Lecture 11-14
Huffman Codes
One Class SVM

Ref: Outlier Analysis, Charu C Agrawal
Ref: Bishop, Christopher M. Pattern recognition and machine learning.
Ref: Tutorial -https://web.mit.edu/zoya/www/SVM.pdf

Source Coding

- Assume a source has alphabet with k symbols
- Symbol s_k occurs with probability p_k
- Average information per symbol is given by entropy

$$H = \sum_{i=1}^{k} p_k \, log_2 \frac{1}{p_k}$$

Huffman Coding

- Calculate symbol probabilities and a lookup table
- Append the test sequence and calculate the bits required per symbol
- Or calculate bits required per window in a sliding window pattern

Equation of a hyperplane

Equation of a straight line $w_1 . x_1 + w_2 . x_2 + b = 0$

$$= > [w_1 \quad w_2] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + b = 0$$

Equation of a straight plane: $w_1 . x_1 + w_2 . x_2 + b = 0$

$$= > [w_1 \quad w_2 \quad w_3] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + b = 0$$

Hence, equation of a generic hyperplane in n dimensional space:

$$= > \mathbf{w}^T . \mathbf{x} + b = 0$$

Linear Classifier

- Obtain equation of the hyperplane that separates the data
- Data points on one side of the plane would give negative value of $\mathbf{w}^T.\mathbf{x} + b$
- Data points on the other side would give positive values of $\mathbf{w}^T.\mathbf{x} + b$

How to represent the classifier constraint?

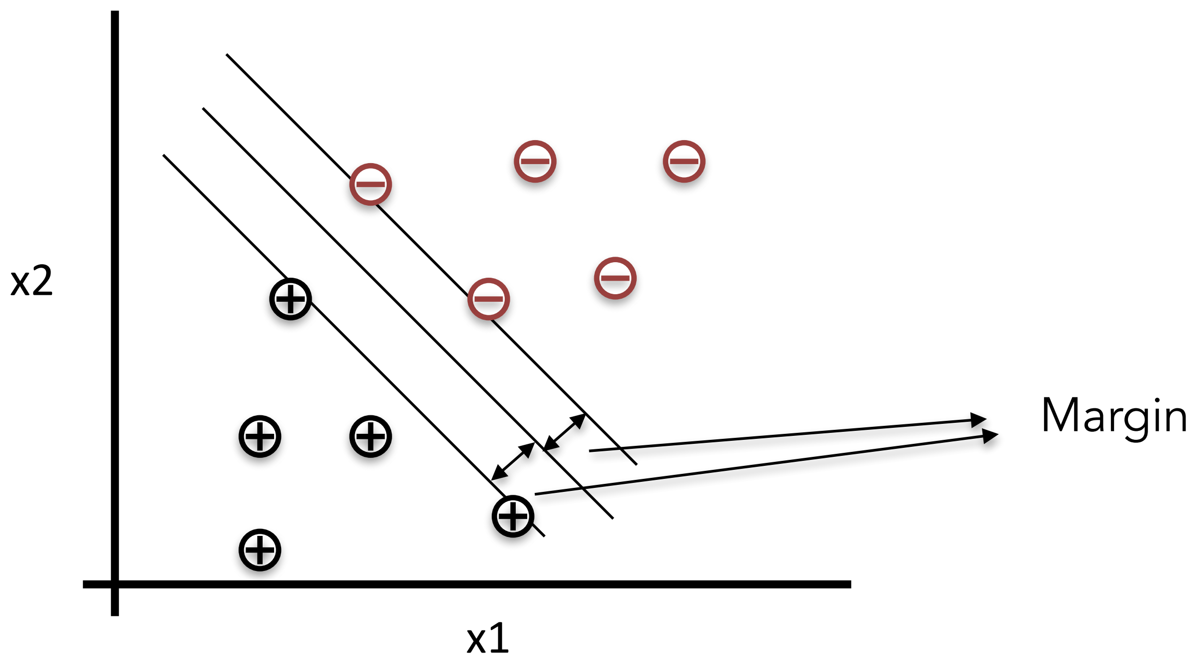
- Let us label one class by +1 and another class by -1
- If the classifier is working correctly, the sign of the hyperplane function and the label should be the same for all n points

$$\forall i \ y_i(\mathbf{w}^T . \mathbf{x_i} + b) \geq 0$$

where y_i are is the label of i^{th} data point.

