

Machine Learning (CS60050)

Instructor: Aritra Hazra

Scribes By:

Akash Kumar Gangwar (20CS60R31)

Soumya Porel (20CS60R36)

Lecture Date: 19th March 2021, Friday

1 RBF centers and Support Vectors

Full name of RBF is "Radial Basis Function". Till now every example of training data plays a part to build hypothesis. what if to a point, a nearby points play more part to shaping or classifying the level of that point rather than the farthest point. So we want to make such a model.

A radial basis function (RBF) is a real-valued function φ whose value depends only on the distance between the input and some fixed point, either the origin, so that $\varphi(\mathbf{x}) = \varphi(\|\mathbf{x}\|)$, or some other fixed point \mathbf{c} , called a center, so that $\varphi(\mathbf{x}) = \varphi(\|\mathbf{x} - \mathbf{c}\|)$. Any function φ that satisfies the property $\varphi(\mathbf{x}) = \varphi(\|\mathbf{x}\|)$ is a radial function. The distance is usually Euclidean distance, although other metrics are sometimes used. They are often used as a collection $\{\varphi_k\}_k$ which forms a basis for some function space of interest, hence the name.

Radial basis is $= e^{(-\gamma \cdot |x - x_n|^2)}$

Linear regression:

$$\begin{bmatrix} e^{(-\gamma \|x_1 - \mu_1\|^2)} & \dots & e^{(-\gamma \|x_1 - \mu_n\|^2)} \\ e^{(-\gamma \|x_2 - \mu_1\|^2)} & \dots & e^{(-\gamma \|x_2 - \mu_n\|^2)} \\ \vdots & \vdots & \vdots \\ e^{(-\gamma \|x_N - \mu_1\|^2)} & \dots & e^{(-\gamma \|x_N - \mu_n\|^2)} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} \approx \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

In above matrix 1st matrix ϕ of size $n \times n$ is the matrix of radial basis, which is invertible. and 2nd matrix w is the matrix of weight. and output matrix y is the matrix of outcomes.

Hence $w = \phi^{-1} y$ (exact interpolation)

here if γ is large then gaussian curve will be much more steeper and if γ is small it will be much smoother.

RBF as a model can be applied for both linear regression as well as linear classification quite easily. and it is a good model because it has more influence of nearer points rather than the farther points.

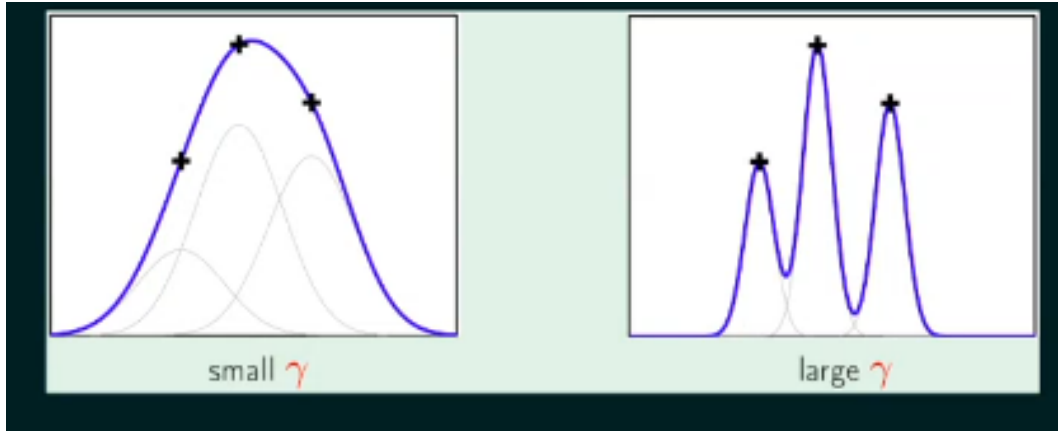


Figure 1: Effect of γ on gaussian curve

1.1 Nearest neighbour

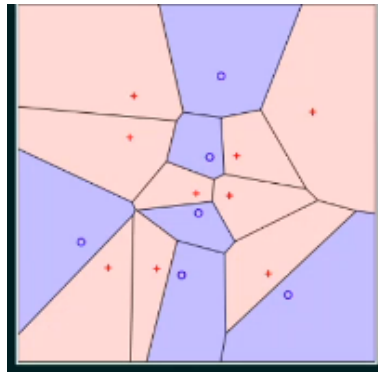


Figure 2: 1-NN

In this we have to choose K - centers where $k \leq N$, suppose $\mu_1, \mu_2, \dots, \mu_k$ and

$$h(x) = \sum_{k=1}^K w_k e^{(-\gamma \|x - \mu_k\|^2)} \quad (1)$$

The question is how we choose μ_k and w_k while learning from the set of data points given to us. Given data points $= x_1, x_2, \dots, x_n$ and classes that i wish to do is $S_1, S_2, S_3 \dots S_k$ My objective function with respect to this is - first of all minimize the distance of all the points with some center k within a particular class, means we have to minimize inter cluster distance that is

$$\min \sum_{k=1}^K \sum_{x_n \in S_k} (\|x_n - \mu_k\|^2) \quad (2)$$

Computationally , it is np-hard.

