hyperplane is $\mathbf{w}^T\mathbf{x} + b = 0$:
- The hyperplane separates the two classes according to,

$$\mathbf{w}^T\mathbf{x} + b \geq 0 \text{ for } d_i = +1$$
$$\mathbf{w}^T\mathbf{x} + b < 0 \text{ for } d_i = -1$$

- The distance between the hyperplane defined by a given $\mathbf{w}$ and b and the closest data point is called the *margin of separation* denoted by ρ.
- SVMs seek to find the optimal hyperplane for which ρ is maximized.
- Let $\mathbf{w}_o$ and b_o be the optimum values of the weight vector and bias.

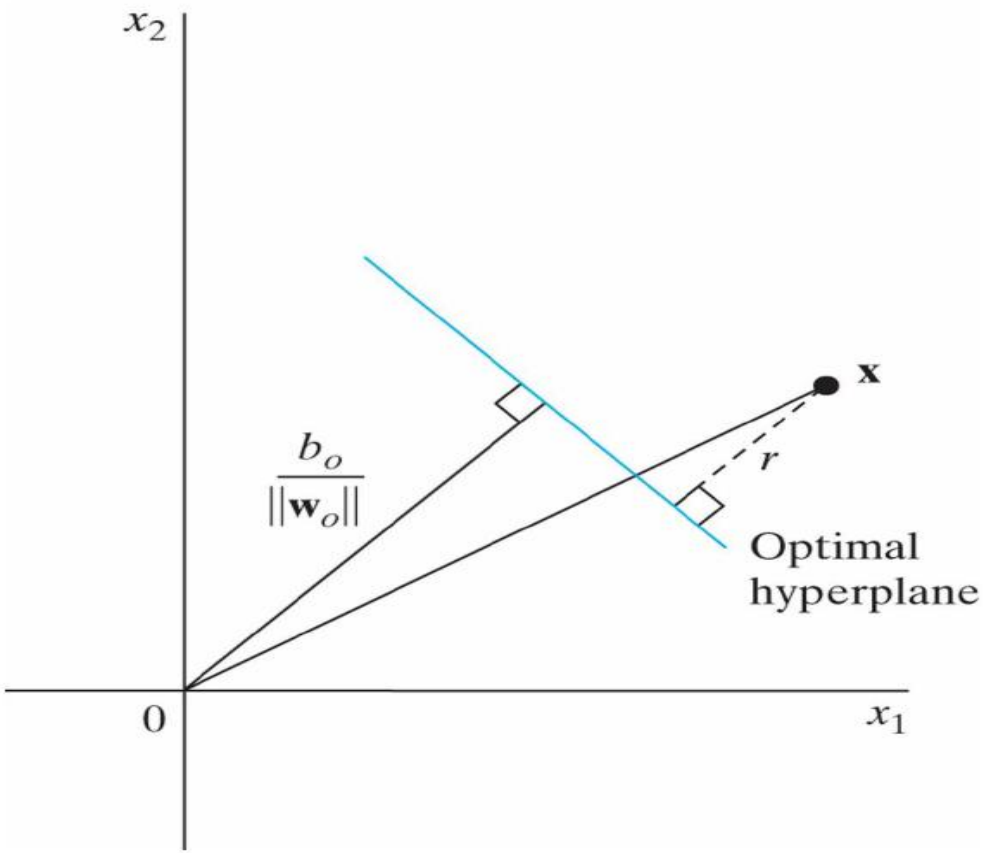
We start with an assumption of linear separability which will be relaxed later.

Illustration of the Optimal Hyperplane



- Support Vectors: input points closest to the separating hyperplane
- Margin of Separation ρ : distance between the separating hyperplane and the closest input point

Distance to the Optimal Hyperplane Illustrated



Distance to the Optimal Hyperplane

- The optimal hyperplane is defined as,

$$\mathbf{w}_o^T \mathbf{x} + b_o = 0$$

- Let r be distance of point $\mathbf{x}$ from the hyperplane.

$$r = \frac{\mathbf{w}_o^T \mathbf{x} + b_o}{\|\mathbf{w}_o\|}$$

- The distance from the origin ($\mathbf{x} = 0$) to the hyperplane is given by $\frac{-b_o}{\|w_o\|}$
- If $b_o > 0$, the origin is on the positive side of the hyperplane.
- If $b_o < 0$, the origin is on the negative side of the hyperplane.
- If $b_o = 0$, the hyperplane passes through the origin.

