UNIT II

Unit - 2 - II

SPARSE KERNEL MACHINES

Introduction :

Limitation of algorithms based on non linear kernels is that the kernel functions $k(x_n, x_m)$ must be evaluated for all possible pairs x_n and x_m of training points

Here we will take up kernel based algorithms that have sparse solutions so that prediction for new inputs depend only on the kernel functions evaluated at a subset of the training data points.

Mapping the data onto higher dimensions is helpful in creating hyperplane based classifiers. In such classifiers we can reduce the data sets to Support vectors as feature vectors

Example: $W_1 = 1, -1, 0.5, -0.5$, $W_2 = -4, 4, -3, 3, -2, 2$

Maximal Margin Classifiers :

Important property of SVM :

Determination of Model parameters correspond to a Convex optimisation problem and hence any local solution is also a global optimum. (Connect this to above example).

Note : SVM is a decision machine and hence does not provide posterior probabilities (Alternate Sparse Kernel Machine is Relevance Vector Machine (RVM)

(1) What is SVM?

SVM is a supervised Learning Algorithm which can be used for classification and Regression problems as Suppor Vector Classification (SVC) and Support Vector Regression (SVR). Generally it is used for small data sets.

Ideology behind SVM :

SVM is based on the idea of finding a hyperplane that separates the features into different domains.

Terminologies used in SVM.

Support vector points : These are points closest to the hyperplane

Margins : The distance of the vectors from the hyperplane are called margins

Margin maximising hyperplanes :

The hyperplane is called as Margin maximising hyperplane.

Hyper plane : (Decision Surface)

The hyperplane is a function which is used to differentiate between features. In 2D hyperplane is a line & in 3D is a plane and the function which classifies in higher dimension is called a hyper plane.

Hard Margin :

The equations π : $b + w^T x = 0$, π^+ : $b + w^T x = 1$ and π^- : $b + w^T x = -1$ are very strict constraints to correctly classify each and every data point. Hence they are called Hard Margins.

(2) Derive the Maximum Margin classifier expression for binary classification of points in SVM

(OR)

Derive an expression for a Hyperplane that classifies the features of Data for SVM classifier.

Let there be m dimensions

The equation of the hyperplane in M dimension is given by

$$y = w_0 + w_1 x_1 + w_2 x_2 + \dots + w_m x_m$$

$$= w_0 + \sum_{i=1}^{m} w_i x_i$$

$$= w_0 + w^T x \quad \text{where } w = (w_1, w_2 \dots w_m)^T.$$

$$x = x_1, x_2 \dots x_m$$

$$= b + w^T x \quad (1) \quad \text{where } b \text{ is the bias term}$$

If $\phi(x)$ denotes the feature space transformation, then SVM for 2-class-classification problem using linear models takes the form

$$y(x) = w^T \phi(x) + b \quad (2)$$

Let there be N training data $x_1, x_2, \dots x_N$

Let the target values be $t_1, t_2, \dots t_N$ where $t_n \in \{-1, 1\}$.

Assuming the data is linearly separable $\exists$ w and b $\ni$:

$$\begin{cases} y(x_n) > 0 & \text{for points having } t_n = 1 \\ y(x_n) < 0 & \text{" " " } t_n = -1 \end{cases} \quad (3)$$

SVM approaches thro' a concept of the MARIGIN (defined to be smallest distance between decision boundary and any of the samples) and this marigin needs to be maximised.

The $\perp^n$ distance of a pt x from the hyperplane defined by $y(x) = 0$ as in (2) is given by $\frac{|y(x)|}{\|w\|}$

Clearly $t_n y(x_n) > 0 \quad \forall n$ from the view of (3)

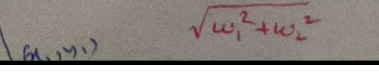
So, Distance from x_n to the decision surface is

$$\frac{t_n y(x_n)}{\|w\|} = \frac{t_n [w^T \phi(x_n) + b]}{\|w\|} \quad (4)$$

$w_1 x + w_2 y + w_3 = 0$

$\perp^n$ distance from (x_1, y_1) :

$$\frac{w_1 x_1 + w_2 x_2 + w_3}{\sqrt{w_1^2 + w_2^2}}$$



The Margin is given by $\perp$ distance to closest x_n from data set.

Claim: Optimise w and b (parameters) to maximise this distance

Hence Maximum Margin is given by

$$\arg\max_{w,b} \left\{ \frac{1}{\|w\|} \min_{x_n} \left[t_n (w^T \phi(x_n) + b) \right] \right\} \quad (5)$$

as w is independent of n, $1/\|w\|$ is taken out of "Min".

