




ERROR-BASED LEARNING: SUPPORT VECTOR MACHINE

CS576 MACHINE LEARNING



Dr. Jin S. Yoo, Professor
Department of Computer Science
Purdue University Fort Wayne

Reference

- J. D. Kelleher et al., Fundamentals of Machine Learning, Ch 7.4.7
- G. James et al., An Introduction to Statistical Learning, Ch 9

Introduction

- In the **two-class classification problem**, *we try and find a plane that separates the classes in feature space from the training data*.

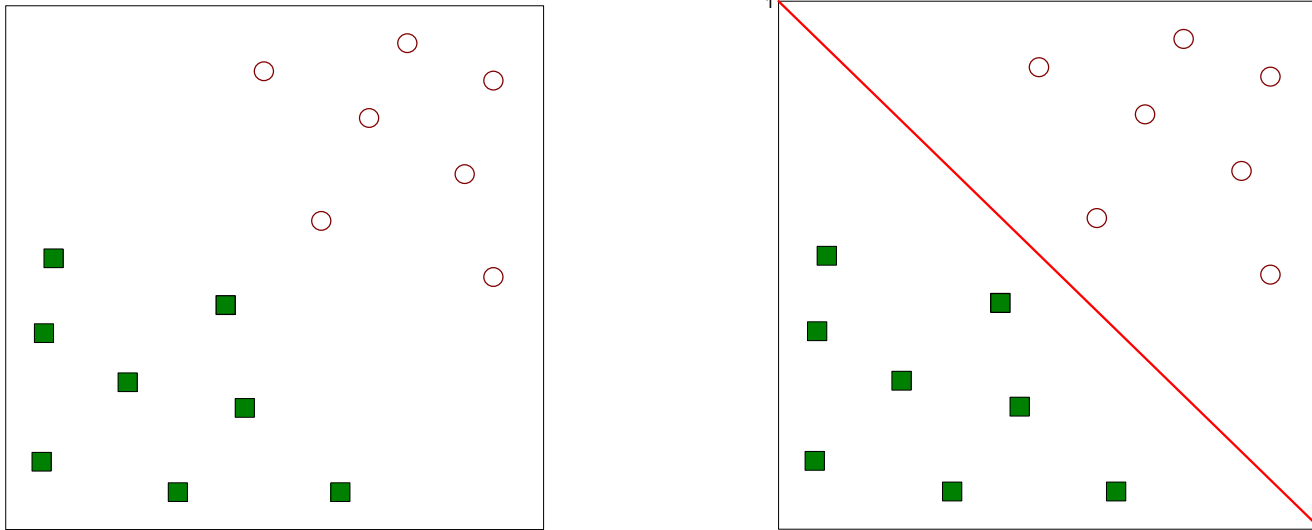


Figure. Data points in two classes in \mathbb{R}^2 . This data can be separated by a linear boundary

- The line (a **hyperplane** in higher dimensions) is called the **decision boundary** (*separating hyperplane*) or **discriminant function**.

Example Dataset

ID	RPM	VIBRATION	STATUS	ID	RPM	VIBRATION	STATUS
1	568	585	good	29	562	309	faulty
2	586	565	good	30	578	346	faulty
3	609	536	good	31	593	357	faulty
4	616	492	good	32	626	341	faulty
5	632	465	good	33	635	252	faulty
6	652	528	good	34	658	235	faulty
7	655	496	good	35	663	299	faulty
8	660	471	good	36	677	223	faulty
9	688	408	good	37	685	303	faulty
10	696	399	good	38	698	197	faulty
11	708	387	good	39	699	311	faulty
12	701	434	good	40	712	257	faulty
13	715	506	good	41	722	193	faulty
14	732	485	good	42	735	259	faulty
15	731	395	good	43	738	314	faulty
16	749	398	good	44	753	113	faulty
17	759	512	good	45	767	286	faulty
18	773	431	good	46	771	264	faulty
19	782	456	good	47	780	137	faulty
20	797	476	good	48	784	131	faulty
21	794	421	good	49	798	132	faulty
22	824	452	good	50	820	152	faulty
23	835	441	good	51	834	157	faulty
24	862	372	good	52	858	163	faulty
25	879	340	good	53	888	91	faulty
26	892	370	good	54	891	156	faulty
27	913	373	good	55	911	79	faulty
28	933	330	good	56	939	99	faulty

Table: A dataset with a categorical feature, STATUS, which indicates 'good' or 'faulty' the day after two measurements were taken.