Optimal Hyperplane and Support Vectors

- The optimal hyperplane is supposed to maximise the margin of separation ρ.
- With that requirement, we need to find $\mathbf{w}_o$ and b_o that satisfy the following constraints:

$$\mathbf{w}_o^T \mathbf{x}_i + b_o \geq +1 \text{ for } d_i = +1$$
$$\mathbf{w}_o^T \mathbf{x}_i + b_o \leq -1 \text{ for } d_i = -1$$

Data points $(\mathbf{x}^{(s)}, d^{(s)})$ for which the above expressions are satisfied as equalities are called the support vectors,

$$\mathbf{w}_o^T \mathbf{x}^{(s)} + b_o = \pm 1$$

- The support vectors directly influence the location of the optimal hyperplane.

Optimal Hyperplane and Support Vectors contd.

- For a support vector $\mathbf{x}^{(s)}$, we know that

$$\mathbf{w}_o^T \mathbf{x}^{(s)} + b_o = \pm 1 \text{ for } d^{(s)} = \pm 1$$

- Distance from a support vector to the optimal hyperplane is,

$$r = \frac{\mathbf{w}_o^T \mathbf{x}^{(s)} + b_o}{\|\mathbf{w}_o\|} = \frac{\pm 1}{\|\mathbf{w}_o\|} \text{ for } d^{(s)} = \pm 1$$

- Thus the margin of separation between two classes is:

$$\rho = 2r = \frac{2}{\| w_o \|}$$

Maximising the margin of separation *between two classes* is equivalent to minimizing the Euclidean norm of the weight vector w_o.

Constrained Optimization Problem

SVM solutions are formulated as convex optimization problems. The four major steps of the formulation are,

- Define the problem of finding the optimal hyperplane as a constrained optimization problem in weight space.
- Construct the Lagrangian function of the optimization problem.
- Derive conditions for optimality.
- Solve the optimization problem using the method of Lagrange multipliers.

Constrained Optimization Problem contd.

- Consider the training data $\mathcal{T} = (\mathbf{x}_i, d_i)_{i=1}^{N}$; find $\mathbf{w}$ and b such that they satisfy:
  - Constraint: $d_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$ for $i = 1, 2, \ldots, N$
  - Minimize Cost function: $\Phi(\mathbf{w}) = \frac{1}{2}(\mathbf{w}^T\mathbf{w})$
- This constrained optimization problem is called the primal problem where the cost function $\Phi(\mathbf{w})$ is a function of $\mathbf{w}$.
- We can solve this problem using the method of Lagrange multipliers.

Method of Lagrange Multipliers

Turn a constrained optimization problem into and unconstrained optimization problem by absorbing the constraints into the cost function, weighted by the Lagrange multipliers.

Example: Adapted from Ravindran et. al., Engineering Optimization: Methods and Applications, Wiley, 2006, pp. 221-222)

- Maximize $F(x) = (x_1)^2 + (x_2)^2$ subject to the constraint $h_1(x) = x_1^2 + x_2^2 = 1$.
- The method of Lagrange multipliers converts this into the following unconstrained problem,

$$L(x; \alpha) = (x_1)^2 + (x_2)^2 - \alpha(x_1^2 + x_2^2 - 1)$$

Lagrange Multipliers contd.

We need to find x_1, x_2, α that minimize

$$L(x;\alpha) = (x_1)^2 + (x_2)^2 - \alpha(2x_1 + x_2 - 2)$$

Set the gradient of L $w.r.t$ to zero

$$\frac{\partial L}{\partial x_1} = 1 - 2\alpha x_1 = 0$$
$$\frac{\partial L}{\partial x_2} = 1 - 2\alpha x_2 = 0$$
$$\frac{\partial L}{\partial \alpha} = x_1^2 + x_2^2 - 1 = 0$$

Solving for x and α: $x_1 = x_2 = \frac{1}{2\alpha}$; Plugging into the third equation: $\left(\frac{1}{2\alpha}\right)^2 + \left(\frac{1}{2\alpha}\right)^2 = 1.$

Thus, $\alpha = \sqrt{\frac{1}{2}}$ and, $(x_1, x_2) = (1/\sqrt{2}, 1/\sqrt{2})$. back

Constructing the Lagrangian Function

Reformulate the constrained optimization problem as a Lagrangian function:

$$J(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{N} \alpha_i \left[d_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 \right]$$

here α_i are nonnegative variables called Lagrange multipliers.