What is Support Vector Machine?

- Aka maximum margin classifier
- Finds a hyperplane with maximum margin



SVM Constraint

- Enforce positive data points to give a value of more than 1 and negative data points a value of less than -1, in other words,

For positive samples $y_i(\mathbf{w}^T.\mathbf{x_i} + b) \geq 1$

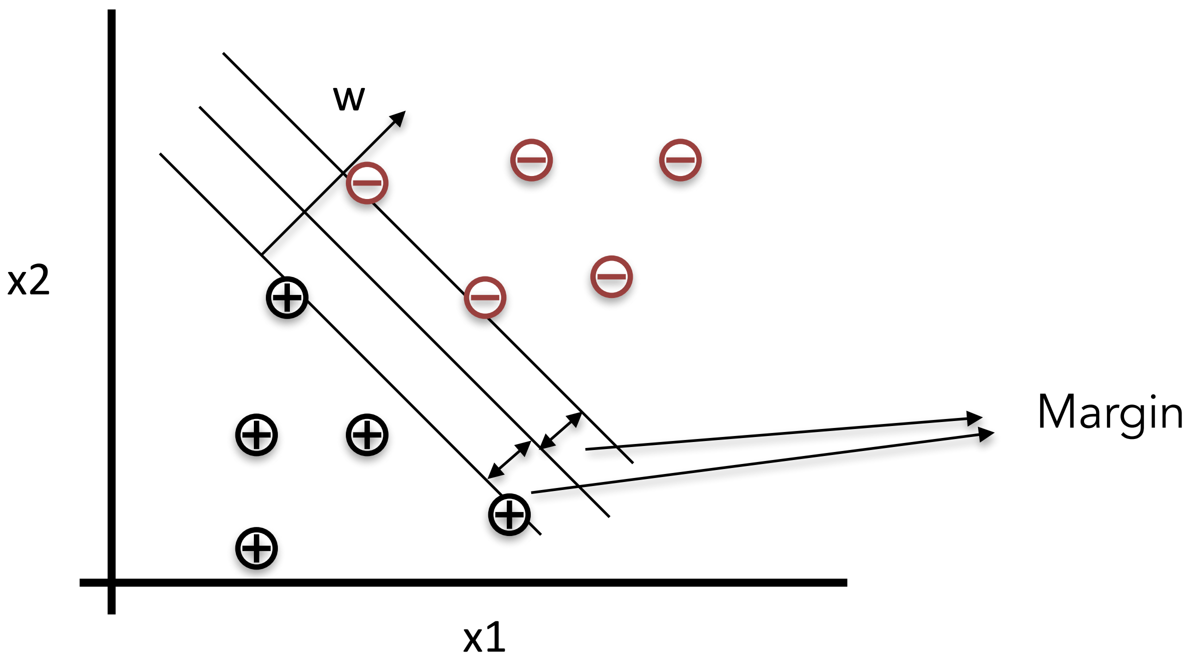
For negative samples $y_i(\mathbf{w}^T.\mathbf{x_i} + b) \leq -1$

Or $\forall i \; y_i(\mathbf{w}^T.\mathbf{x_i} + b) \geq 1$

Calculate margin in terms of w's and b!

- Pick a point p1 on $(\mathbf{w}^T.\mathbf{x}+b)=-1$
- Pick a point p2 on $(\mathbf{w}^T.\mathbf{x}+b)=1$ that is closest to p1
- The closest point will lie in perpendicular direction of the optimal hyperplane
- Distance between these points is the margin
- Which vector is in perpendicular direction?

