RBFAR-OLS

This repository contains code for my research on a modified method of OLS (orthogonal least squares), applied to RBF-AR (radial basis function - autoregressive) models.

Relevant Papers

- Original OLS algorithm for RBF networks
- One of the articles that introduced RBF-AR
- RBF-AR parameter optimization
- RBF-AR applications
- Adaptive learning for RBF-AR methods

Formulation of the problem

$$\begin{aligned} X &\in \mathbb{R}^{l\times n} \\ d &\in \mathbb{R}^{l\times 1} \\ \Psi : &\mathbb{R}^n \to [0,1]^m, \\ \Phi_i &= \Psi(X_i) \\ \mathrm{N} &\in \mathbb{R}^{m\times n} \end{aligned}$$

Note: the target variable d should be normalized, so it has a mean of 0. These models only work well for stationary data, but for data with a trend, this normalization should be done via a trend removal method.

First approach (locally pre-trained coefficients)

$$\begin{aligned} y_i &= \Psi(X_i)\cdot(X_i\cdot \mathrm{N}^T) \\ \implies d \approx y &= (\Phi \odot (X\cdot \mathrm{N}^T))\cdot \underline{1} \\ \implies \mathrm{N} &= ? \end{aligned}$$

Solution: first fit an AR model to each candidate centre, and then select from them and assign weights to each selected one.

Let Ξ_i be the set of training sample indicies associated with the ith candidate centre. We define Ξ_i as follows:

$$\Xi_i = \{j \in [1,l]\,|\Psi_i(X_j) \geq \rho\},$$

where ρ is a threshold for the candidate centre selection, and $\Psi_i(X_j)$ is the ith element of $\Psi(X_j)$

Let ν_i be the coefficient-vector associated with the ith candidate centre.

We fit each linear AR model separately, using a standard least squares approach:

$$\hat{\nu}_i = \arg\min_{\nu_i} \sum_{j \in \Xi_i} (d_j - X_j \cdot \nu_i)^2$$

Let N be the matrix of these coefficients:

$$\mathrm{N} = \begin{bmatrix} \hat{\nu}_1^T \\ \hat{\nu}_2^T \\ \vdots \\ \hat{\nu}_m^T \end{bmatrix}$$

Now we can calculate the activation matrix Σ. This matrix represents the output for each training point i and each linear AR model, associated with the jth candidate centre:

$$\begin{aligned} \Sigma &= \Phi \odot (X \cdot \mathrm{N}^T) \\ \Sigma_{i,j} &= \Psi_j(X_i) \cdot (X_i \cdot \hat{\nu}_j^T) \\ \Sigma &= \begin{bmatrix} \Psi_1(X_1) \cdot (X_1 \cdot \hat{\nu}_1^T) & \Psi_2(X_1) \cdot (X_1 \cdot \hat{\nu}_2^T) & \dots & \Psi_m(X_1) \cdot (X_1 \cdot \hat{\nu}_m^T) \\ \Psi_1(X_2) \cdot (X_2 \cdot \hat{\nu}_1^T) & \Psi_2(X_2) \cdot (X_2 \cdot \hat{\nu}_2^T) & \dots & \Psi_m(X_2) \cdot (X_2 \cdot \hat{\nu}_m^T) \\ \vdots & \vdots & \ddots & \vdots \\ \Psi_1(X_l) \cdot (X_l \cdot \hat{\nu}_1^T) & \Psi_2(X_l) \cdot (X_l \cdot \hat{\nu}_2^T) & \dots & \Psi_m(X_l) \cdot (X_l \cdot \hat{\nu}_m^T) \end{bmatrix} \end{aligned}$$

From now, we can continue with the OLS approach, such that the activations, normally denoted by P are given by the Σ matrix ($P = \Sigma$).

We then select the centres and assign coefficients (θ_i) to each selected centre using the OLS approach.

$$d = \Sigma \cdot \theta + e$$

The details are left out here, as they are similar to the OLS approach described below.

