Churn Prediction with Support Vector Machine

PIMPRAPAI THAINIAM

EMIS8331 Data Mining

Table of Contents

SMU.

Customer Churn

What are Customer Churn and Customer Retention?

» Customer churn (customer attrition) refers to when customers (subscribers or users) discontinue their subscription to that service.

» Customer retention is the activity a company undertakes to prevent customers from switching to alternative companies or cancelling the services (churn or attrition).

» Customer churn (customer attrition) is an important issue for mature industries.

» Customer churn is easy to define in subscription-based businesses.

SMU

Customer Churn

Subscription-based businesses

- Mobile phone service providers
- Insurance companies
- Cable companies
- Financial services companies
- Internet service providers
- Newspapers
- Magazines

Customer Churn

Why Churn Matters?

» Lost customers must be replaced by new customers.

» New customers are expensive to acquire.

SMU.

Customer Churn

Different Kinds of Churn

- Voluntary churn : The customer decides to quit his contract himself because of unsatisfaction with the quality of service.
- Involuntary churn (Forced churn) : The company discontinues the contract itself rather than customer.
- Expected churn : The customer decides to quit his contract himself because of some reasons other than unsatisfaction, for example customer's relocation.

SMU

Customer Churn

Different Kinds of Churn Model

» Predicting who will leave (Churn prediction) : This method is trying to predict which customer will leave and which will stay. The outcome for each customer will be binary outcome.

» Predicting how long customers will stay : This kind of churn modeling is survival analysis. The outcome will be hazard probability which is the probability that the customer will leave before tomorrow.

SMU

Support Vector Machine

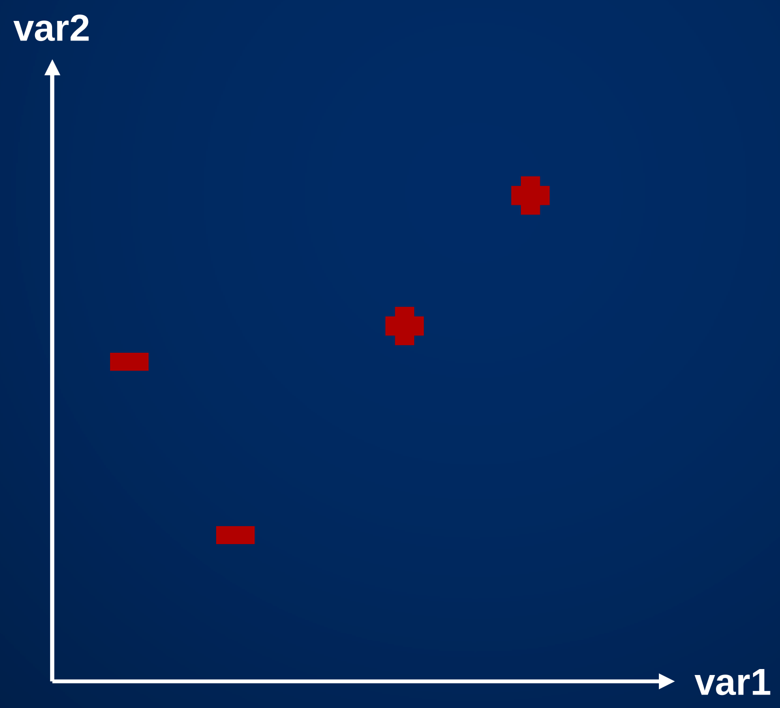
» Support vector machines (SVM) are a group of supervised learning methods that can be applied to classification or regression.

» The original SVM algorithm was invented by Vladimir Vapnik and Alexey Chervonenkis in 1963.

SMU.

