Back to the future: Radial Basis Function networks revisited

Qichao Que, Mikhail Belkin
Department of Computer Science and Engineering
Ohio State University
Columbus, OH 43210
que, mbelkin@cse.ohio-state.edu

Abstract

Radial Basis Function (RBF) networks are a classical family of algorithms for supervised learning. The most popular approach for training RBF networks has relied on kernel methods using regularization based on a norm in a Reproducing Kernel Hilbert Space (RKHS), which is a principled and empirically successful framework. In this paper we aim to revisit some of the older approaches to training the RBF networks from a more modern perspective. Specifically, we analyze two common regularization procedures, one based on the square norm of the coefficients in the network and another one using centers obtained by k-means clustering. We provide a theoretical analysis of these methods as well as a number of experimental results, pointing out very competitive experimental performance as well as certain advantages over the standard kernel methods in terms of both flexibility (incorporating of unlabeled data) and computational complexity. Finally, our results shed light on some impressive recent successes of using soft k-means features for image recognition and other tasks.

1 Introduction

Radial Basis Function (RBF) networks are a classical family of algorithms for supervised learning. The goal of RBF is to approximate the target function through a linear combination of radial kernels, such as Gaussian. Thus the output of an RBF network learning algorithm typically consists of a set of centers and weights for these functions. The key aspect of any RBF network algorithm is capacity control. It is easy to see that any input data (x_i, y_i) can be fitted *exactly* by allowing every data point to be a center and choosing appropriate coefficients. That, of course, is *overfitting* and thus RBF networks need to be regularized. A number of regularization approaches have been proposed previously with various theoretical properties, computational complexity and empirical performance. By far the most successful approach to regularizing RBF's has been based on the *kernel machines*, such as kernel SVM's (K-SVM) or kernel regularized least squares (K-RLS) algorithm. In these approaches the function space is constrained by the norm in a Reproducing Kernel Hilbert Space (RKHS). While kernel methods are often considered to be a different class of algorithms, they are, in fact, types of RBF networks when used with a radial kernel. The kernel methods have become very popular, easily eclipsing earlier RBF algorithms, due to their elegant mathematical formulation grounded in classical functional analysis, the convex nature of optimizations involved and to their strong empirical performance.

In this paper we take a step back by revisiting two common methods for training RBF networks suggested before the success of kernel machines. Specifically, we look at regularization by the squared norm of the coefficients in an RBF network and on selecting centers through k-means clustering. We highlight certain advantages of these approaches compared to the standard kernel methods both in terms of flexibility (by easily incorporating unlabeled data) and scalability for large datasets. Our contributions could be summarized as follows.

- We provide a theoretical analysis of RBF networks whose centers are chosen at random from the same probability distribution as the input data and which is regularized based on the l^2 norm of the coefficient vector. In particular this setting applies to the case when the set of the centers is the training set. We provide generalization bounds under the usual statistical assumptions. It follows from our analysis that the asymptotic convergence rate of this methods equals to the standard rate obtained for kernel machines.
- We analyze another common form of RBF networks, where the centers are obtained from a k-means algorithm. Inspired by its potential for large-scale learning problem, we provide a bound on the generalization error in terms of the quantization error of the output of k-means algorithm. Interestingly, its generalization error suggests that with a proper choice of k, it has same asymptotic rate with the full RBF network, but requires a much less computational cost. It sheds light on the strong performance shown by soft k-means feature embedding used in [5, 6], which are closely related to RBF networks with a certain radial kernel.
- We discuss certain advantages of RBF networks over the standard kernel methods. In particular semi-supervised learning for these RBF's is achieved naturally and without any extra hyperparameters as unlabeled data can simply be used as centers. We discuss why adding unlabeled data can be helpful and provide experimental support for this observation.
- Finally, we provide a number of experimental results to show that RBF's provide comparable performance to the kernel machines.

Related Work. There is a large body of work investigating RBF networks from many different perspectives. Proposed in [2], RBF networks were introduced as a function approximation method and interpreted as an artificial neural networks. Analysis of RBF networks and the connections to approximation theory were explored [13]. Results in [11, 12] showed that any function in the functional space $L^p(\mathbb{R}^d)$ could be approximated by a RBF network arbitrarily well, under a very mild condition on the RBF function. To control the approximation power of the RBF network and avoid overfitting, [10] suggested that RBF network could be regularized by the squared norm of the coefficients (ridge regression) or subset selection. Several other related forms of regularization such as using the information curvature information in [1], have also been proposed. A number of approaches exist for selecting a subset of centers for building a parsimonious RBF network, including [4, 8, 3, 9]. Furthermore, there have been work on the statistical properties of RBF networks. In particularly the insightful work [7] investigated the generalization error of RBF networks and provided generalization guarantees in terms of the number of training data and the number of function basis in the setting of the statistical learning theory. The version of RBF considered in [7] involved a non-convex optimization over the set of centers.