Having completed this step, we get:

$$y = (\Phi \odot (X \cdot \mathrm{N}^T)) \cdot \theta$$

This can be rewritten as the original equation, with the weights modified:

$$y = (\Phi \odot (X \cdot \tilde{\mathrm{N}}^T)) \cdot \underline{1},$$

where $\tilde{\mathrm{N}} = \mathrm{N} \odot \theta$

Second approach (no pre-training)

We flatten the N matrix into a vector ν and rewrite the equation as follows:

$$P \in \mathbb{R}^{l \times (n \cdot m)}$$
$$P_{i,(j-1)\cdot m+k} = \Phi_{i,j} \cdot X_{i,k}$$
$$P = \begin{bmatrix} \Phi_{1,1}X_{1,1} & \Phi_{1,2}X_{1,1} & \dots & \Phi_{1,m}X_{1,n} \\ \Phi_{2,1}X_{2,1} & \Phi_{2,2}X_{2,1} & \dots & \Phi_{2,m}X_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \Phi_{l,1}X_{l,1} & \Phi_{l,2}X_{l,1} & \dots & \Phi_{l,m}X_{l,n} \end{bmatrix}$$

let p_i be the ith column of P

$$\nu = [\mathrm{N}_{1,1}, \mathrm{N}_{1,2}, \dots, \mathrm{N}_{1,n}, \mathrm{N}_{2,1}, \dots, \mathrm{N}_{m,n}]^T$$

$$\implies d \approx y = P \cdot \nu$$

$$A \in \mathbb{R}^{M_S \times M_S},$$

where M_S is the number of selected centres

$$A = \begin{bmatrix} 1 & \alpha_{1,1} & \dots & \alpha_{1,M_S} \\ 0 & 1 & \dots & \alpha_{2,M_S} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

OLS criteria:

- $P = W \cdot A$, where A is an upper triangular matrix, W has orthogonal columns
- $d = W \cdot g + e$

Thus:

$$d \approx W \cdot g = P \cdot \nu = W \cdot A \cdot \nu$$

Centre selection

Initialization:

- define candidate centres C, and calculate Φ accordingly

- calculate P as defined above
- initialize the residuals: $d^{(0)} = d$
- initialize the selected centres: $C^{(0)} = \emptyset$
- initialize W to an empty matrix: $W^{(0)} = \emptyset$
- initialize g to an empty vector: $g^{(0)} = \emptyset$
- initialize the selected indicies $I^{(0)}_{\text{selected}} = \emptyset$

At each iteration ($k \in [1, |C_{\textbf{candidates}}|]$):

$\forall i \in [1, |C_{\textbf{candidates}}|] \setminus I^{(k-1)}_{\textbf{selected}}$:

$$w_i = p_i - \sum_{j=1}^{k-1} \alpha_{i,j} w_j,$$
$$\text{where } \alpha_{i,j} = \frac{p_i^T \cdot w_j}{||w_j||^2}$$

$$g_i = \frac{w_i^T \cdot d^{(k-1)}}{||w_i||^2}$$

$$err_i = g_i^2 \cdot \frac{||w_i||^2}{||d^{(k-1)}||^2} = \frac{\left(w_i^T \cdot d^{(k-1)}\right)^2}{||w_i||^2 \cdot ||d^{(k-1)}||^2}$$

Select the centre c_i with the highest err_i.

$$I^{(k)}_{\text{selected}} = I^{(k-1)}_{\text{selected}} \cup \{i\}$$
$$C^{(k)} = C^{(k-1)} \cup \{c_i\}$$
$$W^{(k)} = W^{(k-1)} \oplus w_i$$
$$g^{(k)} = g^{(k-1)} \oplus g_i$$
$$d^{(k)} = d^{(k-1)} - w_i \cdot g_i$$

Stopping criteria:

1. Maximum number of iterations reached: $k = K_{max}$
2. Convergence: $||d^{(k)}||^2 < \epsilon$
3. No significant improvement: $\frac{||d^{(k)}||^2}{||d^{(k-1)}||^2} < \delta$

If any of the stopping criteria is met, the algorithm terminates with $M_S = k$.