Conditions of Optimality

Differentiating $\partial J(\mathbf{w}, b, \alpha)$ $w.r.t$ $\mathbf{w}$ and b and setting the results to zero,

- From $\frac{\partial J(\mathbf{w},b,\alpha)}{\partial w} = 0$:

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i d_i \mathbf{x}_i.$$

- From $\frac{\partial J(\mathbf{w},b,\alpha)}{\partial b} = 0$:

$$\sum_{i=1}^{N} \alpha_i d_i = 0.$$

Conditions for Optimality contd.

- The optimal solution must meet the Karush-Kuhn-Tucker complementary condition:

$$\alpha_i \Big[d_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 \Big] = 0$$

for all $i = 1, 2, \ldots, N$.

- This means for a multiplier α to be nonzero it must satisfy the above condition.
- Thus, *non-zero* α_i values can be attained only when

$$\Big[d_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 \Big] = 0$$

This means α_i is associated with a support vector $\mathbf{x}^{(s)}$.

Postulating the Dual Problem

Plugging in $\mathbf{w} = \sum_{t=1}^{N} \alpha_t d_t \mathbf{x}_t$ and $\sum_{t=1}^{N} \alpha_t d_t = 0$ back into $J(\mathbf{w}, b, \alpha)$, we get the dual problem.

$$J(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{t=1}^{N} \alpha_t \left[d_t(\mathbf{w}^T\mathbf{x}_t + b) - 1 \right]$$

$$= \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{t=1}^{N} \alpha_t d_t \mathbf{w}^T\mathbf{x}_t - b\sum_{t=1}^{N} \alpha_t d_t + \sum_{t=1}^{N} \alpha_t$$

Calculating $\mathbf{w}^T\mathbf{w} = \sum_{t=1}^{N} \alpha_t d_t \mathbf{w}^T\mathbf{x}_t$ and $\sum_{t=1}^{N} \alpha d_t = 0$

$$J(\mathbf{w}, b, \alpha) = \sum_{t=1}^{N} \alpha_t - \frac{1}{2}\sum_{t=1}^{N} \alpha_t d_t \mathbf{w}^T\mathbf{x}_t$$

$$= \sum_{t=1}^{N} \alpha_t - \frac{1}{2}\sum_{t=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j d_t d_j \mathbf{x}_t^T\mathbf{x}_j$$

Postulating the Dual Problem

Setting,

$$J(\mathbf{w}, b, \alpha) = Q(\alpha).$$

- where all $\alpha_t \geq 0$).
- Thus the primal problem has been transformed into its dual problem.

$$Q(\alpha) = \sum_{t=1}^{N} \alpha_t - \frac{1}{2} \sum_{t=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j d_t d_j \mathbf{x}_t^T \mathbf{x}_j$$

Stating the Dual Problem

- Given the training sample $\mathcal{T} = \left\{ (\mathbf{x}_i, d_i) \right\}_{i=1}^{N}$, find the Lagrange multipliers $\left\{ \alpha_i \right\}_{i=1}^{N}$ that maximize the objective function:

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j \mathbf{x}_i^T \mathbf{x}_j$$

subject to the constraints:

  - $\sum_{i=1}^{N} \alpha_i d_i = 0$
  - $\alpha_i \geq 0$ for all $i = 1, 2, \ldots, N$.

- The problem is stated entirely in terms of the training data $(\mathbf{x}_i, d_i)$, and the dot products $\mathbf{x}_i^T \mathbf{x}_j$ play a key role.

Solution to the Optimization Problem

Once all the optimal Lagrange multipliers $\alpha_{o,i}$ are found, $\mathbf{w}_o$ and b_o can be found as follows:

$$\mathbf{w}_o = \sum_{i=1}^{N} \alpha_{o,i} d_i \mathbf{x}_i$$

and from $\mathbf{w}_o^T \mathbf{x}_i + b_o = d_i$ when $\mathbf{x}_i$ is a support vector:

$$b_o = d^{(s)} - \mathbf{w}_o^T \mathbf{x}^{(s)}$$

when $d^{(s)} = 1$,

$$b_o = 1 - \mathbf{w}_o^T \mathbf{x}^{(s)} = 1 - \sum_{i=1}^{N} \alpha_{o,i} d_i \mathbf{x}_i^T \mathbf{x}^{(s)}$$

SVM for Non-separable Patterns

- For nonseparable patterns, it is not possible to construct a hyperplane without classification errors.
  - Objective is to minimize the probability of classification errors.
- Use Cover's theorem to construct a nonlinear mapping of an input vector to a high-dimensional feature space
- Construct an optimal hyperplane for separating the features identified by the nonlinear transformation.