2 Main Results

Generalization analysis for RBF networks. First, we will concentrate on a particularly simple form of RBF netowrk using a positive semi-definite RBF kernel K, where the centers are the data points and the regularization term equals to the square norm of the coefficients. Choosing the loss function L, we can train the model through minimizing the empirical risk on the training data:

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}\in\mathbb{R}^n} \frac{1}{n}\sum_{i=1}^{n} L(f(x_i), y_i) + \frac{\lambda}{n}\sum_{i=1}^{n} w_i^2, \quad \text{where } f(x) = \frac{1}{n}\sum_{j=1}^{n} w_j K(x, x_j). \tag{1}$$

Denote the output classifier as f_n^*. Using the square loss $L(f(x), y) = (f(x) - y)^2$, the solution is $\boldsymbol{w}^* = \left(\frac{1}{n}\boldsymbol{K}^T\boldsymbol{K} + n\lambda\boldsymbol{I}\right)^{-1}\boldsymbol{K}^T\boldsymbol{y}$, where $\boldsymbol{K}$ is a $n \times n$ matrix with $\boldsymbol{K}_{ij} = K(x_i, x_j)$. The classifier function $f_n^*(x) = \frac{1}{n}\sum_{i=1}^{n} w_i K(x, x_i)$, is equivalent to the solution to a regularized least-square kernel machine with a data-dependent kernel $\hat{K}_W$, where

$$\hat{K}_W(x, z) = \frac{1}{n}\sum_{i=1}^{n} K(x, x_i)K(z, x_i). \tag{2}$$

With a proper set of assumptions, we show that this discrete algorithm has a continuous counterpart for solving a Fredholm integral equation. Given n training data, $(x_1, y_1), \ldots, (x_n, y_n)$, let us assume that x_i are i.i.d. samples from a probability distribution p and the outputs y_i are deterministic, that is $y_i = g(x_i)$. We assume the target function g is bounded by $|g(x)| \leq M$. Consider the following continuous optimization algorithm for approximating the target function $g(x)$,

$$w^* = \min_{w \in L_p^2} \|\mathcal{K}_p w - g\|_p^2 + \lambda \|w\|_p^2, \text{ with the approximator function } f^* = \mathcal{K}_p w^*. \tag{3}$$

The norm $\|\cdot\|_p$ is defined by $\|w\|_p = \left(\int w(x)^2 p(x) dx\right)^{\frac{1}{2}}$ and $L_p^2 = \{w, \|w\|_p^2 < \infty\}$. $\mathcal{H}$ is the RKHS with the RBF kernel K, which is assumed to be positive semi-definite. $\mathcal{K}_p : L_p^2 \to \mathcal{H}$ is an integral operator associated with the kernel K, defined by $\mathcal{K}_p w(x) = \int K(x,u) w(u) p(u) du$. Using f^*, we can decompose the generalization error $\|g - f_n^*\|_p$ into approximation error and estimation error, given in Proposition 1 and Theorem 2 respectively.

Using the techniques in [14], we have the following results regarding the approximation error $\|g - \mathcal{K}_p w^*\|_p$ and estimation error $\|f_n^* - f^*\|_p$.

Theorem 1. *Assuming the target function g satisfies $\|{\mathcal{K}_p}^{-r} g\|_p < \infty$ for $0 < r \leq 2$, we have*

$$\|g - \mathcal{K}_p w^*\|_p \leq \lambda^{\frac{r}{2}} \|{\mathcal{K}_p}^{-r} g\|_p. \tag{4}$$

Theorem 2. *Assuming the target function is uniformly bounded, that is $g(x) < M$ for any x, with probability at least $1 - 2e^{-\tau}$, we have*

$$\|f_n^* - f^*\|_p \leq \frac{3\kappa^2 M(\sqrt{2\tau} + 1 + \sqrt{8\tau})}{\lambda \sqrt{n}} + \frac{4\kappa^2 M \tau}{3\lambda n} + \frac{4\kappa^3 M \tau}{3\lambda^{\frac{3}{2}} n} \tag{5}$$

where $\kappa = \max_x K(x,x)$.

Combine the results from Eqn. (4) and (5), the generalization error $\|f_n^* - g\|_p$ converges to 0 with rate $O(n^{-\frac{r}{2r+4}})$ in probability as $n \to \infty$. In particular, when g is in the range of $\mathcal{K}_p^2$, the convergence rate is $O(n^{-\frac{1}{4}})$. This is the same rate as the one for the least square kernel machine given in [14].