We know that w is perpendicular to the hyperplane



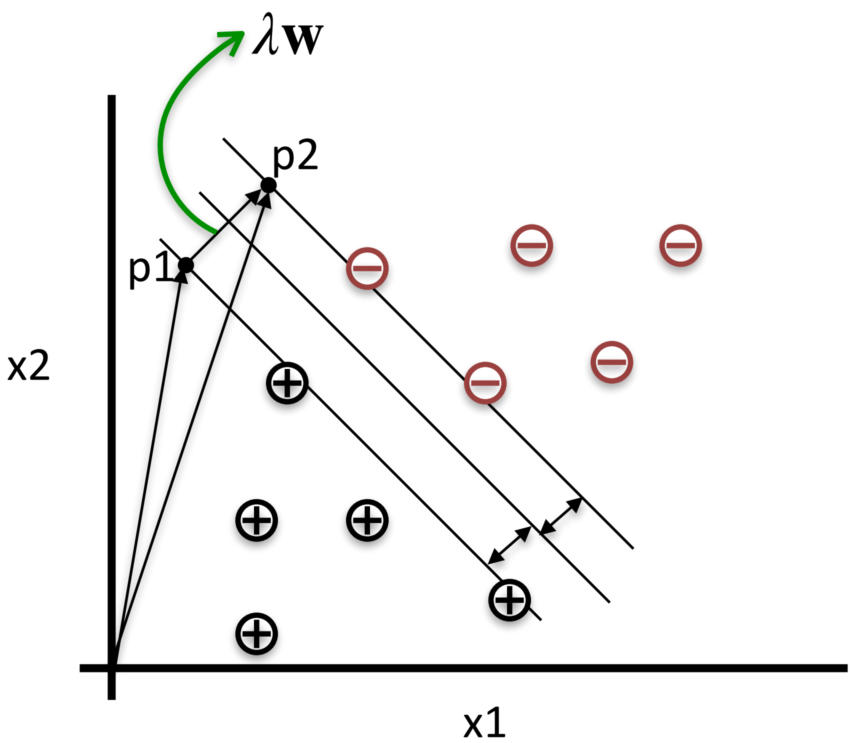
Calculate margin in terms of w's and b!

- Hence, the distance between the hyperplanes can be measured as $\lambda\mathbf{w}$
- But we also know that

$$(\mathbf{w}^T . \mathbf{p_2} + b) - (\mathbf{w}^T . \mathbf{p_1} + b) = 1 - (-1) = 2$$

$$=> \mathbf{w}^T . (\mathbf{p_2} - \mathbf{p_1}) = 2$$

Margin between hyperplanes represented by $(\mathbf{w}^T . \mathbf{x} + b) = -1$ & $(\mathbf{w}^T . \mathbf{x} + b) = 1$



Calculate margin in terms of w's and b!

- Putting $\lambda \mathbf{w}$ for $(\mathbf{p_2} - \mathbf{p_1})$ we get

$$\lambda \mathbf{w}^T \mathbf{w} = 2 => \lambda = \frac{2}{\mathbf{w}^T \mathbf{w}}$$

- Hence the the margin is

$$\frac{2}{\mathbf{w}^T \mathbf{w}} |\mathbf{w}| = \frac{2}{|\mathbf{w}|^2} |\mathbf{w}| = \frac{2}{|\mathbf{w}|}$$

Optimal margin is obtained by maximising $\frac{2}{|\mathbf{w}|}$ or minimising $\frac{|\mathbf{w}|}{2}$ which is equivalent to minimising