If we re-scale $w \to kw$ and $b \to kb$ then the distance of DS from x_n $\left[t_n y(x_n)/\|w\| \right]$ remains unchanged

∴ we can set

$$t_n (w^T \phi(x_n) + b) = 1 \quad (6)$$

In this case all data points satisfy

$$t_n (w^T \phi(x_n) + b) \geq 1 \quad \forall n = 1 \text{ to } N \quad (7)$$

(7) is called Canonical Representation of Hyperplane.

$\Longmapsto$ Maximise $\|w\|^{-1}$ $\longmapsto$ Minimise $\|w\|^2$ and we have an optimization problem.

$$\arg\min_{w,b} \frac{1}{2} \|w\|^2 \quad (8)$$

Subjected to (7), a quadratic programming problem

—

Note:

(1) (8) is a quadratic programming problem. in which we need to minimise a quadratic function subject to a set of linear inequality constraints.

Note (2)

In order to solve constrained optimization problem, we use Lagrangian multipliers $a_n \geq 0$ with one multiplier for each constraints in (7) :

$$L[w, b, a] = \frac{1}{2}\|w\|^2 - \sum_{n=1}^{N} a_n \{[t_n(w^T \phi(x_n) + b)] - 1\} \quad (9)$$

where $a = (a_1, a_2, \ldots a_N)^T$

$$\frac{\partial L}{\partial w} = 0 \implies w = \sum_{n=1}^{N} a_n t_n \phi(x_n) \quad (10)$$

$$\frac{\partial w}{\partial b} = 0 \implies 0 = \sum_{n=1}^{N} a_n t_n \quad (11)$$

Note (3). Eliminating w and b from $L[w, b, a]$ using these conditions it gives dual representation of Max Margin problem.

$$\tilde{L}(a) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{n=1}^{N} a_n a_m t_n t_m k(x_n, x_m) \quad (12)$$

Subjected to constraints

$$a_n \geq 0 \quad \forall n = 1 \text{ to } N \quad (13)$$

$$\sum_{n=1}^{N} a_n t_n = 0 \quad (14)$$

Where $k(x, x') = [\phi(x)]^T [\phi(x')]$

Note (4) Solution of quadratic programming problem in M variables has $O(M^3)$ complexity.

Note (5) : Formulate the Soft Margin Technique of SVM !

The given data may not be linearly separable.

⇒ we need an updation so that one function may skip few outliers. and classify it.

Hence Introduce a slack variable ξ_i so that

$$y_i (w^T x_i + b) \geq 1 - \xi_i \quad \forall i \qquad (15)$$

If $\begin{cases} \xi_i = 0, & \text{the points can be considered correctly classified} \\ \xi_i > 0 & \text{incorrectly classified} \end{cases}$

Then ξ_i can be considered as error associated with x_i

$$\text{Average error} = \frac{1}{n} \sum_{i=1}^{n} \xi_i \qquad (16)$$

Thus Mathematical formulation of Soft Margin SVM becomes

$$\underset{w,\, b}{\text{Minimise}} \quad \frac{1}{2} \|w\|^2 + \sum_{i=1}^{n} \xi_i \qquad (17)$$

Subject to $y_i [w^T x_i + b] \geq 1 - \xi_i \quad \forall i = 1, 2, \ldots n$

Formulation : (Soft Margin Technique)

To find the vector w and the scalar b such that the hyperplane represented by w and b maximises the Margin distance and minimises the loss Term subjected to the condition that all points are correctly classified.

NOTE (8) Interpret Loss function of SVM?

Loss function Interpretation of SVM

We have $z_i = y_i (w^T x_i + b) \longrightarrow (1)$

$z_i \geq 1 \qquad (2)$

If $z_i \geq 1$ then Loss is '0'

$z_i \leq 1$ then Loss increases

Hence

Hinge loss = $\max(0, 1 - z_i)$

NOTE (9) What is dual form of SVM.?

Dual form of SVM :

Consider a data set which is not at all linearly separable.