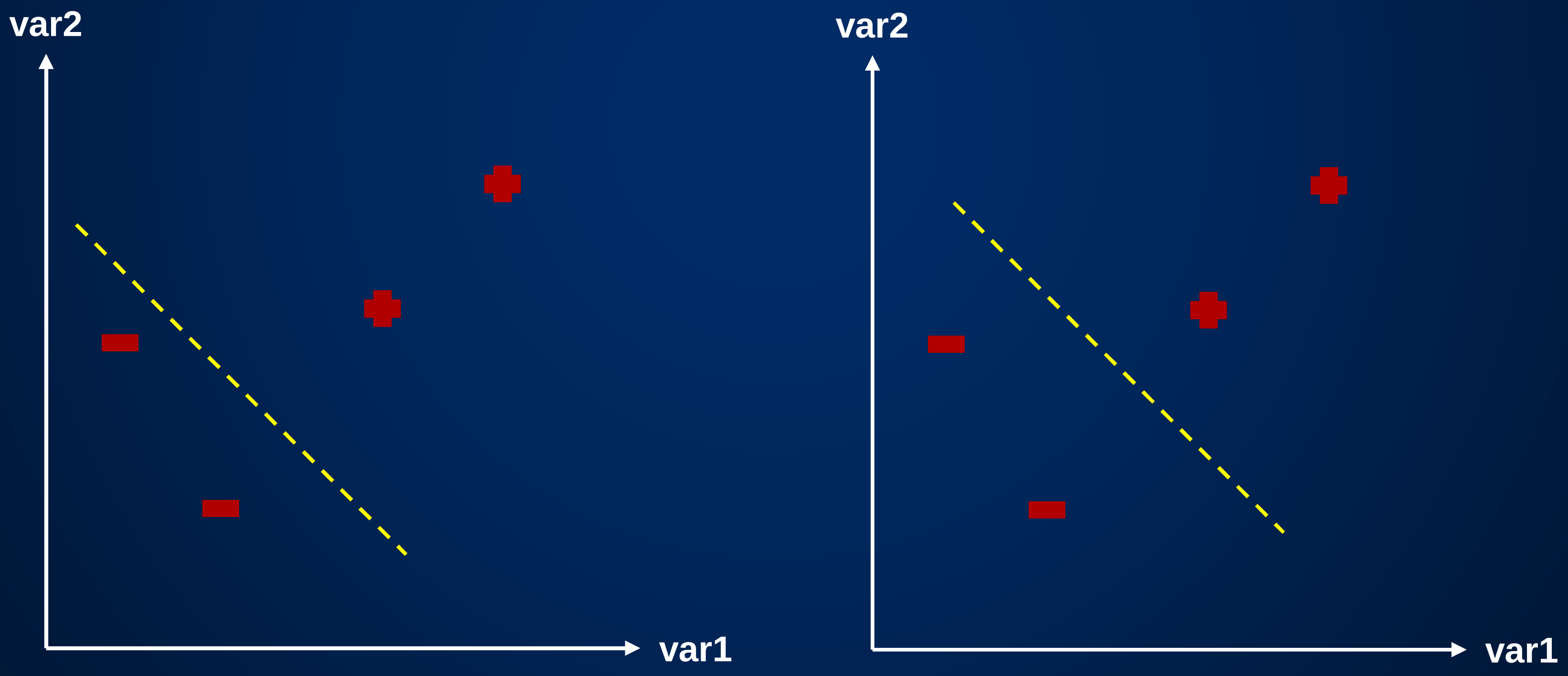
Support Vector Machine

Linearly separable dataset



Support Vector Machine

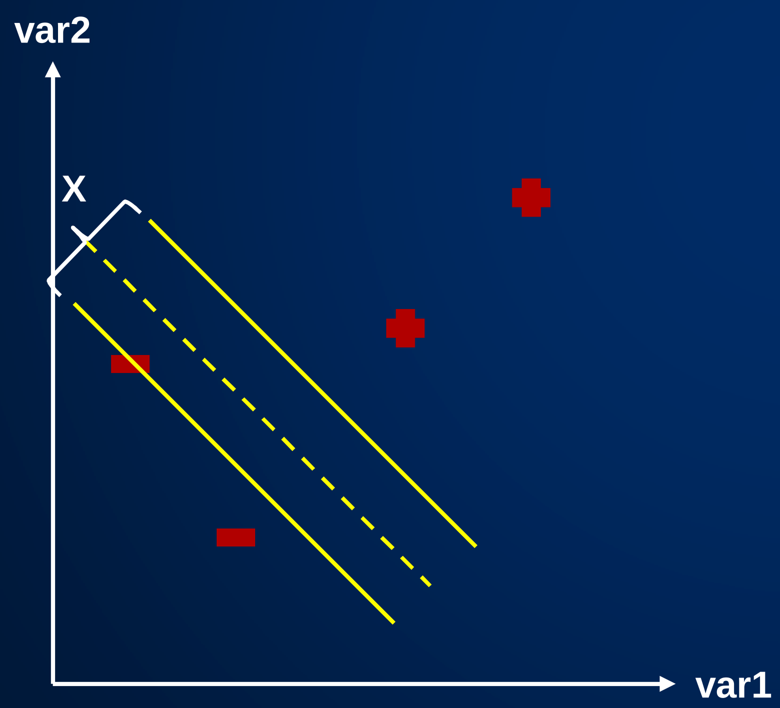
Which line is the best line for classification?



SMU.