Soft-margin Classification

- The margin of separation between two classes is said to be soft if it violates the condition,

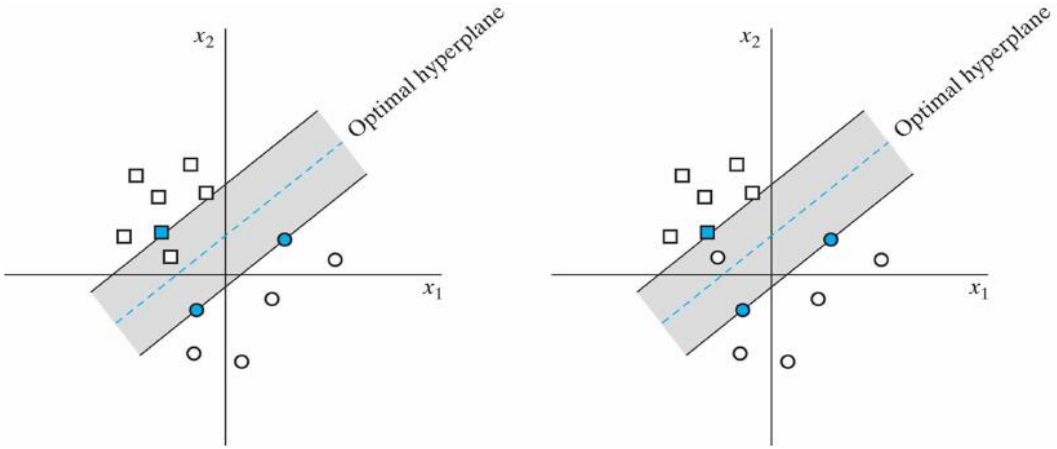
$$d_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$$

- Introduce a new set of variables $\{\xi_i\}_{i=1}^N$ into the definition of the separating hyperplane:

$$d_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i$$

where ξ_i are called *slack variables* which measure the deviation of the data point from the ideal condition of pattern separability.

Soft Margin of Separation



(a) Data point x_i(belonging to class C1, represented by a small square) falls inside the region of separation, but on the correct side of the decision surface.

(b) Data point x_i (belonging to class C2, represented by a small circle) falls on the wrong side of the decision surface.

Primal Problem for Nonseparable Case

- For a training set $\{(\mathbf{x}_i, d_i)\}_{i=1}^{N}$, find the optimal weight vector $\mathbf{w}$ and bias b that satisfy the constraints,

$$d_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0 \text{ for all } i$$

- and minimizes the cost function,

$$\Phi(\mathbf{w}, \xi) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}\xi_i$$

- C is a user-specified positive parameter that controls the complexity of the SVM.
- Higher C implies greater confidence in the training data and lower C implies less confidence in the training data.

Dual Problem for Non-separable Case

- For a training set $\{(\mathbf{x}_i, d_i)\}_{i=1}^{N}$, find the Lagrange multipliers that maximize the objective function,

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j \mathbf{x}_i^T \mathbf{x}_j$$

- subject to the constraints:
  - $\sum_{i=1}^{N} \alpha_i d_i = 0$
  - $0 \leq \alpha_i \leq C$ for all $i = 1, 2, \ldots, N$.

- C is a user-specified positive parameter.