In logistic regression separation of data is by projecting it on to higher dimension by ADDING RELEVENT FEATURES

But in SVM the projection is done on a powerful way which is called 'PRIMAL FORM'

An alternative method is the Dual form of SVM that uses Lagrange's Multiplier Method :

$$\text{Maximise}_{\alpha} \left[\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i^T x_j) \right] \longrightarrow (18)$$

Subject to

$$\left.\begin{aligned} \alpha_i &\geq 0 \quad \forall i = 1 \text{ to } n \\ \sum_{i=1}^{n} \alpha_i y_i &= 0 \end{aligned}\right\} \longrightarrow (19)$$

Remark : If $\alpha_i > 0$, then x_i is the support vector

(1) $\alpha_i = 0$ " " is NOT support vector

(2) To solve actual problem we do not require actual data. Instead dot product of all pairs sufficient

(3) Calculation of 'b' also requires dot product only

(4) Major advantage of Dual SVM is that it depends only on 'α'.

Note (10) What is Kernel Trick?

The Kernel is a way of computing the dot product of two vectors x and y in some (high dimensional) feature space, [Sometimes it is called 'Generalised dot product'.

Note (11) Write a short note on various types of Kernels used in SVM?

S. NO	Type of Kernel	Expression	Remarks.
1.	Linear Kernel		
2.	Polynomial	$K(x_1, x_2) = (a + x_1^T x_2)^b$	b – deg of Kernel a = const term.
3.	Radial Basis function. (RBF) / Gaussian Kernel.	$K(x_1, x_2) = \exp(-\gamma \lVert x_1 - x_2 \rVert^2)$	$\lVert x_1 - x_2 \rVert \rightarrow$ Euclidean distance bet. x_1 & x_2

Parameters :

(1) C : Inverse of strength of regularisation

: Behaviour : As C increases the model gets overfit

As C decreases the model gets underfit

(2) γ : Used only for RBF Kernel

: Behaviour : As γ increases the model gets overfit

As γ decreases the model gets underfit

Pros and Cons of SVM:

Pros:

(1) Really effective in higher dimension

(2) Effective when the number of features more than training examples

(3) Best Algorithm when classes are separable

(4) Hyperplane is affected by support vectors and thus outliers have less impact

(5) Suited for extreme case binary classification

Cons:

(1) Computationally expensive for larger data sets

(2) Does not perform well in case of overlapped classes

(3) Selecting the appropriate kernel function is tricky

(4) Selection of appropriate hyperparameters requiring sufficient generalisation performance

Preparing data for SVM:

Numerical Conversion: As inputs are numerical instead of categorical, Most commonly used are 'one hot encoding' Label-encoding etc.

Binary Conversion:

One Verses the rest method / One verses one method.

Overlapping class Distributions :

(1) What is the work of regularisation parameter C in SVM? (in box constraints)

It controls the trade-off between minimising training errors and controlling model complexity.

(2) State the -Khun- Tucker conditions of the Lagrangian formulation of SVM and derive the box constraints for overlapping class Distributions?

Based on the assumption that training data are linearly separable in feature space $\phi(x)$, the resulting SVM gives exact separation of the training data in original input space x, although corresponding decision boundary is non-linear. But in practice, the class-conditional distributions may overlap. Hence our goal is to maximise the margin while softly penalising points that lie on the wrong side of the margin boundary.

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

Where C controls the trade off between the slack variable penalty and the Margin.

The Corresponding Lagrangian formulation is given by

$$L[w, b, a] = \frac{1}{2}\|w\|^2 + C\sum_{n=1}^{N} \xi_n - \sum_{n=1}^{N} a_n\{t_n y(x_n) - 1 + \xi_n\} - \sum_{n=1}^{N} \mu_n \xi_n \longrightarrow (2)$$

where $a_n, \mu_n \geq 0$ are Lagrangian multipliers.

The KKT conditions are given by

$$a_n \geq 0 \quad (3)$$

$$t_n y(x_n) - 1 + \xi_n \geq 0 \quad (4)$$

$$a_n (t_n y(x_n) - 1 + \xi_n) = 0 \quad (5)$$

$$\mu_n \geq 0 \quad (6)$$

$$\xi_n \geq 0 \quad (7)$$

$$\mu_n \xi_n = 0 \quad (8) \quad \forall n = 1 \text{ to } N$$

For optimising,

$$\frac{\partial L}{\partial w} = 0 \implies w = \sum_{n=1}^{N} a_n t_n \phi(x_n) \quad (9)$$

$$\frac{\partial L}{\partial b} = 0 \implies \sum_{n=1}^{N} a_n t_n = 0 \quad (10)$$

$$\frac{\partial L}{\partial \xi_n} = 0 \implies a_n = C - \mu_n \quad (11)$$

Eliminating w, b and $\{\xi_n\}$ from the Lagrangian,

$$\tilde{L}(a) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m K(x_n, x_m) \quad (12)$$

Now Minimise (12) wrt dual variables $\{a_n\}$, subject to

$$0 \leq a_n \leq C \quad (13)$$

$$\sum_{n=1}^{N} a_n t_n = 0 \quad (14) \quad \text{where } n = 1, 2, \ldots N$$

(13) are known as 'BOX CONSTRAINTS'.