Support Vector Machine

Y > X

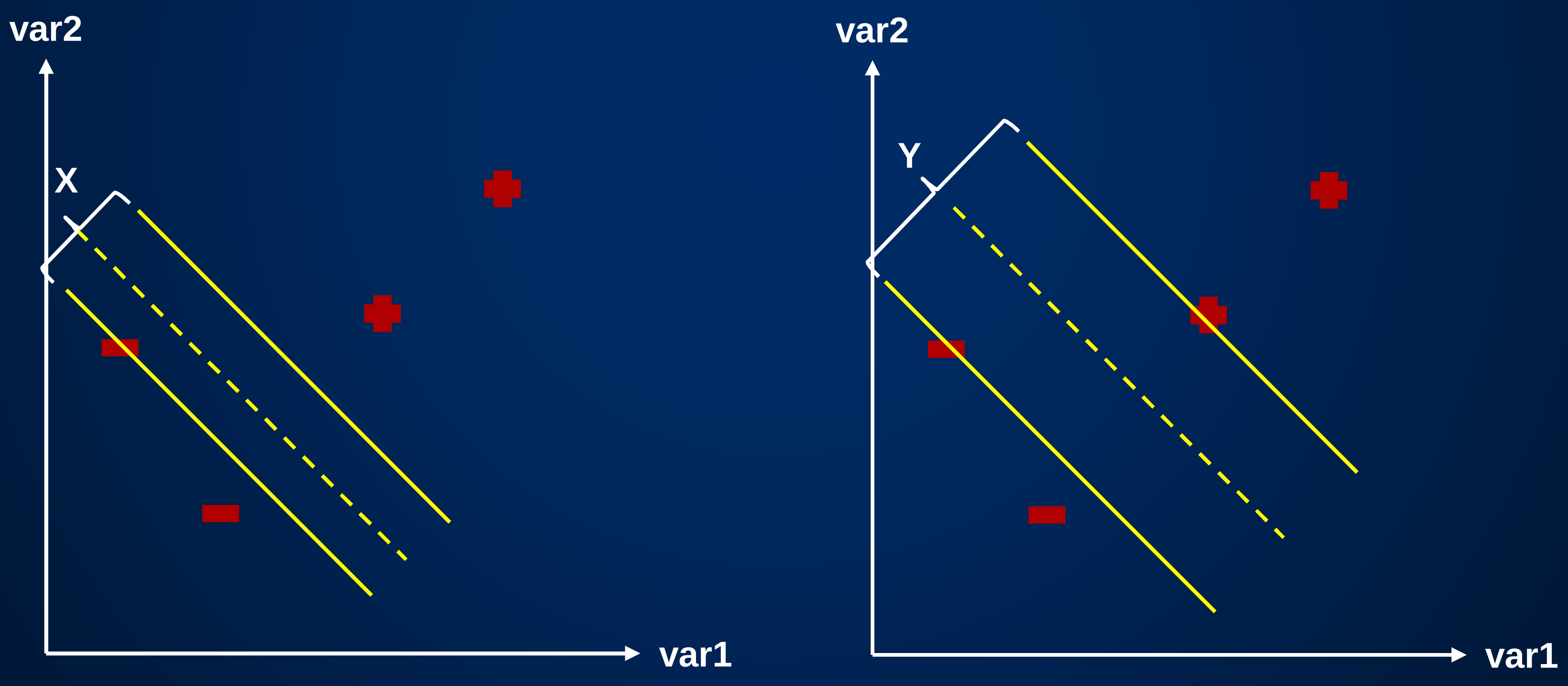


SMU.

Support Vector Machine

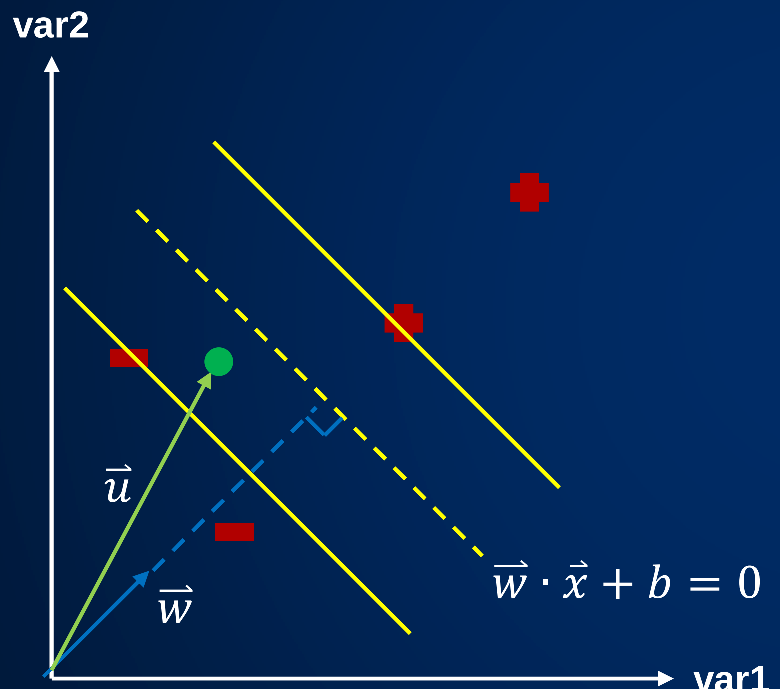
Why is bigger margin better?