$$\frac{|\mathbf{w}|^2}{2} = \frac{\mathbf{w}^T\mathbf{w}}{2}$$

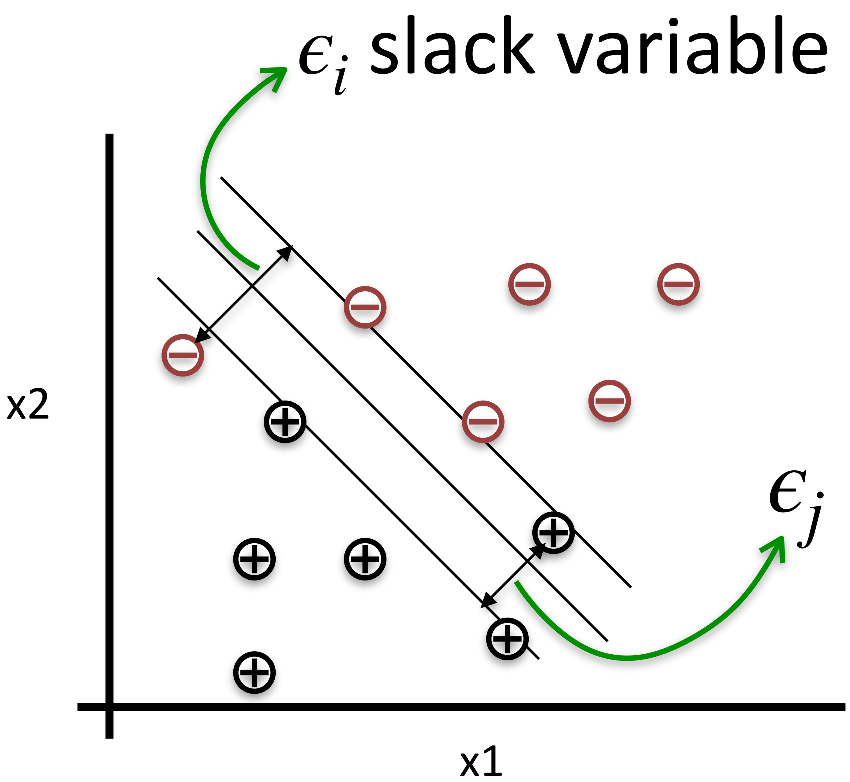
subject to the following constraints:

$$\forall i \ \ y_i(\mathbf{w}^T . \mathbf{x_i} + b) \geq 1$$

What if the data isn't perfectly linearly separably, due to noise or wrong labelling!

- The earlier scheme will not be able to find any hyperplane
- Allow some data points to be wrongly classified but minimise this error as well

Modified Constraint



$$\forall i \; y_i(\mathbf{w}^T . \mathbf{x_i} + b) \geq 1 - \epsilon_i$$

Let us penalise data points on the wrong side of the hyperplanes!

- For data points without error

$$1 - y_i(\mathbf{w}^T . \mathbf{x_i} + b) \leq 0$$

No penalty

- For data points with error

$$1 - y_i(\mathbf{w}^T . \mathbf{x_i} + b) \geq 0$$

Large penalty

How to represent penalty in a single equation?

Total Penalty

- Calculate weighted sum of penalties

$$\sum_{i=1}^{n} \alpha_i(1 - y_i(\mathbf{w}^T . \mathbf{x_i} + b))$$

- Choose weights such that it works for both types of data points, with and without error

$$\sum_{i=1}^{n} \underset{0 \le \alpha_i \le C}{Max} \quad \alpha_i(1 - y_i(\mathbf{w}^T . \mathbf{x_i} + b))$$

Refined Minimisation Function Function

$$L = \underset{\mathbf{w},b}{Min}\ [\frac{\mathbf{w}^T\mathbf{w}}{2} + \sum_{i=1}^{n} \underset{\alpha_i \geq 0}{Max}\ \ \alpha_i(1 - y_i(\mathbf{w}^T . \mathbf{x_i} + b))]$$

How to minimise this function?

Simplified (Dual) Form

$$L = \underset{\alpha}{Max}\ \Big[\underset{\mathbf{w},b}{Min}\Big[\frac{\mathbf{w}^T\mathbf{w}}{2} + \sum_{i=1}^{n} \alpha_i(1 - y_i(\mathbf{w}^T . \mathbf{x_i} + b))\Big]\Big]$$

$$Let \quad J = \Big[\frac{\mathbf{w}^T\mathbf{w}}{2} + \sum_{i=1}^{n} \alpha_i(1 - y_i(\mathbf{w}^T . \mathbf{x_i} + b))\Big]\Big]$$

Obtain w and b in terms of data points and $\alpha_i's$

Set $\frac{\partial J}{\partial \mathbf{w}} = 0$ and $\frac{\partial J}{\partial \mathbf{b}} = 0$