(3) Express the 2nd order polynomial kernel in terms of its components namely vector function and feature space.

Soln:

$$K(x, z) = (1 + x^T z)^2$$

$$= (1 + x_1 z_1 + x_2 z_2)^2$$

$$= 1 + 2x_1 z_1 + 2x_2 z_2 + x_1^2 z_1^2 + 2x_1 z_1 x_2 z_2 + x_2^2 z_2^2$$

$$= (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, \sqrt{2}x_1 x_2, x_2^2)(1, \sqrt{2}z_1, \sqrt{2}z_2, z_1^2, \sqrt{2}z_1 z_2, z_2^2)^T$$

$$= \phi(x)^T \phi(z)$$

(4) How do you relate SVM with logistic regression?

They can be related by their error functions given by

$$E_{SV}(y_n t_n) = (1 - y_n t_n)_+$$ It can be viewed as an approximation to the misclassification error in SVM.

In logistic regression, with $p[t = \frac{1}{y}] = \sigma(y)$ and $p[t|y] = \sigma(yt)$. For this error function is given by

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

(5) Write a short note on Multiclass SVM

Commonly used approach focusses constructing K-Separate SVMs in which Kth model $y_k(x)$ is trained using the C_kth class data as positive examples and the data from remaining K-1 classes as negative examples. This is one-verses the rest approach. Two disadvantages: 1. Individual classifiers can lead to inconsistent results in which an input is assigned to multiple classes simultaneously. 2. Due to imbalanced training data the symmetry of original problem could be lost.

Another approach (one verses one) is to train $K(K-1)/2$ different 2 classes SVMs on all possible pairs of classes, and then to classify test points according to which class has the highest number of votes In this approach more amount of significant computation is required

⑥ Explain how SVM can be extended to Regression problems? (OR) Write down the steps involved in converting the general SVM to be applied for regression problem

By preserving the property of Sparseness one can extend SVM to regression problems :-

Step 1 : Introduction of ϵ - intensive error function having a linear cost associated with errors outside the insensitive region as

$$E_\epsilon [y(x) - t] = \begin{cases} 0 & |y(x) - t| < \epsilon \\ |y(x) - t| - \epsilon & \text{otherwise} \end{cases} \quad (1)$$

(Refer Fig 1)

for minimising a regularised error function.

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

Step (2) Re expressing the optimization function as

$$\text{Min } E_r(w, \epsilon) = \frac{1}{2} \|w\|^2 + C \sum_{n=1}^{N} E_\epsilon (y(x_n) - t_n) \quad (3)$$

subjected to $t_n \leq y(x_n) + \epsilon + \xi_n$ (4)

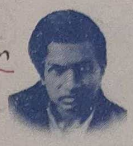
$t_n \geq y(x_n) - \epsilon - \hat{\xi}_n$ (5)

where ξ_n are slack variables, $(\hat{\xi}_n)$, $t_n > y(x_n) + \epsilon$ and $t_n < y(x_n) - \epsilon$ converted to (4) & (5) Refer Fig 2.

Step (3) Modification of error function in view of (3), (4) and (5) :

$$E_r(w, \xi_n, \hat{\xi}_n) = \frac{1}{2} \|w\|^2 + C \sum_{n=1}^{N} (\xi_n + \hat{\xi}_n) \quad (6)$$

(6) to be minimised subject to $\xi_n, \hat{\xi}_n \geq 0$ & (4) and (5)



Step (4) : Construction and Optimization of Lagrangian function with Lagrangian multipliers $a_n, \hat{a}_n, \mu_n, \hat{\mu}_n \geq 0$:

$$L[w, b, \xi_n, \hat{\xi}_n]$$

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

$$- \sum_{n=1}^{N} a_n (\epsilon + \xi_n + y_n - t_n) - \sum_{n=1}^{N} \hat{a}_n (\epsilon + \hat{\xi}_n + t_n - y_n)$$

⟶ (7)