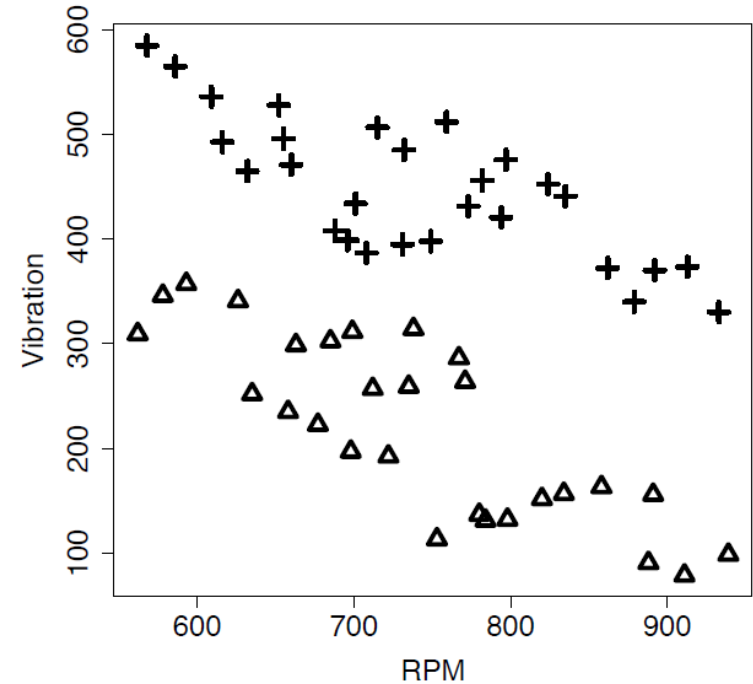
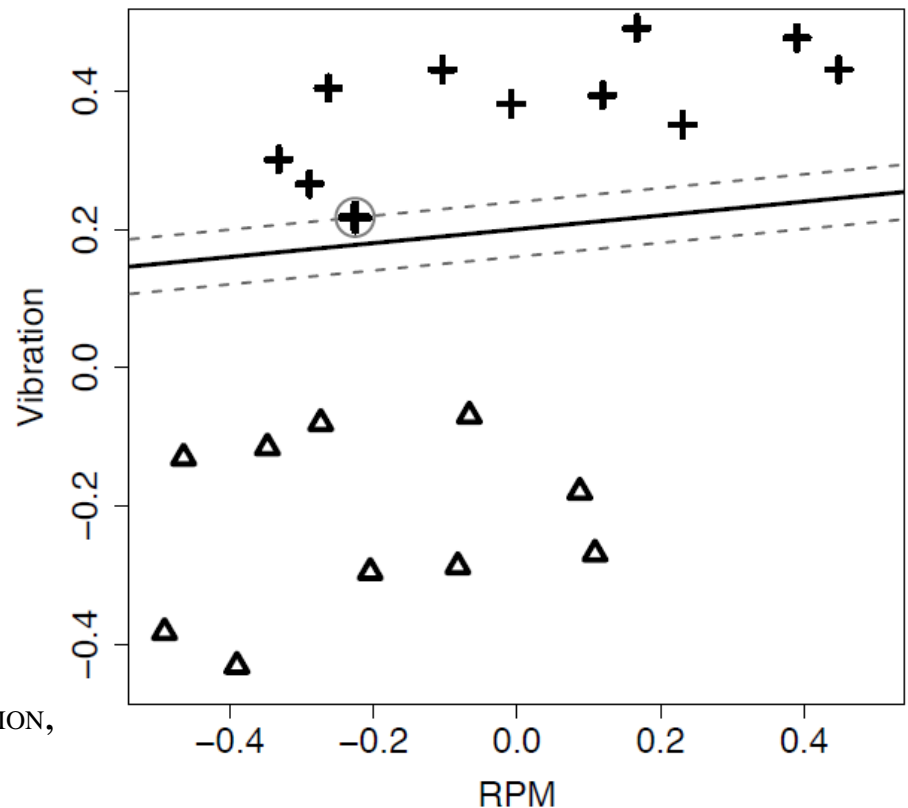


Figure: A scatter plot of the RPM and VIBRATION descriptive features from the generators dataset where 'good' generators are shown as crosses and 'faulty' generators are shown as triangles.