RBF networks for semi-supervised learning. We will highlight the difference between RBF's and the standard kernel methods in the semi-supervised setting. We first observe that using unlabeled data in the RBF setting is a simple matter of adding additional centers for unlabeled points, writing $f(x) = \frac{1}{n} \sum_{i=1}^{n+m} w_i h(\|x - x_i\|)$ where m is the number of unlabeled points in Eqn. (1). The regularization penalty constrains the complexity of the function class to avoid overfitting. It is easy to see that unlabeled data changes the resulting RBF classifier. A natural question of comparison to kernel machines arises. We can put $f(x) = \frac{1}{n} \sum_{j=1}^{n+m} w_j h(\|x - x_i\|)$ in the standard kernel framework, where the only difference will be using the norm $\|f\|_{\mathcal{H}} = \boldsymbol{w}^T K \boldsymbol{w}$ (instead of $\|f\|_{\mathcal{H}_w} = \boldsymbol{w}^T \boldsymbol{w}$ for RBF). However it follows from the representer theorem[1] that the output of a kernel machine will ignore the unlabeled data by putting zero weights on unlabeled points.

This difference could also be illustrated by a simple example. Consider a classification problem with the (marginal) data distribution $p(x) = N(\mathbf{0}, \text{diag}([9,1]))$. Given two labeled points, positive one $x_p = (-4,3)$ and negative one $x_n = (4,-3)$, consider two candidate classifier functions showed in Figure 1. It is clear that both f_1 and f_2 have 0 empirical risk on the two labeled points x_p and x_n. However, their norms are different in the standard RKHS $\mathcal{H}$ corresponding to K and the data-dependent RKHS $\mathcal{H}_w$ corresponding to the kernel $\hat{K}_W$ in Eqn. (2). In particular, $\|f_1\|_{\mathcal{H}} = \|f_2\|_{\mathcal{H}}$ and $\frac{\|f_1\|_{\mathcal{H}_w}}{\|f_1\|_{\mathcal{H}_w}} \approx 54.6$. Thus, f_1 and f_2 are equivalently good solutions from the point of a kernel machine with kernel K, as both the empirical risk and regularization term are the same. On the other hand, the solution f_1 has a much higher regularization penalty and the RBF framework would select f_2 over f_1.

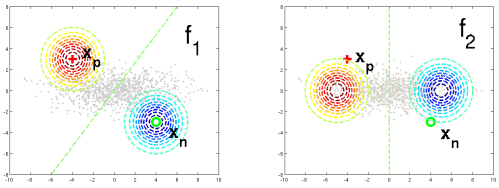

Figure 1: Contours and classification boundaries for f_1 (left) and f_2 (right). Labeled points x_p and x_n, gray are sampled from p. $\|f_1\|_{\mathcal{H}} = \|f_2\|_{\mathcal{H}}$, however $\|f_1\|_{\mathcal{H}_w} \gg \|f_2\|_{\mathcal{H}_w}$

k-means RBF networks for large-scale learning. In this section, we will give analysis for another form of RBF network, which uses k-means centers as the basis. As k-means provides a concise

[1] Observe that the solution of the kernel machine is optimal over the whole RKHS space. As f belongs to the RKHS, the extra centers will make no difference in the final form of the solution.

representation of the data, it is natural to replace the training set with its k-means centers for radial basis functions. Given the cluster weights, $P_n(C_i) = \#\{x_j \in C_i\}/n$, the classifier is learned by

$$\boldsymbol{w}^*_{k,p} = \arg\min_{\boldsymbol{w}\in\mathbb{R}^k} \frac{1}{n}\sum_{i=1}^{n} L(f(x_i), y_i) + \frac{\lambda}{k}\sum_{i=1}^{k} P_n(C_i)w_i^2, \text{ where } f(x) = \sum_{i=1}^{k} w_i h(\|x - c_i\|)P_n(C_i).$$

The output classifier is denoted by $f_{k,p}$. Note that this algorithm is equivalent to train a linear classifier given the non-linear feature embeddings based on the k-means centers. Given this explicit feature mebedding, the algorithm could be much more efficient for both training and testing when we have $k \ll n$, because the computational cost for evaluating the classifier for a single point is on the order of $O(k)$, instead of $O(n)$ in traditional kernel methods. Moreover, this RBF network consists of the k centers and their corresponding coefficients, thus takes memoroy of size $O(kd+k)$ for storing the model, which is also much smaller than $O(nd+n)$ for a regular kernel methods.

More importantly, we also have the following result regarding its generalization error.