Step (5) Simplication of constraints and setting the derivatives to 0

Substituting for $y(x)$, $w^T\phi(x) + b$, (*)

$$\frac{\partial L}{\partial w} = 0 \iff w = \sum_{n=1}^{N}(a_n - \hat{a}_n)\phi(x_n) \quad (8)$$

$$\frac{\partial L}{\partial b} = 0 \iff \sum_{n=1}^{N}(a_n - \hat{a}_n) = 0 \quad (9)$$

$$\frac{\partial L}{\partial \xi_n} = 0 \iff a_n + \mu_n = C \quad (10)$$

$$\frac{\partial L}{\partial \hat{\xi}_n} = 0 \iff \hat{a}_n + \hat{\mu}_n = C \quad (11)$$

Step (6) Construction of Dual : (∵ Elimination of corresponding variables from Lagrangian)

$$\text{Maximise } \tilde{L}(a, \hat{a}) = -\frac{1}{2}\sum_{n=1}^{N}\sum_{n=1}^{N}(a_n - \hat{a}_n)(a_m - \hat{a}_m)\,k(x_n, x_m)$$

$$- \epsilon\sum_{n=1}^{N}(a_n + \hat{a}_n) + \sum_{n=1}^{N}(a_n - \hat{a}_n)t_n \quad (12)$$

where $k(x, x') = \phi(x)^T\phi(x')$

This is a constrained maximization problem. $a_n, \hat{a}_n$, $\mu_n, \hat{\mu}_n \geq 0$, (10) and (11) yields

Step (7) Construction of box constraints:

$$0 \leq a_n \leq C \quad (13)$$

$$0 \leq \hat{a}_n \leq C \quad (14)$$

Step (8) Simplification to programmable form of Dual:

Substitution (8) in to (*)

$$y(x) = \sum_{n=1}^{N} (a_n - \hat{a}_n) k(x, x_n) + b \quad (15)$$

Corresponding Kuhn - Tucker conditions yield

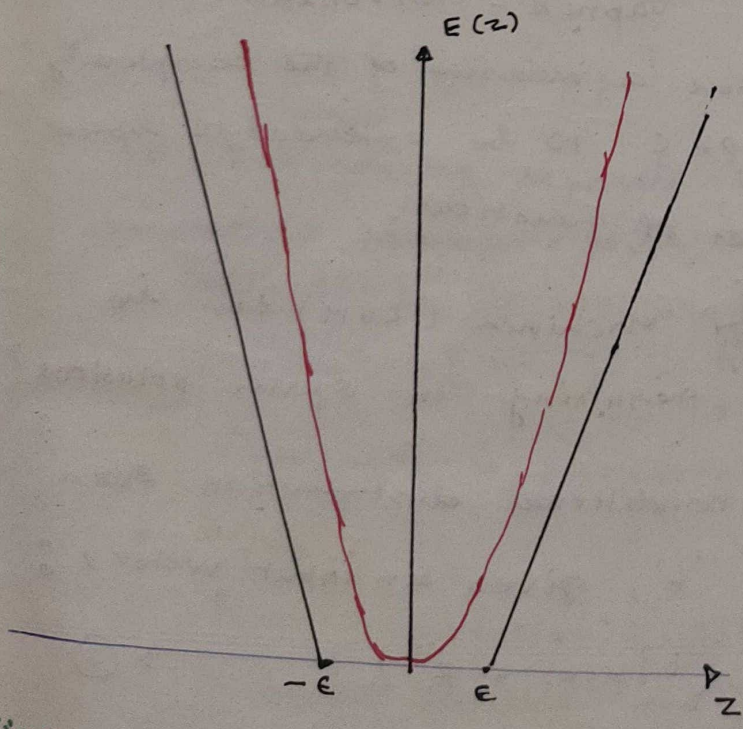
$$a_n [\epsilon + \xi_n + y_n - t_n] = 0 \quad (16)$$

$$\hat{a}_n [\epsilon + \hat{\xi}_n - y_n + t_n] = 0 \quad (17)$$

$$(C - a_n) \xi_n = 0 \quad (18)$$

$$(C - \hat{a}_n) \hat{\xi}_n = 0 \quad (19)$$

where $$b = t_n - \epsilon - \sum_{m=1}^{N} (a_m - \hat{a}_m) k(x_n, x_m) \quad (20)$$