1.2 Support vectors

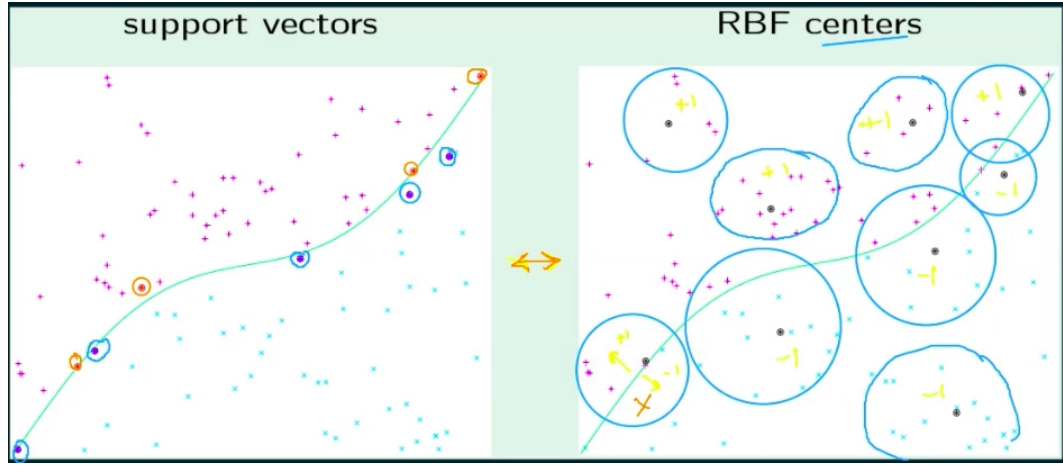


Figure 3: Support vectors and RBF centers

Let us now compare support vectors with the RBF centers. In figure 3, we can see 9 support vectors and 9 RBF centers on the left and right side respectively for the same dataset. The support vectors are trying to find out a separating hyper plane between the 2 labels while the RBF centers are trying to divide the dataset into different clusters in an unsupervised manner. The RBF centers are created without considering the labels (y_n). Let us see how RBF centers can also be used to determine the labels.

If we have K RBF centers (μ_1 to μ_K) and we want to approximate y_n for the n^{th} instance then we need to choose K weights such that equation 3 holds. We can only approximate y_n as some of the clusters will contain mixed data points from both labels. We could put an equal sign in equation 3 if all the clusters were uniform. We can see that if the margin is bigger in getting the support vectors then the clusters will be more disjoint in terms of the labels and i.e. they will be more uniform.

$$\sum_{k=1}^K w_k e^{(-\gamma \|x_n - \mu_k\|^2)} \approx y_n \quad (3)$$

We need equation 3 to hold for all N instances in the dataset. So, we have N equations and K unknowns. The matrix format for this is shown in equation 4.

$$\begin{bmatrix} e^{(-\gamma \|x_1 - \mu_1\|^2)} & \dots & e^{(-\gamma \|x_1 - \mu_K\|^2)} \\ e^{(-\gamma \|x_2 - \mu_1\|^2)} & \dots & e^{(-\gamma \|x_2 - \mu_K\|^2)} \\ \vdots & \vdots & \vdots \\ e^{(-\gamma \|x_N - \mu_1\|^2)} & \dots & e^{(-\gamma \|x_N - \mu_K\|^2)} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_k \end{bmatrix} \approx \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \end{bmatrix} \quad (4)$$

In equation 4, left matrix in the L.H.S is the matrix of radial basis (ϕ), the right matrix in the L.H.S is the weight matrix (w) and the matrix in the R.H.S is the outcome matrix (y).