Theorem 3. *Suppose the target function is uniformly bounded $g(x) \leq M$ for any x, and the RBF kernel K is translation invariant such that $K(x,z) = h(\|x-z\|^2)$ with a monotonic decreasing function h satisfying the Lipschitz condition: $|h(v) - h(u)| \leq L|u-v|$. For the estimation error $\|f^* - f^*_{k,p}\|_p$, we have*

$$\|f^*_{k,p} - f^*\|_p \leq \left(\frac{\kappa^2 + \kappa^{\frac{5}{2}}}{\lambda} + \frac{2\kappa^3}{\lambda^{\frac{3}{2}}}\right)\frac{\sqrt{2\tau}M}{\sqrt{n}} + 8L\left(\frac{\kappa^2}{\lambda} + \frac{\kappa^3}{\lambda^{\frac{3}{2}}}\right)MQ_k(\mathcal{C})$$

with probability at least $1 - 2e^{-\tau}$.

In addition to the error term depending on n, the estimation error bound for k-means RBF network also contains a term that depends on the quantization error $Q_k(\mathcal{C}_k)$. Interestingly, it suggests that as long as the quantization error $Q_k(\mathcal{C}_k) = O(n^{-\frac{1}{2}})$, k-means RBF network could still perform as well as the full RBF network asymptotically. Since the quantization error $Q_k(\mathcal{C}_k)$ decreases to 0 as $k \to n$, a proper choice of $k < n$ could always be found such that $Q_k(\mathcal{C}_k) = O(n^{-\frac{1}{2}})$.

Experiments. We compare the empirical performance between the kernel machines and the RBF network. We see that RBF network gives comparable performance with the kernel machines and RBF network performs consistently better in the case of semi-supervised learning, except for one data set, without add extra hyperparameters.

	K-SVM	K-RLSC	RBFN-hinge	RBFN-LS	k-means RBFN
MNIST	1.5	**1.32**	1.72	1.35	3.3
MNIST-rand	15.8	13.7	16.6	14.3	**10.6**
MNIST-img	23.4	20.9	23.4	**20.8**	25.5
SVHN 60k	20.5	**18.7**	24.2	18.8	26.1
Adult	**14.5**	15.6	15.8	15.6	15.6
Cover Type	28.4	27.8	28.0	**27.6**	35.7
Cod-RNA	4.60	**3.55**	3.94	3.73	4.05

Table 1: Classification **Errors** (%) for *supervised learning* with whole training data using K-SVM, K-RLSC, RBF network with hinge loss and square loss, and$k$-means RBF for $k = 1000$.

	K-SVM	K-RLSC	RBFN-hinge	RBFN-LS
MNIST	26.8	26.0	27.4	**23.3**
MNIST rand	51.4	48.5	**35.9**	38.3
MNIST img	59.2	52.7	63.1	**51.0**
SVHN 60k	75.5	73.1	79.5	**72.4**
Adult	18.8	19.1	19.5	**18.4**
Cover Type	58.3	58.7	**57.3**	57.9
Cod-RNA	6.62	**6.30**	7.83	7.12

Table 2: Classification **Errors** (%) for *semi-supervised learning*, with 100 labeled points, using K-SVM, K-RLSC, ridge RBF network with hinge loss and least-square loss.

References

[1] Chris Bishop. Improving the generalization properties of radial basis function neural networks. *Neural computation*, 3(4):579–588, 1991.

[2] David S Broomhead and David Lowe. Radial basis functions, multi-variable functional interpolation and adaptive networks. Technical report, DTIC Document, 1988.

[3] S Chen, ES Chng, and K Alkadhimi. Regularized orthogonal least squares algorithm for constructing radial basis function networks. *International Journal of Control*, 64(5):829–837, 1996.

[4] Sheng Chen, Colin FN Cowan, and Peter M Grant. Orthogonal least squares learning algorithm for radial basis function networks. *Neural Networks, IEEE Transactions on*, 2(2):302–309, 1991.

[5] Adam Coates, Andrew Y Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *AISTATS*, 2011.

[6] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS 2011, Workshop on deep learning and unsupervised feature learning*, 2011.

[7] Partha Niyogi and Federico Girosi. On the relationship between generalization error, hypothesis complexity, and sample complexity for radial basis functions. *Neural Computation*, 8(4):819–842, 1996.

[8] Mark JL Orr. Regularised centre recruitment in radial basis function networks. In *Centre for Cognitive Science, Edinburgh University*. Citeseer, 1993.

[9] Mark JL Orr. Regularization in the selection of radial basis function centers. *Neural computation*, 7(3):606–623, 1995.

[10] Mark JL Orr et al. Introduction to radial basis function networks, 1996.

[11] Jooyoung Park and Irwin W Sandberg. Universal approximation using radial-basis-function networks. *Neural computation*, 3(2):246–257, 1991.

[12] Jooyoung Park and Irwin W Sandberg. Approximation and radial-basis-function networks. *Neural computation*, 5(2):305–316, 1993.

[13] Tomaso Poggio and Federico Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, 1990.

[14] Steve Smale and Ding-Xuan Zhou. Shannon sampling ii: Connections to learning theory. *Applied and Computational Harmonic Analysis*, 19(3):285–302, 2005.