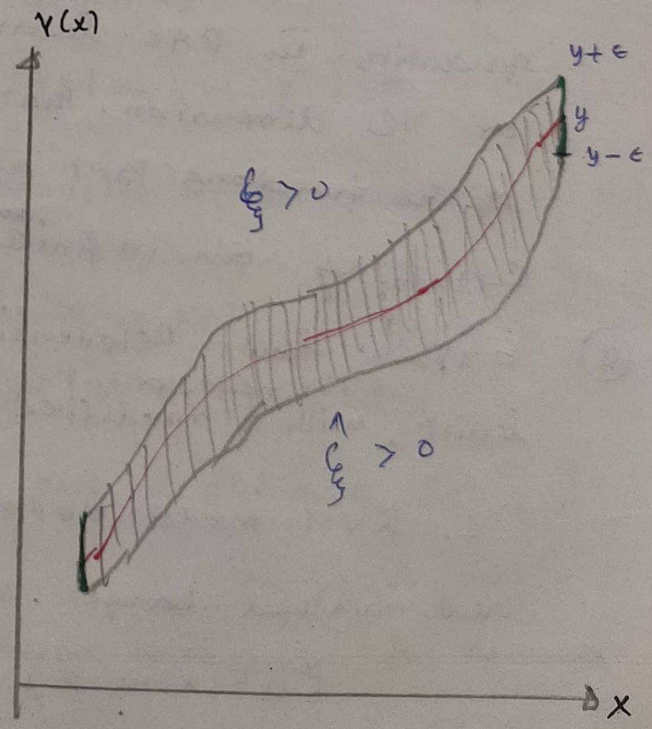


Fig 1: Plot of ε-insensitive error fn in which error increases linearly with distance beyond the insensitive region.

(7) Write a short note on 'computational Learning Theory'? (CLT)

CLT is a framework that provides full motivation to SVM originated from "Probably Approximately Correct" (PAC) whose goal is to understand how large a data set needs to be in order to give 'Good Generalisation'. It provides bounds for computational cost of learning.

In PAC learning, we say a function $f(x, D)$ drawn from the space F of such functions on the basis of the training set D has good generalization if its expected error rate is below some threshold ϵ, so that

$$E_{x,t}\left[I\left(f(x; D) \neq t \right) \right] < \epsilon$$

where $I(\cdot)$ is an idicator function.

PAC learning aims to provide bounds for the minimum size of N of data set needed to meet this criterian. A key quantity in PAC learning is Vapnik-Chervonenkis dimension or VC dimension. that provides a measure of the complexity of the functions that allows PAC to be extended to spaces containing an infinite number of functions.

(8) Explain how Relevence Vector Machines (RVM) can be built (for regression), with a modified prior resulting in sparse solutions?

1. RVM model defines a conditional distribution for a real-valued target variable 't', given an input vector x:

$$P[t | x, w, \beta] = N\left(t | y(x), \beta^{-1}\right) \longrightarrow (1)$$

where $\beta = 1/\sigma^2$ being the noise precision and mean given in the linear form 22:

$$y(x) = \sum_{i=1}^{N} w_i \phi_i(x) = w^T \phi(x) \longmapsto (2)$$

with non-linear basis functions $\phi_i(x)$ including a constant term as bias of (2).

2. General expression takes the SVM-like form

$$y(x) = \sum_{n=1}^{N} w_n k(x, x_n) + b \longmapsto (3)$$ where b - bias

Here number of parameters is $M = N + 1$

Unlike SVM (a) There is no restriction in positive definite kernel

(b) The basis functions are not tied either in number or location to the training data points

(Distinguish RVM for regression from SVM for regression ↑)

Step ① Learning choice of α and β

3. * Let there be N observations of input vector x

* The data matrix X have nth row as x_n^T, $n = 1$ to N.

* Target values be $t = (t_1, t_2, \ldots t_n)^T$

* Then Likelihood function is given by

$$p[t|x, w, \beta] = \prod_{n=1}^{N} p(t_n | x_n, w, \beta^{-1}) \longmapsto (4)$$

* Introducing separate hyperparameter α_i for each of the weight parameters w_i (instead of single shared hyperparameter as in SVM), the weight prior is given by

$$p[w|\alpha] = \prod_{i=1}^{M} N(w_i | 0, \alpha_i^{-1}) \longmapsto (5)$$

where α_i is precision of w_i and $\alpha = (\alpha_1, \alpha_2 \ldots \alpha_M)^T$

* Using ⟶ (6) for linear regression models, Posterior distribution for the weights is again Gaussian and is given by

$$p[w|t, X, \alpha, \beta] = N(w|m, \Sigma) \longmapsto (7)$$

(4) Step ② Evaluating mean and covariance of posterior:

Where mean and covariance are given by

$$m = \beta \Sigma \Phi^T t \quad (8)$$

$$\Sigma = (A + \beta \Phi^T \Phi)^{-1} \quad (9)$$

Where Φ is $N \times M$ design matrix with $\Phi_{ni} = \varphi_i(x_n)$ and $A = \text{diag}(\alpha_i)$ and K in (3) is the symmetric $(N+1) \times (N+1)$ Kernel matrix with elements from $k(x_n, x_m)$

The values of α, and β in (7) are determined using type-2 maximum likelihood, (known as EVIDENCE Approximation)

[(Qn: What is Evidence Approximation in RVM)

Ans: It is type-2 maximum likelihood that maximises the Marginal likelihood function obtained by integrating weight parameters)]

Thus $$p[t|x, \alpha, \beta] = \int p(t|x, w, \beta)\, p(w|\alpha)\, dw \quad \longrightarrow (10)$$

As it is convolution of two Gaussians, It can be evaluated through log likelihood in the form

$$\log p[t|x, \alpha, \beta] = \log N[t|0, C]$$

$$= -\frac{1}{2}\left[N \log 2\pi + \log|C| + t^T C^{-1} t\right]$$

where $t = (t_1, t_2, \ldots t_N)^T$ and $\longrightarrow (11)$

$$C = \beta^{-1} I + \Phi A^{-1} \Phi^T \quad \longrightarrow (12)$$

For maximising (11), Setting derivatives of Marginal likelihood to zero yields

5. Step (3) : Re-estimating Hyper parameters :

$$\alpha_i^{new} = \frac{\gamma_i}{m_i^2} \quad \longrightarrow (13)$$

$$(\beta^{new})^{-1} = \frac{\|t - \Phi m\|^2}{N - \sum_i \gamma_i} \quad \longrightarrow (14)$$

Where m_i - i^{th} component of the posterior mean m as in (8)

γ_i = a measure corresponding to w_i determined by data defined as $\gamma_i = 1 - \alpha_i \Sigma_{ii}$ where Σ_{ii} is the i^{th} diagonal component of posterior covariance given by (9) ⟶ (15)

(6) Step (4) Re estimating mean and covariance : Until suitable convergence criterian is satisfied.

(9) What do you mean 'Relevent Vectors' in RVM?

In 'Relevent Vector Machine', in the general expression $y(x) = \sum_{n=1}^{N} w_n k(x, x_n) + b$, the inputs x_n corresponding to remaining non zero weights are called 'Relevent Vectors'

They are identified through the mechanism of automatic relevent determination.

(10) What is the alternative procedure for improving training speed of RVM?

Ans : Sparsity Mechanism.

(11) Explain the Mathematical Analysis of Mechanism of Sparsity in the context of 'Relevent Vector Machine'

1. Origin of Sparsity in Bayesian Linear Models :

Consider a data set comprising $N = 2$ observations t_1, t_2 together with a model having single basis function $\phi(x)$, with hyper parameter α_i with isotropic noise having precision β.

From marginal likelihood of (11) and (12) of previous derivation of RVM for regression, we can write

$\left(P(t|a,\beta) = N[t|0,C] \right)$ in which,

Covariance matrix takes the form

$$C = \frac{1}{\beta} I + \frac{1}{a} \Phi \Phi^T \longrightarrow (1)$$

where Φ denotes the N-dimensional vector $(\phi(x_1), \phi(x_2))^T$ and similarly $t = (t_1, t_2)^T$

This is a zero-mean Gaussian process over t with covariance C. Given a particular observation for t, we need to find a^* and β^* by maximising marginal likelihood on a particular a_i and then determine its stationary points explicitly. :

We have

$$\log p[t|X,a,\beta] = -\frac{1}{2}\left[N \log 2\pi + \log|C| + t^T C^{-1} t \right] \quad (1)$$

from (12) is previous bw

$$C = \beta^{-1} I + \Phi A^{-1} \Phi^T \longmapsto (2) \quad \text{from (12)}$$

$$= \beta^{-1} I + \sum_{j \neq i} a_j^{-1} \Phi_j \Phi_j^T + a_i^{-1} \Psi_i \Psi_i^T$$

$$= C_{-i} + a_i^{-1} \Psi_i \Psi_i^T \longrightarrow (3)$$

where Ψ_i denotes the ith column of Φ; $[(\phi_i(x_1), \phi_i(x_2) \ldots$ $\ldots \phi_i(x_n))$ and C_{-i} represents the matrix C with the contribution from basis function i removed. The determinant and inverse of C can be written as

$$|C| = |C_{-i}| \left[1 + a_i^{-1} \Psi_i^T C_{-i}^{-1} \Psi_i \right] \longrightarrow (4)$$

$$C^{-1} = C_{-i}^{-1} - \frac{C_{-i}^{-1} \Psi_i \Psi_i^T C_{-i}^{-1}}{a_i + \Psi_i^T C_{-i}^{-1} \Psi_i} \longrightarrow (5)$$