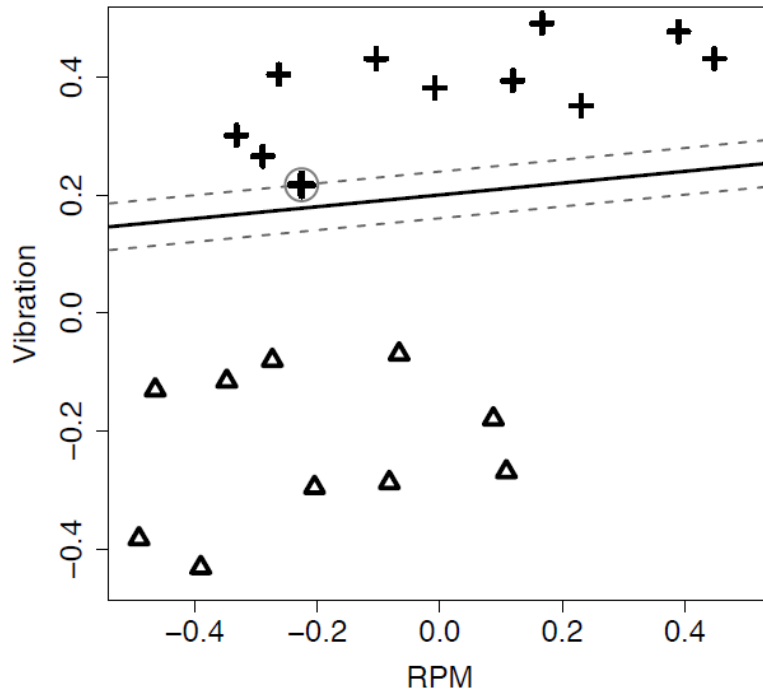
Margin and Margin Extents

- This distance to the nearest training instance, based on perpendicular distance, from the decision boundary is known as the *margin*.
- The dashed lines on either side of the decision boundary which show the extent of the margin are called as the *margin extents*

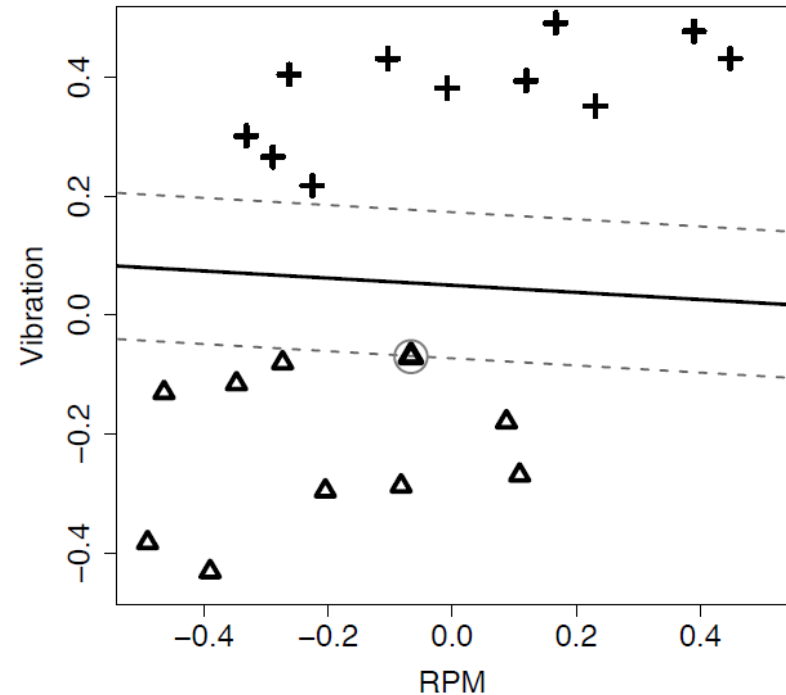
Figure: A small sample of the generators dataset with two features, RPM and VIBRATION, and two target levels, 'good' (shown as crosses) and 'bad' (shown as triangles).



Example: Decision Boundary and Margin Extents



(a) A decision boundary with a very small margin



(b) A decision boundary with a large margin.

- The decision boundary shown on (b) should distinguish between the two target levels much more reliably than (a).

Discriminative Models

- **Discriminative models**, including Logistic Regression and Support Vector Machine, aim to identify the boundary separating different classes within the feature space.
- Unlike **generative models** that predict how data for each class is produced, **discriminative models** prioritize differentiating between classes.
- Support Vector Machine (SVM) (a type of discriminative model) focuses on finding the optimal separating hyperplane that maximizes the margin, ensuring the best distinction between the classes of the target feature.