Y > **X**



SMU

Support Vector Machine



$$\vec{w} \cdot \vec{u} \geq c$$

$$Given\ c = -b$$

Decision Rule :

$$\vec{w} \cdot \vec{u} + b \geq 0 \quad then +$$

$$\vec{w} \cdot \vec{x}_+ + b \geq 1$$

$$\vec{w} \cdot \vec{x}_- + b \leq -1$$

$$y_i\ such\ that\ \ y_i = +1\ \ for + sample$$

$$y_i = -1\ \ for - sample$$

$$y_i(\vec{w} \cdot \vec{x}_+ + b) \geq 1$$

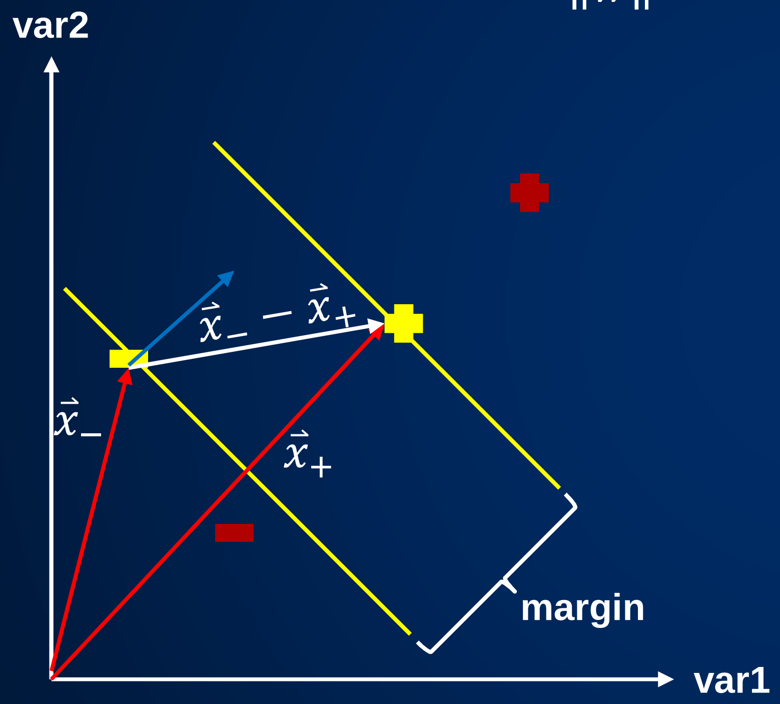
$$y_i(\vec{w} \cdot \vec{x}_- + b) \geq 1$$

$$y_i(\vec{w} \cdot \vec{x}_i + b) - 1 \geq 0$$

SMU.

Support Vector Machine

 $= \frac{\vec{w}}{\|w\|}$, is a unit vector and perpendicular to a median line.