Differentiating Vectors

let $g = \mathbf{w}^T\mathbf{w}$ where $\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \dots \end{bmatrix}$

$$\frac{\partial g}{\partial \mathbf{w}} = \begin{bmatrix} \frac{\partial g}{\partial \mathbf{w_1}} \\ \frac{\partial g}{\partial \mathbf{w_2}} \\ \dots \end{bmatrix}$$ but $g = \mathbf{w}^T\mathbf{w} = |\,\mathbf{w}\,|^2 = w_i^2 + w_2^2 \dots$ so

$$\frac{\partial g}{\partial \mathbf{w}} = \begin{bmatrix} \frac{\partial(w_1^2 + w_2^1 \dots)}{\partial \mathbf{w_1}} \\ \frac{\partial(w_1^2 + w_2^1 \dots)}{\partial \mathbf{w_2}} \\ \dots \end{bmatrix} = \begin{bmatrix} 2w_1 \\ 2w_2 \\ \dots \end{bmatrix} = 2\mathbf{w}$$

Similarly $$\frac{\partial(\mathbf{w}^T\mathbf{X})}{\partial \mathbf{w}} = \begin{bmatrix} \frac{\partial(w_1^2 + w_2^1 \dots)}{\partial \mathbf{w_1}} \\ \frac{\partial(w_1 \,.\, x_1 + w_2 \,.\, x_2 \dots)}{\partial \mathbf{w_2}} \\ \dots \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \dots \end{bmatrix} = \mathbf{x}$$

Obtain w and b in terms of data points and $\alpha_i' s$

Setting $\frac{\partial J}{\partial \mathbf{b}} = 0$ gives us $\sum_{i=1}^{n} \alpha_i y_i = 0$

and setting $\frac{\partial J}{\partial \mathbf{w}} = 0$ gives us $\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i x_i$

Optimisation function to calculate

$$\alpha_i' s$$

$$\max_{\alpha} [\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j]$$

Use quadratic programming to solve it!

Finally calculate b using support vectors

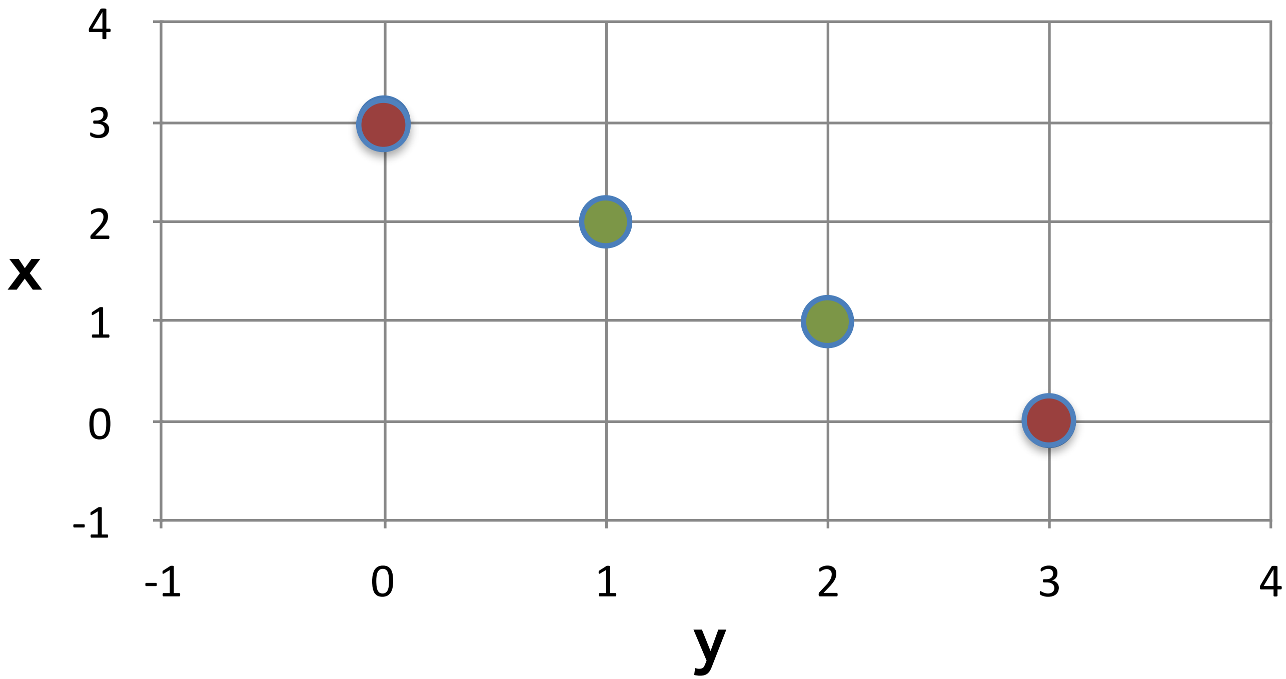
$$\forall s \ y_s(\mathbf{w}^T . \mathbf{x_s} + b) = 1$$