Support Vector Machine (SVM)

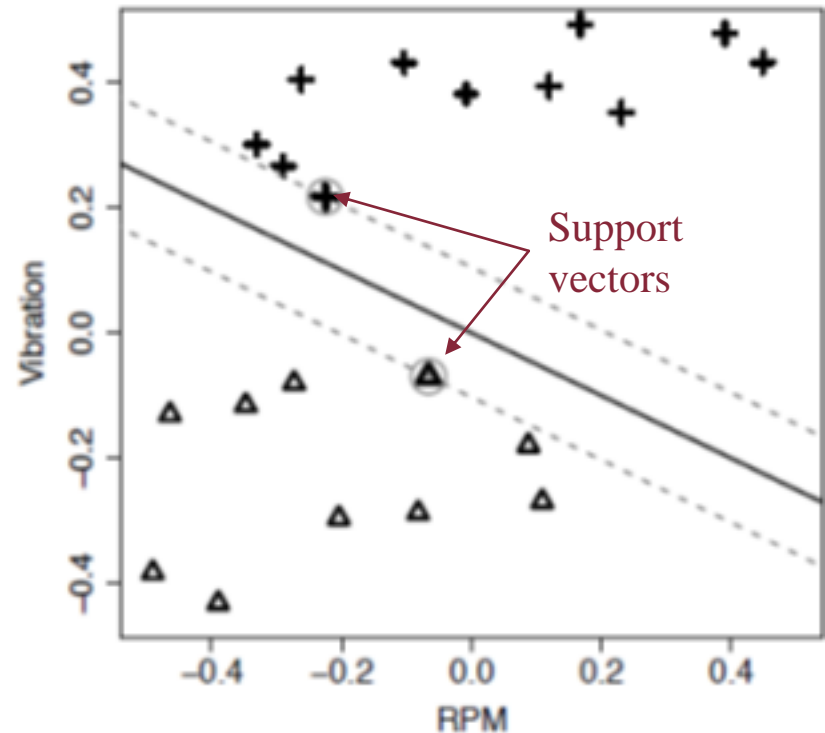
- **Support Vector Machine (SVM)** is a predictive model rooted in error-based learning.
- It was emerged in the 1990s from computer science.
- SVM is a **discriminative classifier** that learns linear or nonlinear decision boundaries in the feature space to separate the classes
- It is an advanced (/ generalized) version of a simple and intuitive classifier called the *maximal margin classifier*.
- SVM is renowned as a top ‘out of the box’ classifier – often performs well with its default settings, without requiring extensive tuning or customization.

SVM vs Logistic Regression

- **Support Vector Machine:** SVMs are designed to locate the hyperplane (or boundary) that creates the largest margin between class clusters in the feature space.
- **Logistic Regression:** Unlike SVMs that prioritize margin optimization, logistic regression seeks a decision boundary that efficiently distinguishes between classes.
 - For binary logistic regression, it calculates the likelihood (or probability) of an input belonging to a specific class, using a set threshold (typically 0.5) to determine the classification.

Support Vectors

- The perpendicular distance from the decision boundary to the closest training instance is referred to as the margin.
- The training data points which lie along the margin extents (dashed lines) known as *support vectors*.
- There will always be at least one support vector for each class of the target feature.
- **Support vectors are the most important instances** because they define the decision boundary.



Defining the Separating Hyperplane

- A **separating hyperplane** can be represented as :

$$w_0 + \mathbf{w} \cdot \mathbf{d} = 0$$

where \mathbf{w} is the weight vector $\langle w_1, w_2, \dots, w_m \rangle$

- When $w_0 + \mathbf{w} \cdot \mathbf{d} = 0$, the data point \mathbf{d} **lies on the hyperplane**.
- The position relative to the hyperplane can be determined by evaluating the sign of the function:

- Data points **lying above the hyperplane** (on the one side of the hyperplane) satisfy:

$$w_0 + \mathbf{w} \cdot \mathbf{d} > 0$$

- Data points **lying below the hyperplane** (on the other side of the hyperplane) satisfy:

$$w_0 + \mathbf{w} \cdot \mathbf{d} < 0$$

Constraints for Training a SVM

- The function $w_0 + \mathbf{w} \cdot \mathbf{d}$ represents the distance of a data point \mathbf{d} from the decision boundary.
- To find a hyperplane that distinguishes between the two target levels, -1 and +1, **the constraints required by the training process** are

$$w_0 + \mathbf{w} \cdot \mathbf{d} \leq -1 \text{ for } t_i = -1, \text{ and}$$

$$w_0 + \mathbf{w} \cdot \mathbf{d} \geq +1 \text{ for } t_i = +1$$

- For simplicity, we can combine these two constraints into a single constraint

$$t_i \times (w_0 + \mathbf{w} \cdot \mathbf{d}) \geq 1$$

Here, t_i is always equal to either -1 or +1 in our case.