$$y_i(\vec{w} \cdot \vec{x}_i + b) - 1 = 0 \qquad for\ i\ in\ gutter$$

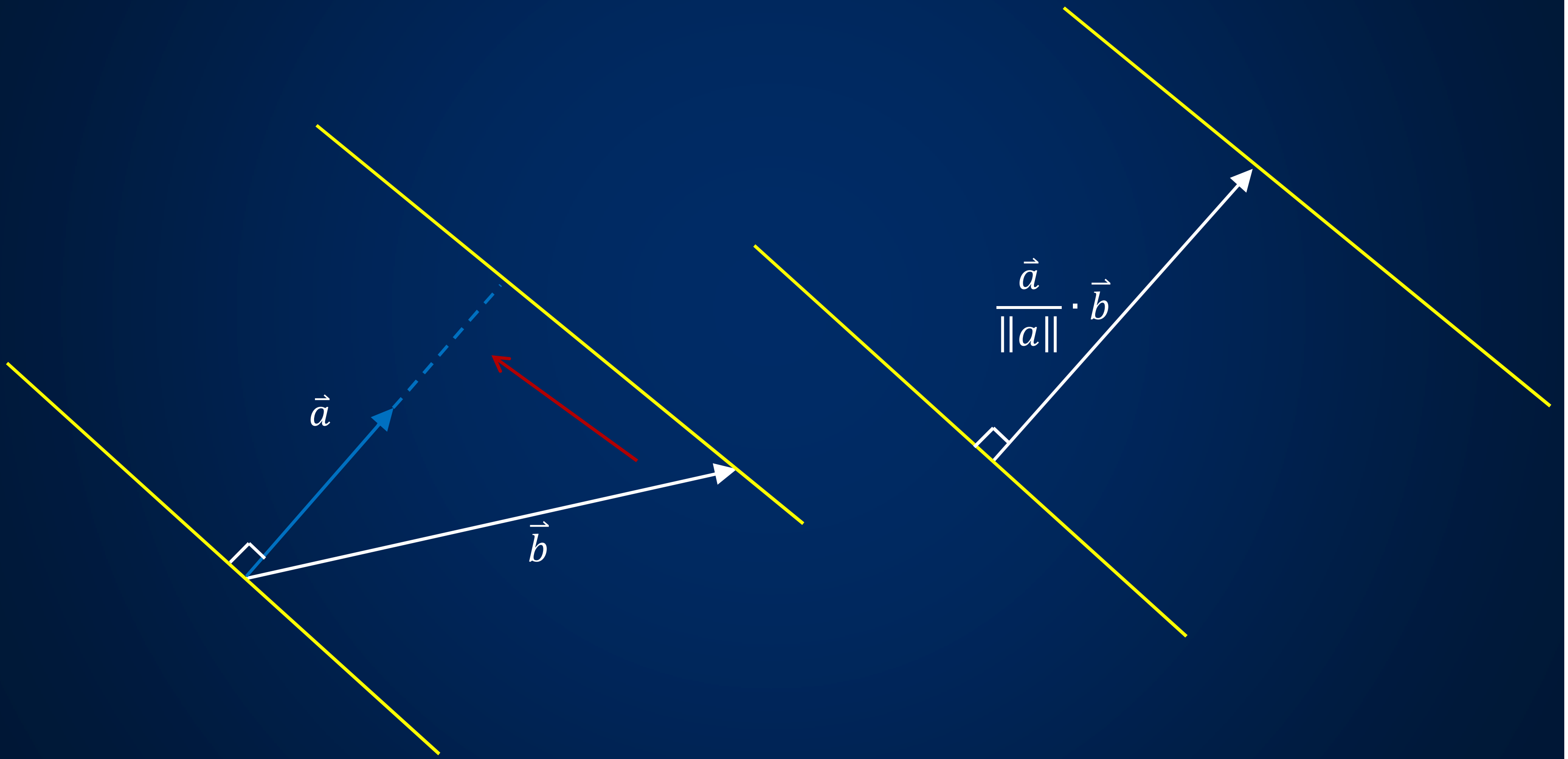
a unit vector

$$Margin = (\vec{x}_+ - \vec{x}_-) \cdot \frac{\vec{w}}{\|w\|}$$

$$= \frac{(1-b) + (1+b)}{\|w\|}$$

$$= \frac{2}{\|w\|}$$

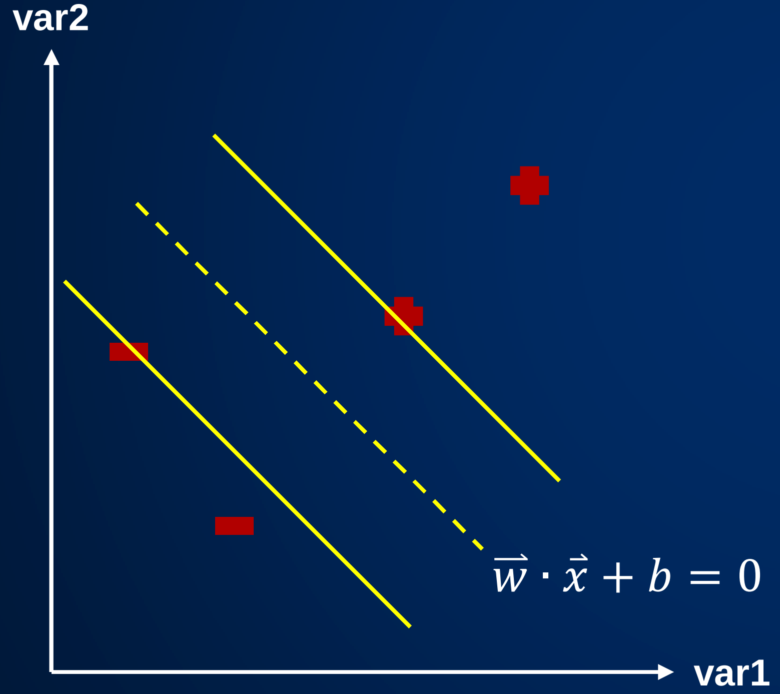
Support Vector Machine



SMU.