- Find the support vectors by observing the values of Lagrange variables
- Use the above equation to calculate b for each support vector and take average

Decision Function

$$f(\mathbf{x}) = sign(\mathbf{w}^T\mathbf{x} + b) = sign(\sum_{i=1}^{n} \alpha_i y_i \mathbf{x_i}^T\mathbf{x} + b)$$

How to separate the following points using a straight line?



You can linearly separate the data by adding one more feature, x*y to the dataset.

Example

Let there be two points (x_1, x_2) and (y_1, y_2)

Let the transformation functio be

$$\phi(\mathbf{x}) = [\ 1 \quad x_1^2 \quad \sqrt{2}x_1x_2 \quad x_2^2 \quad \sqrt{2}x_1 \quad \sqrt{2}x_2\ ]$$

$$\phi(\mathbf{y}) = [\ 1 \quad y_1^2 \quad \sqrt{2}y_1y_2 \quad y_2^2 \quad \sqrt{2}y_1 \quad \sqrt{2}y_2\ ]$$

$$\begin{aligned}\phi(\mathbf{x})^T\phi(y) &= 1 + x_1^2y_1^2 + 2x_1x_2y_1y_2 + x_2^2y_2^2 + 2x_1y_1 + 2x_2y_2 \\ &= 1 + (x_1y_1 + x_2x_2)^2 + 2x_1y_1x_2y_2 \\ &= (1 + (x_1y_1 + x_2x_2))^2\end{aligned}$$

$$K(\mathbf{x}, \mathbf{y}) = (1 + \mathbf{x}^T\mathbf{y})^2$$

Kernel Trick

$$f(\mathbf{x}) = sign(\mathbf{w}^T\phi(\mathbf{x}) + b) = sign(\sum_{i=1}^{n} \alpha_i y_i \phi(\mathbf{x_i})^T \phi(\mathbf{x}) + b)$$

Transformation

$$= sign(\sum_{i=1}^{n} \alpha_i y_i K(\mathbf{x_i}, \mathbf{x}) + b)$$

- Instead of transforming the data to new space, obtain a function (K()) that can directly calculate the dot product.
- Both decision function as well as the optimisation function to calculate Lagrange multipliers depends on the dot product of feature vectors, not actual transformed feature vectors.

Various Kernels

- Gaussian
- Radial Basis Function (RBF)
- Polynomial
- Sigmoid
- Hyperbolic tangent
- Laplace RBF

One Class SVM

- Treat all data as normal data
- Transform data using kernel trick such that data points are separated from the origin by a big margin
- Find a decision boundary that separates origin and data points and as far as possible from the center

Ref: Schölkopf, Bernhard, et al. "Estimating the support of a high-dimensional distribution." *Neural computation* 13.7 (2001): 1443-1471.

Decision Boundary and Decision Function

- Decision Boundary

$$\mathbf{w}^T \phi(\mathbf{x}) - \rho = 0$$

- Decision Function

$$f(\mathbf{x}) = sign(\mathbf{w}^T \phi(\mathbf{x}) - \rho) = sign(\sum_{i=1}^{n} \alpha_i K(\mathbf{x_i}, \mathbf{x}) - \rho)$$

One Class SVM Constraints

- Normal

$$\mathbf{w}^T\phi(\mathbf{x})_i - \rho \geq 0$$

- With slack variable

$$\mathbf{w}^T\phi(\mathbf{x})_i - \rho \geq \epsilon_i$$

Minimization Function

$$L = \frac{\mathbf{w}^T\mathbf{w}}{2} + \frac{1}{\nu n}\sum_{i=1}^{n} \underset{\alpha_i \geq 0}{Max} \quad \alpha_i(\rho - \mathbf{w}^T.\phi(\mathbf{x_i})) - \rho$$

- Remaining steps are similar to two-class SVM
- The effectiveness of one-class SVM depends on the transformation function's capability to separate origin from normal data points