With the above equations (1) can be written in the form

$$L(a) = L(a_{-i}) + \lambda(a_i) \longrightarrow (6)$$

Where $L(\alpha_{-i})$ is simply the log marginal likelihood with basis function ψ_i omitted and.

$$\lambda(\alpha_i) = \frac{1}{2}\left[\log \alpha_i - \log(\alpha_i + s_i) + \frac{q_i^2}{\alpha_i + s_i}\right] \quad (7)$$

and contains all of the dependence on α_i and

$$s_i = \psi_i^T C_{-i}^{-1} \psi_i \quad (8)$$

$$q_i = \psi_i^T C_{-i}^{-1} t \quad (9)$$

Here s_i is called sparsity and q_i known as Quality of ψ_i

[* Large value of s_i relative to $q_i \Rightarrow$ Basis function ψ_i is more likely pruned from model.

* 'quality' represents a measure of alignment of ψ_n with error bet. training set and Y_{-i} of predictions with ψ_i excluded]

[Qn: Interpret the two terms 'sparsity' and 'quality' in sparsity mechanism.] ↑

Now stationary points are given by

$$\frac{d\lambda(\alpha_i)}{d\alpha_i} = \frac{\alpha_i^{-1} s_i^2 - (q_i^2 - s_i)}{2(\alpha_i + s_i)^2} = 0 \quad (10)$$

We get $$\alpha_i = \frac{s_i^2}{q_i^2 - s_i}$$

---

(12) Write down Sequential Sparse Bayesian Learning Algorithm:

Sequential Sparse Bayesian Learning Algorithm:

1. For solving regression problem Initialise precision parameter β
2. Initialise one basis function ϕ_1 with hyperparameter α_1, set $\alpha_i = s_i^2 / (q_i^2 - s_i)$. With the remaining hyperparameter α_j for $j \neq i$ initialised to ∞ so that only ψ_i is included in the model.

3. Evaluate Σ and m along with q_i & s_i for all basis fns.

4. Select a candidate basis function φ_i

5. If $q_i^2 > s_i$ and α_i finite then update α_i using $\alpha_i = \frac{s_i^2}{q_i^2 - s_i}$

6 If $q_i^2 > s_i$ and $\alpha_i = \infty$, then add φ_i to the model and evaluate hyperparameter α_i using $\alpha_i = s_i^2 / (q_i^2 - s_i)$

7. If $q_i^2 \leq s_i$ and α_i finite, then remove φ_i & set $\alpha_i = \infty$

8. For solving a regression problem update β.

9. check for Convergence. If not converged Go to 3

(13) Build a 'Relevence Vector Machine' for classification problem?

Consider a 2 class problem with binary target variable $t \in \{0, 1\}$.

With a linear combination of basis functions transformed by a logistic sigmoid function the model becomes

$$y(x, w) = \sigma[w^T \varphi(x)] \quad (1)$$ where $\sigma(\cdot)$ is the logistic sigmoid function given by $\sigma(x) = \frac{1}{1+e^{-x}}$.

Initialise a hyper parameter α. Consider Laplace approximation for this fixed α, Mode of the posterior distribution is obtained by maximising:

$$\log p[w|t, \alpha] = \log\{p(t|w)\, p(w|\alpha)\} - \log p(t|\alpha)$$

$$= \sum_{n=1}^{N} \{t_n \log y_n + (1 - t_n) \log(1 - y_n)\} - \frac{1}{2} w^T A w + C \quad (2)$$

where $A = \text{diag}(\alpha_i)$. Using iterative reweighted least squares, gradient vector and Hessian matrix of the log posterior distribution

(2) $\Rightarrow \nabla \left[\log p(w|t,\alpha)\right] = \Phi^T(t-y) - \alpha w$ $\longmapsto$ (3)

$\nabla\nabla \log p[w|t,\alpha] = -(\Phi^T B \Phi + A)$ $\longmapsto$ (4)

Where B is an $N \times N$ diagonal matrix with elements $b_n = y_n(1-y_n)$ the vector $y = (y_1, \ldots y_N)^T$, and Φ is the design matrix with elements $\phi_{ni} = \phi_i(x_n)$.