Support Vector Machine

$Maximize\ \frac{2}{\|w\|}$ ➡ $Maximize\ \frac{1}{\|w\|}$ ➡ $Minimize\ \|w\|$ ➡ $Minimize\ \frac{1}{2}\|w\|^2$



$Minimize\ \frac{1}{2}\|w\|^2$

$subject\ to$

$y_i(\vec{w}\cdot\vec{x}_i+b)\geq 1 \qquad for\ i=1,2,\ldots,N$

Use Lagragian Method and Quadratic Programming to solve for $\vec{w}$ and b.

$\vec{w}\cdot\vec{x}+b=0$

SMU

Support Vector Machine

Lagrangian Method

$$L(w,b,\alpha)=\frac{1}{2}\|w\|^2-\sum_{i=1}^{N}\alpha_i\,[y_i(\vec{w}\cdot\vec{x}_i+b)-1]$$ ----------- (1)

$$\frac{\partial L}{\partial w}=\vec{w}-\sum_{i=1}^{N}\alpha_i y_i\vec{x}_i=0 \qquad \vec{w}=\sum_{i=1}^{N}\alpha_i y_i\vec{x}_i$$ ----------- (2)

$$\frac{\partial L}{\partial b}=-\sum_{i=1}^{N}\alpha_i y_i=0 \qquad \sum_{i=1}^{N}\alpha_i y_i=0$$ ----------- (3)

Plug (2) and (3) into (1)

$$L(\alpha)=\frac{1}{2}\left(\sum_{i=1}^{N}\alpha_i y_i\vec{x}_i\right)\cdot\left(\sum_{j=1}^{N}\alpha_j y_j\vec{x}_j\right)-\left(\sum_{i=1}^{N}\alpha_i y_i\vec{x}_i\right)\cdot\left(\sum_{j=1}^{N}\alpha_j y_j\vec{x}_j\right)-\sum_{i=1}^{N}\alpha_i y_i+\sum_{i=1}^{N}\alpha_i$$

$$L(\alpha)=-\frac{1}{2}\left(\sum_{i=1}^{N}\alpha_i y_i\vec{x}_i\right)\cdot\left(\sum_{j=1}^{N}\alpha_j y_j\vec{x}_j\right)+\sum_{i=1}^{N}\alpha_i$$

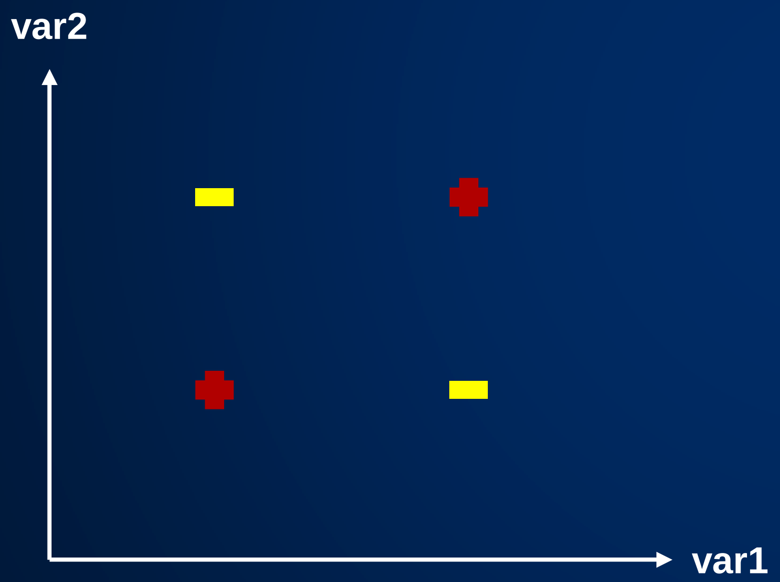
$$L(\alpha)=\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j\,\boxed{\vec{x}_i\cdot\vec{x}_j}$$

⋮

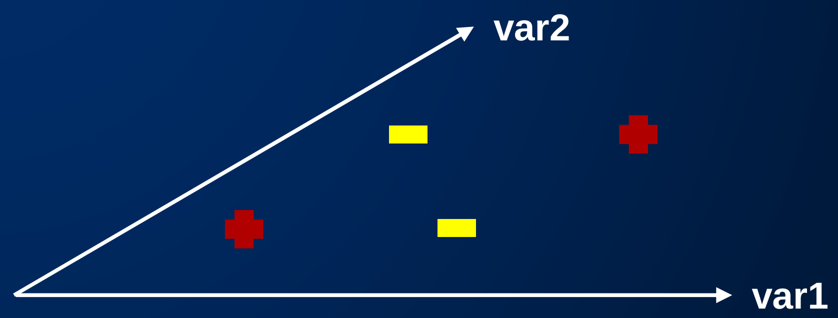
Support Vector Machine

Linearly unseparable dataset

How can we separate this dataset?