Example

- Classes: *good* (+1), *faulty*(-1)
- The support vectors are highlighted in the figures
- The two different decision boundaries which satisfy the constraints:
- For $t_i = -1, w_0 + \mathbf{w} \cdot \mathbf{d} \leq -1$
- For $t_i = +1, w_0 + \mathbf{w} \cdot \mathbf{d} \geq +1$

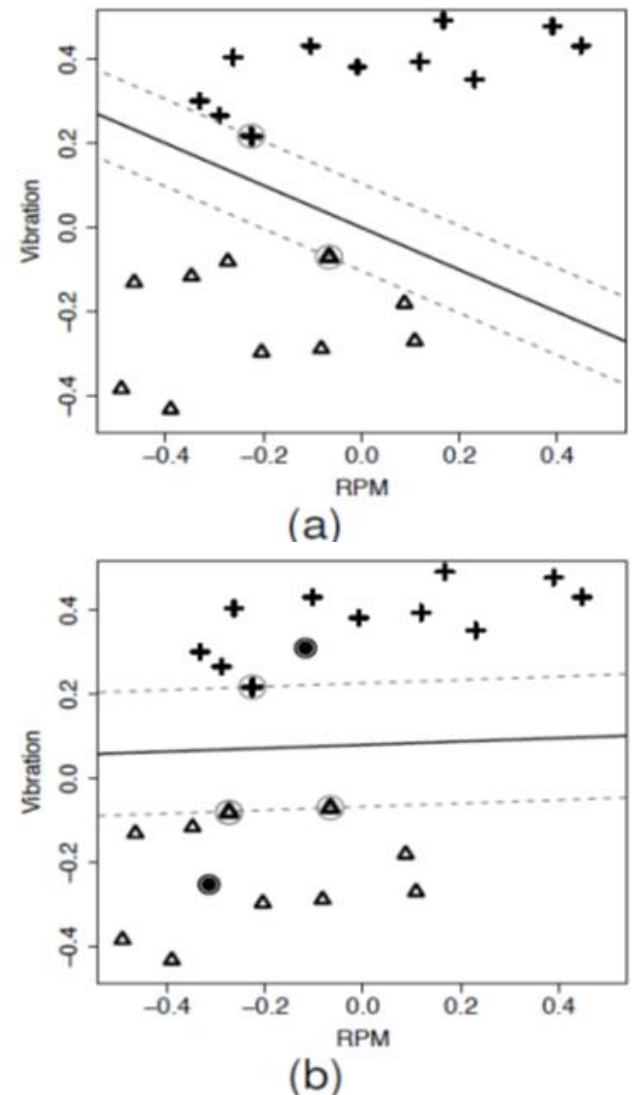


Figure: two target levels, 'good' (shown as crosses) and 'bad' (shown as triangles).

Optimization Criterion for Training a SVM

- We have an optimization criterion which allows us to choose between multiple different decision boundaries that satisfy the constraint given.
- The **optimization criterion** defined in terms of the **perpendicular distance** from any instance to the decision boundary is :

$$dist(\mathbf{d}) = \frac{abs(w_0 + \mathbf{w} \cdot \mathbf{d})}{\|\mathbf{w}\|}$$

where $w_0 + \mathbf{w} \cdot \mathbf{d}$ represents the distance of a data point \mathbf{d} from the decision boundary, and

$\|\mathbf{w}\|$ is known as the **Euclidean norm** of \mathbf{w} and is calculated as $\|\mathbf{w}\| = \sqrt{\mathbf{w}[1]^2 + \mathbf{w}[2]^2 + \dots + \mathbf{w}[m]^2}$ and represents the “strength” or “length” of the vector \mathbf{w}

Optimization Criterion (cont.)