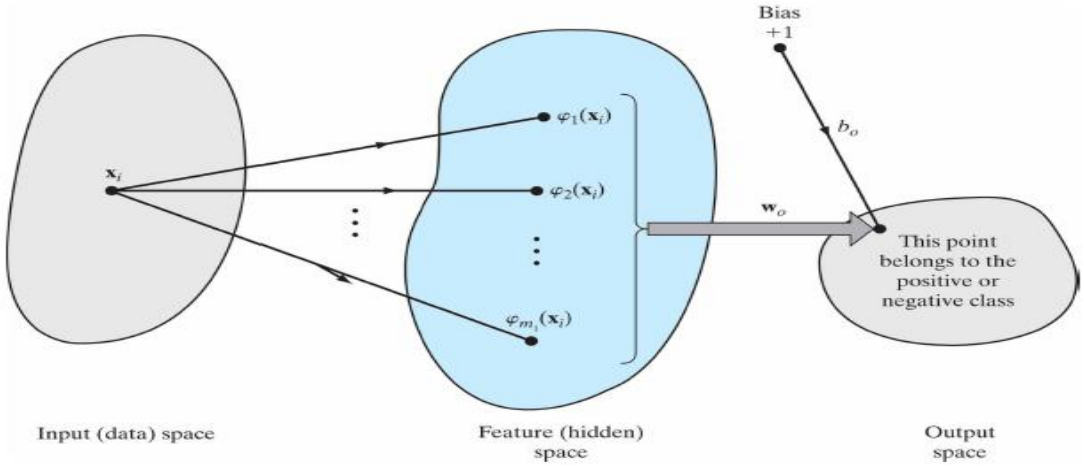
Solution of the Nonseparable Case

- Proceeding similar to the separable case and using the method of Lagrange multipliers, we get the solution:

$$\mathbf{w}_o = \sum_{i=1}^{N_s} \alpha_{o,i} d_i \mathbf{x}_i$$

$$b_o = d_i(1 - \xi_i) - \mathbf{w}_o^T \mathbf{x}_i$$

Underlying Philosophy of SVM for Pattern Classification



SVM operation hinges on two mathematical operations:

(i) Nonlinear mapping from the input space to a high-dimensional feature space.

(ii) Construction of a hyperplane for separating the features into classes.

Determining the Size of the Hidden Space

1. The number of features in the hidden space is determine by the number of support vectors.
2. Thus, SVMs provide an analytical method for finding the optimal size of the hidden space, or the optimal number of hidden neurons.
3. This is a significant improvement over MLPs trained with backpropagation and Radial-basis function networks.
4. Optimality of the classification task is assured!

SVM as a Kernel Machine

- Let input vector $\mathbf{x}$ be m_0-dimensional and let $\varphi(x)$ denote a set of nonlinear functions that map the input to an infinite feature space.
- The decision surface or hyperplane for this case is given as,

$$\mathbf{w}^T \Phi(\mathbf{x}) = 0$$

- Using the steps from linear SVM, we can express the weight vector as,

$$\mathbf{w} = \sum_{i=1}^{N_s} \alpha_i d_i \Phi(\mathbf{x}_i)$$

where N_s is the number of support vectors.

- Thus the decision surface in the output space is,

$$\sum_{i=1}^{N_s} \alpha_i d_i \Phi^T(\mathbf{x}_i)\Phi(\mathbf{x}) = 0.$$

Inner-product Kernel

- The scalar term $\Phi^T(\mathbf{x})\Phi(\mathbf{x}_i)$ is the inner product between two vectors in the feature space.
- Expressing the inner product as a *inner-product kernel* $k(\mathbf{x}, \mathbf{x}_i)$:

$$k(\mathbf{x}, \mathbf{x}_i) = \Phi^T(\mathbf{x})\Phi(\mathbf{x}_i) = \sum_{j=0}^{\infty} \phi_j(\mathbf{x})\phi_j(\mathbf{x}_i)$$

- Using the *kernel* function, the optimal hyperplane may be expressed as:

$$\sum_{i=1}^{N_s} \alpha_i d_i k(\mathbf{x}, \mathbf{x}_i) = 0$$

The Kernel Trick

Some important observations:

1. We don't need to explicitly calculate the weight vector w_o. Specifying the $kernel\ k(x, x_i)$ is sufficient for pattern classification in the output space. This is known as the kernel trick.

2. The above reason is why SVMs are also known as Kernel Machines.

3. Even though we assumed the feature space to be of infinite dimensionality, the equation of the hyperplane has a finite number of terms equal to the number of training patterns.

Selecting Kernel Functions

- Mercer's theorem tells us whether a kernel function can be used to construct an SVM.

- This theorem states that kernel functions $kernel\ k(x, x_i)$ that fulfill certain conditions (continuos, symmetric, positive semi-definite) can be expressed in terms of an inner-product in a nonlinearly mapped feature space.

- The expansion of the kernel functions $kernel\ k(x, x_i)$ permits us to construct a decision surface that is nonlinear in the input space but linear in the output space.