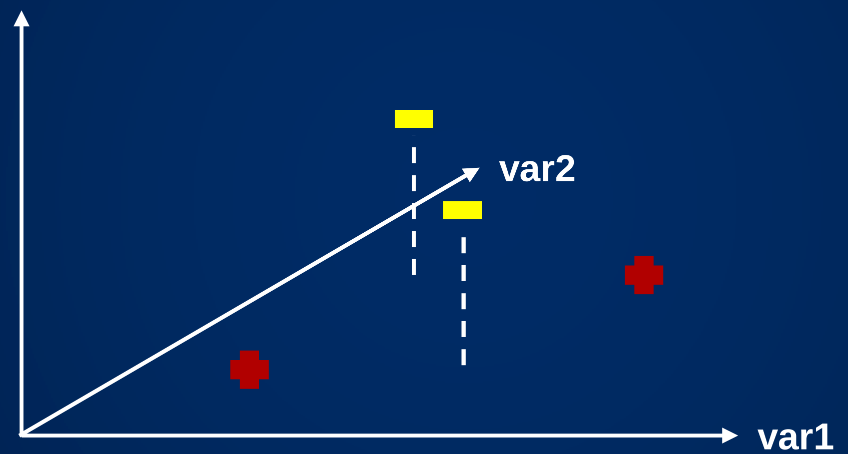
Top-view



Side-view

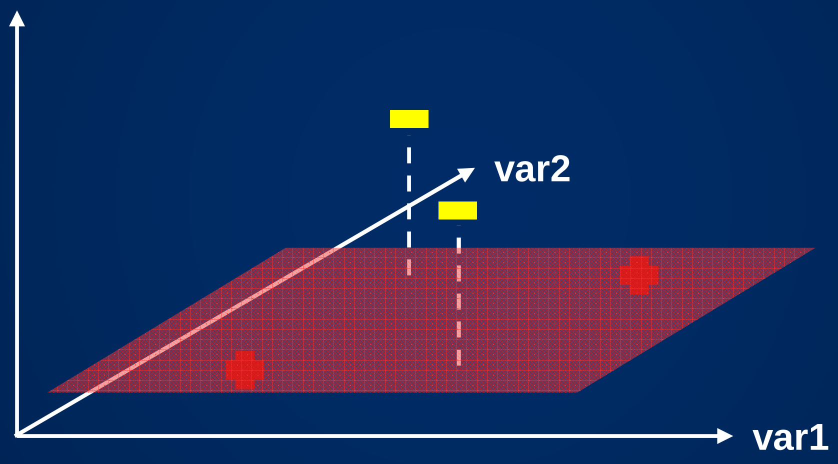
Support Vector Machine

Linearly unseparable dataset



Support Vector Machine

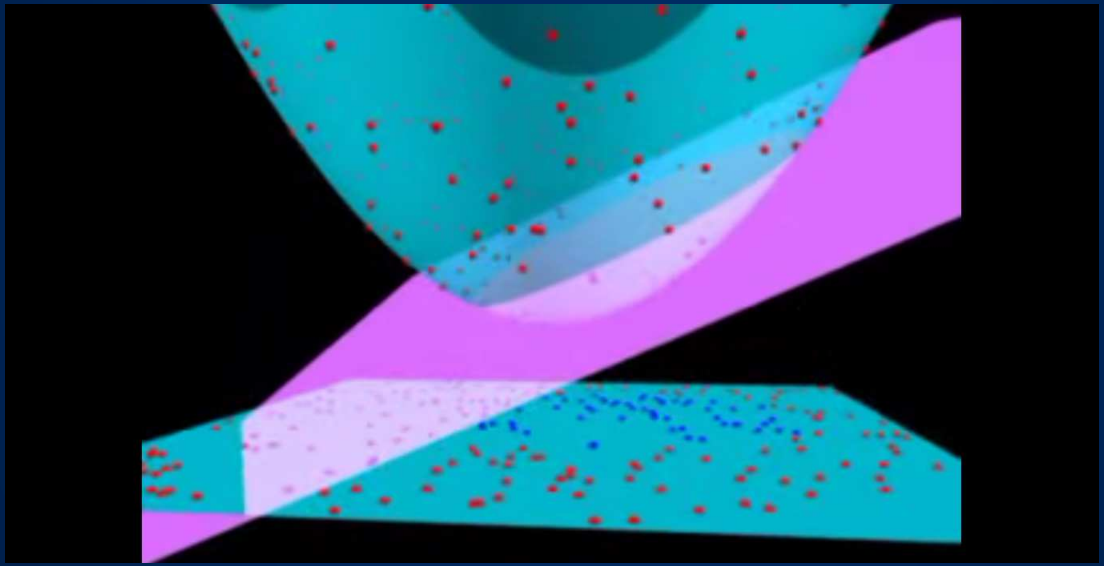
Linearly unseparable dataset



SMU.