If $\phi^T \phi$ is invertible then using pseudo inverse we can determine w using the below equation:

$$w = \phi^T \phi^{-1} \phi^T y \quad (5)$$

2 RBF network

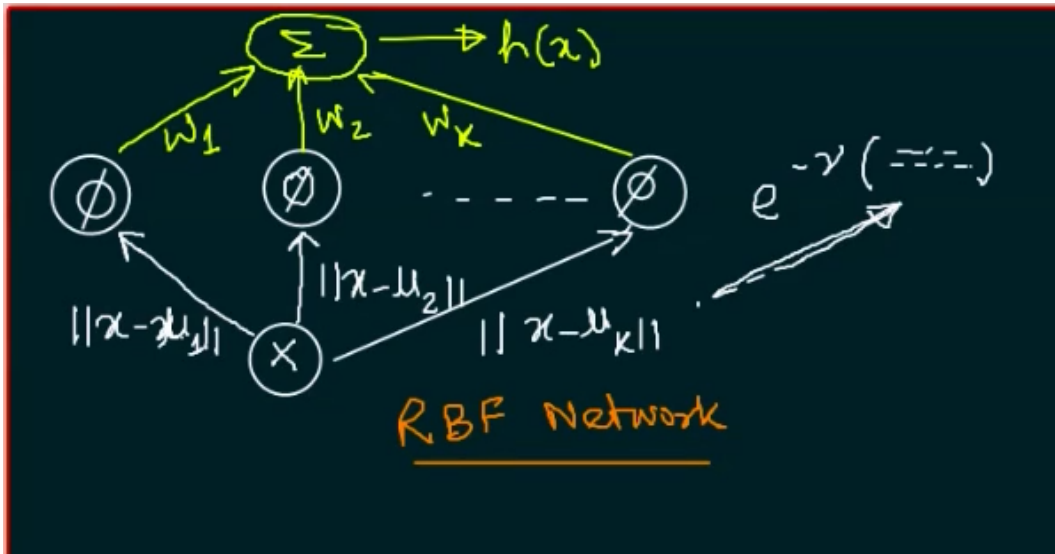


Figure 4: RBF network

Now let us look at RBF as a network. In figure 4, we have X in the first level and in the second level, a basis function ϕ acts over x with a distance of $\|x - \mu_1\|$. Another basis function acts with a distance of $\|x - \mu_2\|$ and so on up-to $\|x - \mu_K\|$. The ϕ function applies the operation $e^{-\gamma(a)}$ for the input a given to it. In the second layer, we have the weights w_1, w_2, \dots, w_K and finally in the third level, the weights are composed using a sum unit and the hypothesis $h(x)$ is produced.

Now let us compare this RBF network with a traditional neural network as shown in figure 5. Both the RBF network and the neural network consists of 2 layers and applies the non-linear functions ϕ and sigmoid respectively over their input. The neural network consists of multiple weights and it applies backpropagation algorithm to learn them. As the basis function used in RBF network is a complicated one, we cannot apply backpropagation algorithm their. Instead we learn the w and μ values iteratively while minimizing the error.

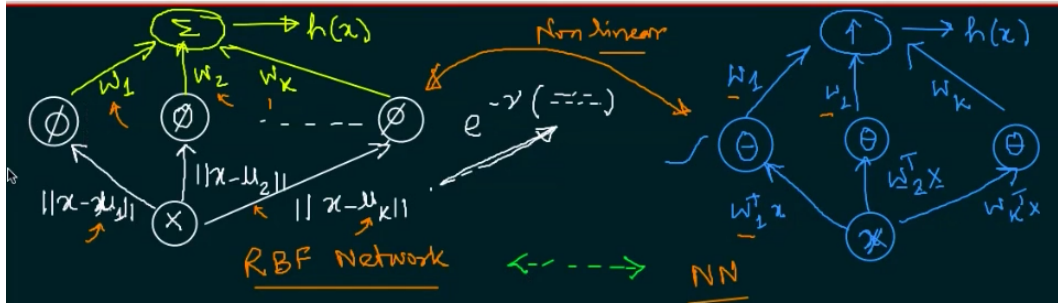


Figure 5: Comparing RBF network with neural network

In the traditional neural network also the backpropagation algorithm tries to minimize the error.

We can also look at support vector machine as a 2 layer network where in the first layer we get the support vectors and in the second layer we get the final margin.

So, we can see that there is a resemblance of the RBF network with the neural network. The traditional neural network have a probabilistic interpretation of the non-linear sigmoid layer while the RBF network has a Gaussian interpretation of its non-linear layer where the nearer data points have more influence on the input.

3 Choosing γ

The γ has an impact on everything. So, we need to choose good value for it. We can do this using EM algorithm (EM stands for Expectation Maximization) as mixture of Gaussians. We can solve this in 2 steps as:

1. Fix γ and solve w_1, w_2, \dots, w_K
2. For w_1, w_2, \dots, w_K , minimize the error with respect to γ

These 2 steps are done iteratively until we converge.

We may choose multiple γ values as well where for some set of points the curve will be more flatter and for some other set of points the curve will be more steeper. So, we will need to fix $\gamma_1, \gamma_2, \dots, \gamma_3$ and perform the above 2 steps until we converge to some local minima. Convergence is guaranteed as we have a fixed set of points.