$$dist(\mathbf{d}) = \frac{abs(w_0 + \mathbf{w} \cdot \mathbf{d})}{\|\mathbf{w}\|}$$

- In the context of SVM, the support vectors are defined as $abs(w_0 + \mathbf{w} \cdot \mathbf{d}) = 1$ in a mathematical way
 - The decision boundary in SVM is defined by $(w_0 + \mathbf{w} \cdot \mathbf{d}) = 0$
 - The two margin hyperplanes are determined by $w_0 + \mathbf{w} \cdot \mathbf{d} = 1$ and $w_0 + \mathbf{w} \cdot \mathbf{d} = -1$
- So, the **distance** from any instance along the margin extends to the decision boundary is $\frac{1}{\|\mathbf{w}\|}$, and **the size of the margin** is $\frac{2}{\|\mathbf{w}\|}$, because the margin is symmetrical to either side of the decision boundary,

Goal of Training Support Vector Machine

- The **goal** of training a support vector machine is

- **maximize** $\frac{2}{\|\mathbf{w}\|}$ \leftarrow minimizing $\|\mathbf{w}\|$

- , subject to the constraint

$$t_i + (\mathbf{w}_0 + \mathbf{w} \cdot \mathbf{d}) \geq 1 \quad \leftarrow \text{ensuring every data points}$$

should be the correct side of the hyperplane

- With the objective function and constraints in place, training a SVM is framed as **a constrained quadratic optimization problem**
 - This type of problem is defined in terms of (1) **a set of constraints** and (2) **an optimization criterion**
- We are trying to minimize a quadratic function in terms of $\|\mathbf{w}\|$.

Support Vector Machine Model

- After solving the optimization problem, the SVM can be represented by the model equation,

$$M_{\alpha, w_0}(q) = \sum_{i=1}^s (t_i \times \alpha_i \times (\mathbf{d}_i \cdot \mathbf{q}) + w_0)$$

- $(\mathbf{d}_1, t_1), \dots, (\mathbf{d}_s, t_s)$ are **s support vectors** (instances composed of descriptive features and a target feature)
 - \mathbf{d}_i : The set of descriptive features of the i th support vector.
 - t_i : The class label of the i th support vector.
- α_i : The **Lagrange multiplier** for the i th support vector which is determined during the training
- \mathbf{q} is the set of descriptive features for a query instance
- $\mathbf{d}_i \cdot \mathbf{q}$: The dot product between the i th support vector and the query point \mathbf{q}
- The bias term w_0 is added at the end.
- A support vector machine model **uses only the support vectors, $(\mathbf{d}_1, t_1), \dots, (\mathbf{d}_s, t_s)$** , to define the separating hyperplane and hence to make the actual model predictions.

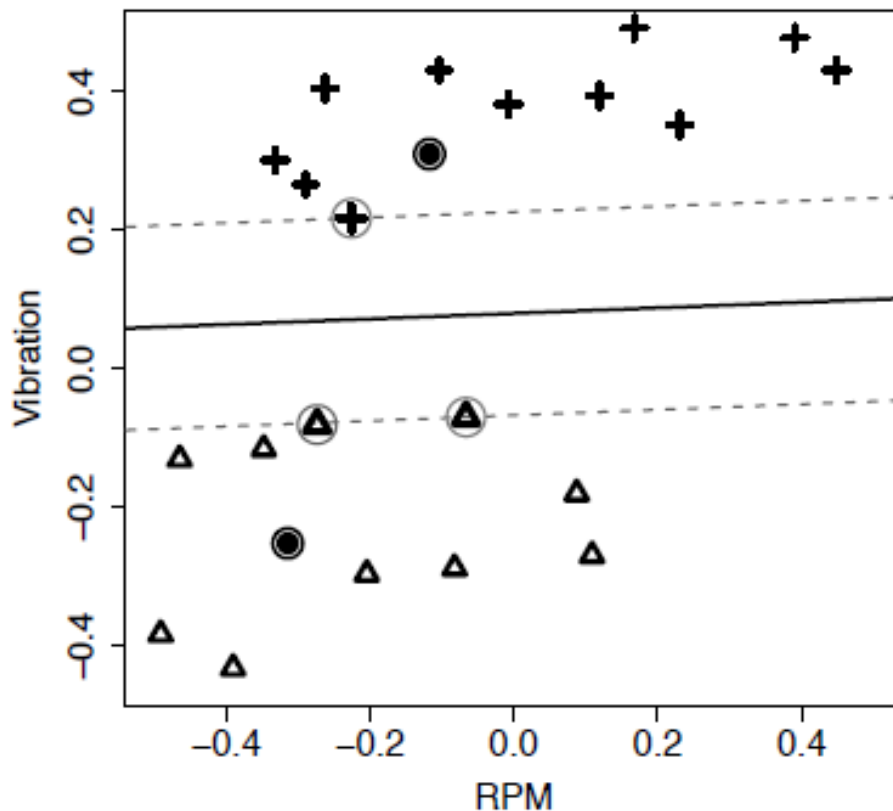
Prediction by Support Vector Machine Model