Support Vector Machine

Linearly unseparable dataset [3D SVM]



(https://www.youtube.com/watch?v=3liCbRZPrZA)

Transformation function : $\varphi(x, y) = xy(x^2 + y^2)$

This is not kernel function.

SMU

Support Vector Machine

The Kernel Trick

$$L(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \boxed{\vec{x}_i \cdot \vec{x}_j}$$

$$L(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \boxed{\varphi(\vec{x}_i) \cdot \varphi(\vec{x}_j)}$$ (z-dimensional space)

Kernel function : $K\left(\vec{x}_i, \vec{x}_j\right) = \varphi(\vec{x}_i) \cdot \varphi(\vec{x}_j)$

The kernel function is the function that provide us the dot product of the 2 vectors in z space without visiting the z space.

$$L(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \boxed{K\left(\vec{x}_i, \vec{x}_j\right)}$$

SMU

Support Vector Machine

Commonly used kernel functions

Linear Kernel $k(x, y) = x^T y + c$ c : optional constant

Polynomial Kernel $k(x, y) = (\alpha x^T y + c)^d$
α : *slope*, *c* : *constant*, *d* : *polynomial degree*

Exponential Kernel $k(x, y) = exp\left(-\frac{\|x - y\|}{2\sigma^2}\right)$

Laplacian Kernel $k(x, y) = exp\left(-\frac{\|x - y\|}{\sigma}\right)$

Sigmoid Kernel $k(x, y) = tanh(\alpha x^T y + c)$

SMU.

Churn Prediction with Support Vector Machine

Model of Customer Churn Prediction on Support Vector Machine

(by Xia, Guo-en, and Wei-dong Jin, 2008)

» Churn prediction for mobile telecommunication company.

» The average churn rate in mobile telecommunication is 2.2% per month.

» The acquisition cost of a new customer is about $300 - $600.

» The acquisition cost is about 5-6 times of retention cost of an existing customer.

SMU

Churn Prediction with Support Vector Machine

» The 2 datasets used are (1) Mobile telecommunication dataset (2) Home telecommunication carry dataset.

» The researchers used Radial basis kernel function with $u = 0.12$ for dataset (1) and $u = 1$ for dataset (2).

Radial Basis Kernel : $k(x, y) = exp(-\mathrm{u}||x - y||^2)$

» The accuracy rate, hit rate, coverage rate, and lift coefficient from SVM were compared with the artificial neural network, decision tree, logistic regression, and naïve bayesian classifier.

SMU

Support Vector Machine

Customer state	Prediction churn	Prediction non-churn
Actual churn	A	B
Actual non-churn	C	D

$$Accuracy\ Rate = \frac{A + D}{A + B + C + D}$$

$$Hit\ Rate = \frac{A}{A + C}$$

$$Coverage\ Rate = \frac{A}{A + B}$$

$$Lift\ Coefficient = \frac{Hit\ Rate}{Churn\ Rate\ of\ the\ test\ set}$$

SMU

Churn Prediction with Support Vector Machine

Prediction results from dataset (1)

Model type	Accuracy rate	Hit rate	Coverage rate	Lift coefficient
SVM	0.9088	0.8333	0.4018	6.2186
ANN	0.8983	0.7538	0.3625	5.6256
Decision tree C4.5	0.8386	0.3869	0.3437	2.8876
Logistic regression	0.8716	0.6190	0.1160	4.6198
Naive bayesian classifiers	0.8782	0.7142	0.1562	5.3305

Prediction results from dataset (2)

Model type	Accuracy rate	Hit rate	Coverage rate	Lift coefficient
SVM	0.5963	0.7141	0.1620	1.5975
ANN	0.5569	0.7500	0.0139	1.6779
Decision tee C4.5	0.5248	0.4657	0.4236	1.0417
Logistic regression	0.5890	0.7012	0.1412	1.5686
Naive bayesian classifiers	0.5549	0.6250	0.0116	1.3982

THANK YOU

Q & A