Examples of Kernel Functions Usable for SVMs

TABLE 6.1 Summary of Mercer Kernels

Type of support vector machine	Mercer kernel $k(\mathbf{x}, \mathbf{x}_i), i = 1, 2, ..., N$	Comments
Polynomial learning machine	$(\mathbf{x}^T\mathbf{x}_i + 1)^p$	Power p is specified *a priori* by the user
Radial-basis-function network	$\exp\left(-\frac{1}{2\sigma^2}\|\mathbf{x} - \mathbf{x}_i\|^2\right)$	The width σ^2, common to all the kernels, is specified *a priori* by the user
Two-layer perceptron	$\tanh(\beta_0\mathbf{x}^T\mathbf{x}_i + \beta_1)$	Mercer's theorem is satisfied only for some values of β_0 and β_1

SVM Solution Using the Kernel Function

The $\mathbf{x}^T\mathbf{x}_i$ from the nonlinear solution is replaced with $k(\mathbf{x}, \mathbf{x}_i)$.

- For a training set $\{(\mathbf{x}_i, d_i)\}_{i=1}^N$, find the Lagrange multipliers that maximize the objective function,

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j k(\mathbf{x}, \mathbf{x}_i)$$

- subject to the constraints:
  - $\sum_{i=1}^N \alpha_i d_i = 0$
  - $0 \leq \alpha_i \leq C$ for all $i = 1, 2, \ldots, N$.
- C is a user-specified positive parameter.

XOR Problem

- Objective: To design an SVM to solve the XOR problem.

TABLE 6.2 XOR Problem

Input vector $\mathbf{x}$	Desired response d
$(-1, -1)$	-1
$(-1, +1)$	$+1$
$(+1, -1)$	$+1$
$(+1, +1)$	-1

XOR Problem

Define the following kernel,

$$k(\mathbf{x}, \mathbf{x}_i) = (1 + \mathbf{x}^T \mathbf{x}_i)^2$$
$$\text{with } \mathbf{x} = [x_1, x_2]^T, \mathbf{x}_i = [x_{i1}, x_{i2}]^T$$

Expanding the kernel, we get:

$$k(\mathbf{x}, \mathbf{x}_i) = 1 + x_1^2 x_{i1}^2 + 2x_1 x_2 x_{i1} x_{i2} + x_2^2 x_{i2}^2 + 2x_1 x_{i1} + 2x_2 x_{i2}$$

The input mapped into the feature space is given by,

$$\Phi(\mathbf{x}) = [1, x_1^2, \sqrt{2}x_1 x_2 x_2^2, \sqrt{2}x_1, \sqrt{2}x_2]$$

Similarly,

$$\Phi(\mathbf{x}_i) = [1, x_{i1}^2, \sqrt{2}x_{i1} x_{i2}, x_{i2}^2, \sqrt{2}x_{i1}, \sqrt{2}x_{i2}]^T$$

XOR Problem

The kernel matrix is given by the inner product of $(\mathbf{x}, \mathbf{x}_i)$,

$$\mathbf{K} = \begin{bmatrix} 9 & 1 & 1 & 1 \\ 1 & 9 & 1 & 1 \\ 1 & 1 & 9 & 1 \\ 1 & 1 & 1 & 9 \end{bmatrix}$$

Creating the objective function for the dual form of the optimization problem,

$$\begin{aligned} Q(\alpha) = & \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 \\ & - \frac{1}{2}(9\alpha_1^2 - 2\alpha_1\alpha_2 - 2\alpha_1\alpha_3 + 2\alpha_1\alpha_4 \\ & + 9\alpha_2^2 + 2\alpha_2\alpha_3 - 2\alpha_2\alpha_4 + 9\alpha_3^2 - 2\alpha_3\alpha_4 + 9\alpha_4^2) \end{aligned}$$

XOR Problem

Optimizing $Q(\alpha)$ w.r.t the four Lagrange multipliers yields,

$$\begin{aligned} 9\alpha_1 - \alpha_2 - \alpha_3 - \alpha_4 &= 1 \\ -\alpha_1 + 9\alpha_2 + \alpha_3 - \alpha_4 &= 1 \\ -\alpha_1 + \alpha_2 + 9\alpha_3 - \alpha_4 &= 1 \\ \alpha_1 - \alpha_2 - \alpha_3 + 9\alpha_4 &= 1 \end{aligned}$$

The optimum values of the Lagrange multipliers are,

$$\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = \frac{1}{8}$$

This indicates that all 4 data points are support vectors.