$$\mathbb{M}_{\alpha, w_0}(\mathbf{q}) = \sum_{i=1}^S (t_i \times \alpha[i] \times (\mathbf{d}_i \cdot \mathbf{q}) + w_0)$$

- This equation gives the decision value for a query point \mathbf{q} .
 - If $\mathbb{M}_{\alpha, w_0}(\mathbf{q}) > 1$, predict the **positive class** of the query instance, \mathbf{q} .
 - If $\mathbb{M}_{\alpha, w_0}(\mathbf{q}) < -1$, predict the **negative class** of the query instance, \mathbf{q} .

Example: Optimal Support Vector Machine

- The figure below shows the optimal decision boundary and associated support vectors for the generators dataset



- The **support vectors** are
 - $(\langle -0.225, 0.217 \rangle, +1)$,
 - $(\langle -0.066, -0.069 \rangle, -1)$,
 - $(\langle -0.273, -0.080 \rangle, -1)$.
- The value of w_0 is -0.1838,
- The values of the α parameters determined during training are $\langle 22.056, 6.998, 16.058 \rangle$.

Figure: In the generators dataset case, 'good' is the positive level and set to +1, and 'faulty' is the negative level and set to -1.

Example: Prediction using a SVM

$$\mathbb{M}_{\alpha, w_0}(\mathbf{q}) = \sum_{i=1}^S (t_i \times \alpha[i] \times (\mathbf{d}_i \cdot \mathbf{q}) + w_0)$$

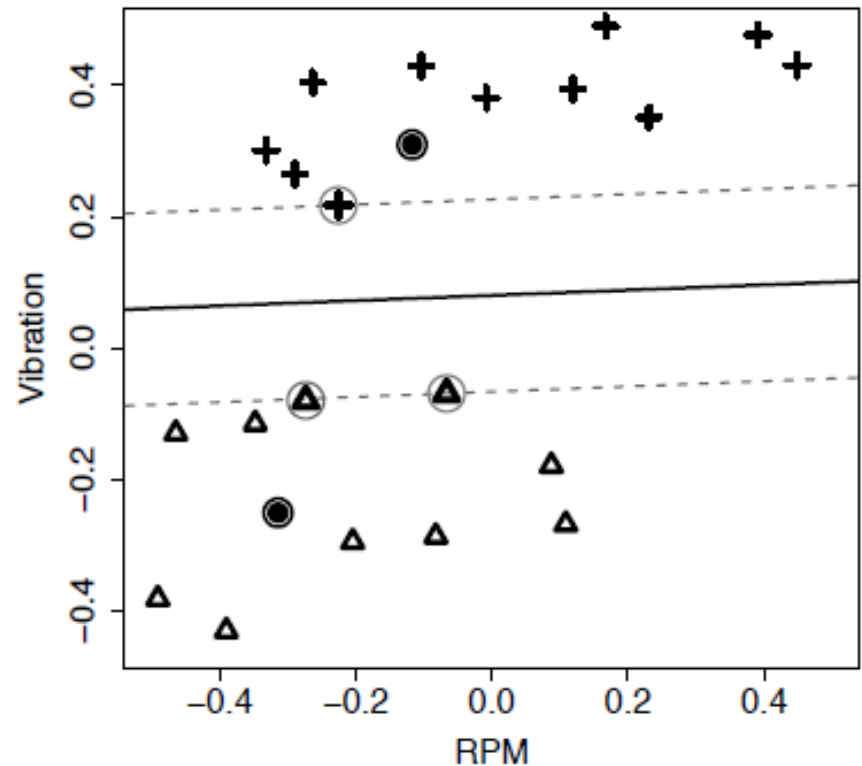
■ **Trained Model :** $\mathbb{M}_{\alpha, -0.1838}(\mathbf{q})$
 $= (1 \times 22.056 \times ((-0.225 \times \mathbf{q}_{RPM}) + (0.217 \times \mathbf{q}_{Vibration}))) + (-0.1838))$

■ Two new query instances

$\langle \text{RPM}, \text{VIBRATION} \rangle$ are:

$$\mathbf{q}_1 = \langle -0.314, -0.251 \rangle$$

$$\mathbf{q}_2 = \langle -0.117, 0.31 \rangle.$$



Example (cont.)