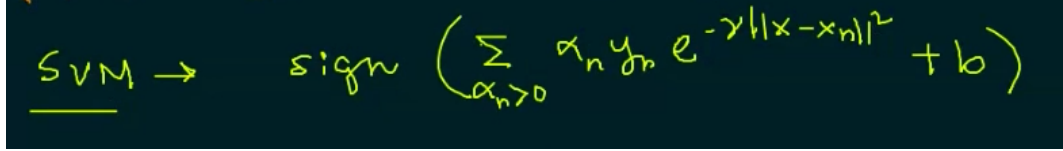
4 RBF as Kernel

RBF can also be applied in terms of kernels as shown in equation 6 and explained in previous lecture:

$$K(x, X') = e^{-\gamma \|X - X'\|^2} \quad (6)$$

RBF was actually invented to be used as a kernel and later people found out that how to get learning models from the Gaussian function.

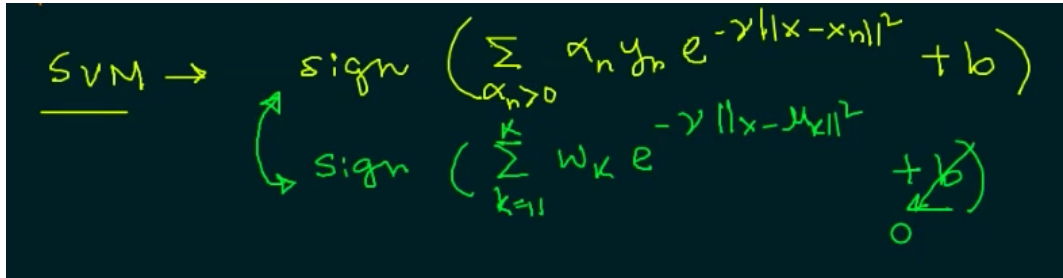
If we use equation 6 in the SVM equation then the SVM equation will boil down to what is shown in figure 6.



The image shows a handwritten equation on a dark background. On the left, 'SVM' is underlined and followed by an arrow pointing to the right. The equation is: $\text{sign} \left(\sum_{\alpha_n > 0} \alpha_n y_n e^{-\gamma \|x - x_n\|^2} + b \right)$. The summation is written with a subscript $\alpha_n > 0$.

Figure 6: RBF kernel in SVM equation

This is quite similar to the what we did in RBF classification. Figure 7 compares the RBF classification equation with the SVM equation. These two methods are solving the same optimization problem but they follow different maximization or minimization criteria and produce different kinds of separating boundary.



The image shows two handwritten equations on a dark background. On the left, 'SVM' is underlined and followed by an arrow pointing to the right. The equation is: $\text{sign} \left(\sum_{\alpha_n > 0} \alpha_n y_n e^{-\gamma \|x - x_n\|^2} + b \right)$. On the right, 'RBF' is underlined and followed by an arrow pointing to the left. The equation is: $\text{sign} \left(\sum_{k=1}^K w_k e^{-\gamma \|x - x_k\|^2} + b \right)$. A curved arrow points from the SVM equation to the RBF equation, and another curved arrow points from the RBF equation back to the SVM equation, indicating a comparison or mapping between the two.

Figure 7: Comparing the SVM equation with the RBF classification equation

5 RBF as Regularizer

Regularization tries to find smooth hypothesis. The smoothness can be expressed in terms of the k^{th} derivative of the hypothesis with respect to the data points i.e.

$$\frac{\partial^k h}{\partial x^k}$$

The more abrupt are the derivatives, less smooth is the hypothesis $h(x)$.

When we want to use interpret RBF as a regularizer, we would obviously want to minimize the error which can be expressed as minimizing the below term:

$$\sum_{n=1}^N (h(x) - y_n)^2$$

We also need to consider the regularization parameter λ . and in order to bring smoothness, we also need to minimize $\frac{\partial^k h}{\partial x^k}$

So, when we combine all this, we get equation 7 as our error function that we have to minimize.

$$E = \sum_{n=1}^N (h(x) - y_n)^2 + \lambda \sum_{k=0}^{\infty} a_k \int_{-\infty}^{\infty} \left(\frac{\partial^k h}{\partial x^k} \right)^2 dx \quad (7)$$

Minimizing the error function of equation 7 minimizes the error as well as the abruptness of the hypothesis. Depending on the regularization constant (λ), we will get a smoother hypothesis directly from RBF as it can be derived that when we try to solve equation 7 by minimizing the error, the hypothesis finally becomes an RBF.

6 Summary

RBF is a very generic model:

1. It can be used for regression and as well as classification.
2. We can also use RBF for clustering. RBF can be represented as a 2 layer network where it resembles the traditional neural network.
3. Traditionally, RBF has been used in SVMs to form the kernels.
4. RBF is very smooth in terms of building a hypotheses and it is a very good regularization function.