XOR Problem

The optimum value of $Q(\alpha)$,

$$Q_o(\alpha) = \frac{1}{4}$$

Accordingly,

$$\frac{1}{2}\|\mathbf{w}_o\|^2 = \frac{1}{4}$$

or

$$\|\mathbf{w}_o\| = \frac{1}{\sqrt{2}}$$

XOR Problem

The optimum weight vector,

$$\mathbf{w}_o = \frac{1}{8}[-\phi(\mathbf{x}_1) + \phi(\mathbf{x}_2) + \phi(\mathbf{x}_3) - \phi(\mathbf{x}_4)]$$

$$= \frac{1}{8}\left[- \begin{bmatrix} 1 \\ 1 \\ \sqrt{2} \\ 1 \\ -\sqrt{2} \\ -\sqrt{2} \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ -\sqrt{2} \\ 1 \\ -\sqrt{2} \\ \sqrt{2} \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ -\sqrt{2} \\ 1 \\ \sqrt{2} \\ -\sqrt{2} \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ \sqrt{2} \\ 1 \\ \sqrt{2} \\ \sqrt{2} \end{bmatrix} \right] = \begin{bmatrix} 0 \\ 0 \\ -1/\sqrt{2} \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The first element $\mathbf{w}_o$ is the bias $b = 0$.

XOR Problem

The optimal hyperplane is given by,

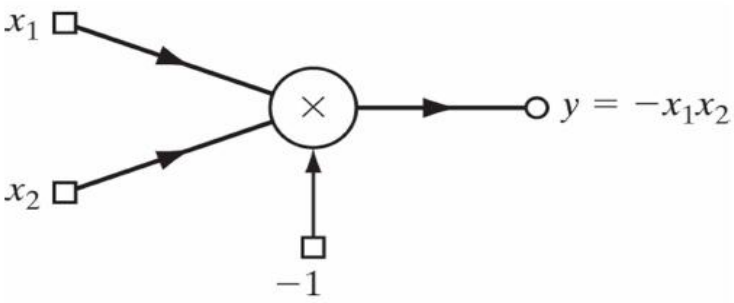
$$\mathbf{w}_o^T \Phi(\mathbf{x}) = 0$$

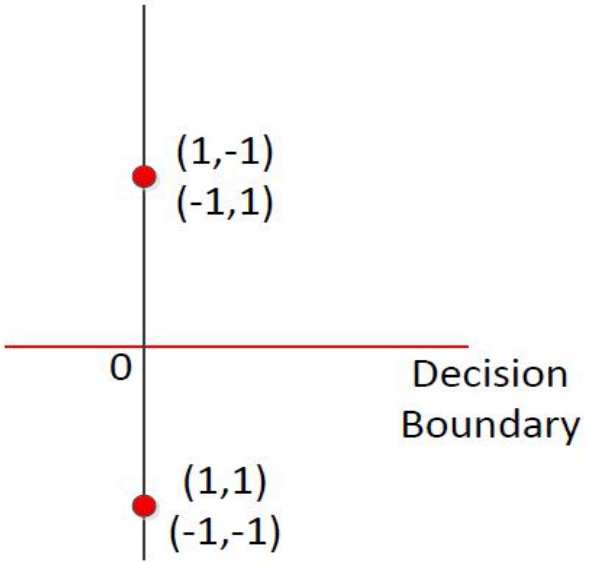
$$\begin{bmatrix} 0 & 0 & -1/\sqrt{2} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \\ \sqrt{2}x_1 \\ \sqrt{2}x_2 \end{bmatrix} = 0$$

$$-x_1x_2 = 0$$

XOR Problem: Solution

The polynomial form of the SVM for the XOR problem is $-x_1x_2 = 0$,





Polynomial machine for solving the XOR problem. Image on the right shows the induced images in the feature space due to the four data points of the XOR problem

Final Project Report and Exam

- Final project report is due on May 4, 2018 at 4:00 PM US CDT
- Final Exam will be posted on blackboard on May 4, 2018 at 4:00 PM US CDT and will be due on May 7, 2018 at 11:59 PM your local time.

Final Project Report and Exam

- Final project report is due on May 5, 2017 at 4:00 PM US CDT
- Final Exam will be posted on blackboard on May 5, 2017 at 4:00 PM US CDT and will be due on May 8, 2017 at 11:59 PM your local time.