- For the first query instance, $\mathbf{q}_1 = \langle -0.314, -0.251 \rangle$, the output of the support vector machine model is:

$$\begin{aligned} M_{\alpha, w_0}(\mathbf{q}_1) &= (1 \times 23.056 \times ((-0.225 \times -0.314) + (0.217 \times -0.251)) - 0.1838) \\ &\quad + (-1 \times 6.998 \times ((-0.066 \times -0.314) + (-0.069 \times -0.251)) - 0.1838) \\ &\quad + (-1 \times 16.058 \times ((-0.273 \times -0.314) + (-0.080 \times -0.251)) - 0.1838) \\ &= -2.145 \end{aligned}$$

- The model output is less than -1, so this query is predicted to be a 'faulty' generator.
- For the second query instance, the model output is 1.592, so this instance is predicted to be a 'good' generator.

Basis Functions and Support Vector Machine

- **Basis functions** can be used with support vector machines to handle training data that is **not linearly separable**.
- To use basis functions, we update the **constraint** for training a support vector machine like

$$t_i \times (w_0 + \mathbf{w} \cdot \boldsymbol{\phi}(\mathbf{d})) \geq 1 \text{ for all } i$$

where $\boldsymbol{\phi}$ is a set of basis functions applied to the descriptive features \mathbf{d} , and \mathbf{w} is a set of weights containing one weight for each member of $\boldsymbol{\phi}$.

- Typically, the number of basis functions in $\boldsymbol{\phi}$ is larger than the number of descriptive features, so the application of the basis functions moves the data into a higher-dimensional space.
- Here, we expect that a linear separating hyperplane will exist in this higher-dimensional space even though it does not in the original feature space.

SVM Model with Basis Functions

- The support vector machine model with basis functions becomes

$$\mathbb{M}_{\alpha, \phi, w_0}(\mathbf{q}) = \sum_{i=1}^S (t_i \times \alpha[i] \times (\phi(\mathbf{d}_i) \cdot \phi(\mathbf{q})) + w_0)$$

- This equation requires a dot product calculation between the result of applying the basis functions to the query instance and to each of the support vectors which is repeated multiple times during the training process.
- A dot product of two high-dimensional vectors is a **computationally expensive** operation. So we need efficient methods for training the support vector machine.

Kernel Trick

- We can use a clever trick – the **kernel trick** - to avoid the dot product computation.
- The same result obtained by calculating the dot product of the descriptive features of a support vector and a query instance after having applied the basis functions can be obtained by applying a much less costly **kernel function**, *kernel*, to the original descriptive feature values of the support vector and the query.

Prediction Model with Kernel

- The support vector machine model with *kernel* becomes
$$\mathbb{M}_{\alpha, \text{kernel}, w_0}(\mathbf{q}) = \sum_{i=1}^S (t_i \times \alpha[i] \times \text{kernel}(\mathbf{d}_i \cdot \mathbf{q}) + w_0)$$
- A wide range of standard kernel functions can be used with support vector machines including:

Linear kernel $\text{kernel}(\mathbf{d}, \mathbf{q}) = \mathbf{d} \cdot \mathbf{q} + c$
where c is an optional constant

Polynomial kernel $\text{kernel}(\mathbf{d}, \mathbf{q}) = (\mathbf{d} \cdot \mathbf{q} + 1)^p$
where p is the degree of a polynomial function

Gaussian radial basis kernel $\text{kernel}(\mathbf{d}, \mathbf{q}) = \exp(-\gamma \|\mathbf{d} - \mathbf{q}\|^2)$
where γ is a manually chosen tuning parameter

Summary

- **Support Vector Machines (SVM)** models are trained in a slightly different way than regression models, but the concepts underpinning both approaches are similar.
- The main advantages of SVM models are that they are robust to overfitting and perform well for very high-dimensional problems.
- SVM models are just one of a whole range of error-based approaches that are active areas for machine learning research, and new approaches are constantly being developed.