Finding the coefficients

$$d = W \cdot g + e = P \cdot \nu = W \cdot A \cdot \nu + e \approx W \cdot \hat{g}$$

$$\text{W has orthogonal columns} \implies W^T \cdot W = H,$$

where H is a diagonal matrix, and thus invertible

$$\text{We can calculate } \hat{g} \text{ by } H^{-1} \cdot W^T \cdot d = \hat{g}$$

$$\begin{aligned} \implies A \cdot \hat{\nu} &= H^{-1} \cdot W^T \cdot d \\ A \cdot \hat{\nu} &= \hat{g} \end{aligned}$$

which is easily and efficiently solvable, due to A being an upper triangular matrix.

SVD-based center selection

Let us assume, we have calculated the activation matrix P, using either approach. Let us then compute its SVD:

$$P = Q \cdot \Sigma \cdot V^T,$$

where Q and V are orthogonal matrices, and Σ is a diagonal matrix containing the singular values of P.

We then get the following equation for the optimal weights:

$$\hat{\nu} = \text{argmin}_\nu \{ \|d - P \cdot \nu\|_2^2 + \alpha \|\nu\|_2^2 \},$$

where α is a regularization parameter.
We can transform this as follows:

$$
\begin{aligned}
\hat{\nu} &= \operatorname{argmin}_{\nu}\{\|d - P\cdot\nu\|_2^2 + \alpha\|\nu\|_2^2\} \\
&= \operatorname{argmin}_{\nu}\{d^Td - 2\nu^TP^Td + \nu^TP^TP\nu + \alpha\nu^T\nu\} \\
\implies \frac{\delta}{\delta\nu} &= -2P^Td + 2P^TP\nu + 2\alpha\nu = 0 \\
\implies \hat{\nu} &= \left(P^TP + \alpha I\right)^{-1} P^Td \\
&= \left(V\Sigma^TQ^TQ\Sigma V^T + \alpha VV^T\right)^{-1} V\Sigma^TQ^Td \\
Q \text{ is orthonormal} \implies Q^TQ = I \implies \hat{\nu} &= \left(V\Sigma^TI\Sigma V^T + \alpha VV^T\right)^{-1} V\Sigma^TQ^Td \\
&= \left(V\Sigma^T\Sigma V^T + \alpha VV^T\right)^{-1} V\Sigma^TQ^Td \\
&= \left(V\left(\Sigma^T\Sigma + \alpha I\right)V^T\right)^{-1} V\Sigma^TQ^Td \\
V \text{ is orthogonal} \implies V^{-1} = V^T \implies \hat{\nu} &= V\left(\Sigma^T\Sigma + \alpha I\right)^{-1} V^TV\Sigma^TQ^Td \\
V \text{ is orthogonal} \implies V^TV = I \implies \hat{\nu} &= V\left(\Sigma^T\Sigma + \alpha I\right)^{-1} \Sigma^TQ^Td \\
&= V\begin{bmatrix} \frac{\sigma_1}{\sigma_1^2+\alpha} & 0 & \cdots & 0 \\ 0 & \frac{\sigma_2}{\sigma_2^2+\alpha} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{\sigma_n}{\sigma_n^2+\alpha} \end{bmatrix} Q^Td
\end{aligned}
$$

$$
\boxed{\hat{\nu} = V\left[\begin{matrix} \frac{\sigma_1}{\sigma_1^2+\alpha} & 0 \\ 0 & \frac{\sigma_2}{\sigma_2^2+\alpha} \\ \vdots & \vdots \\ 0 & 0 \end{matrix}\right.}
$$

This means that:

$$
\begin{aligned}
y &= P\hat{\nu} \\
&= Q\Sigma V^T\hat{\nu} \\
&= Q\Sigma V^TV\begin{bmatrix} \frac{\sigma_1}{\sigma_1^2+\alpha} & 0 & \cdots & 0 \\ 0 & \frac{\sigma_2}{\sigma_2^2+\alpha} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{\sigma_n}{\sigma_n^2+\alpha} \end{bmatrix} Q^Td \\
&= Q\Sigma\begin{bmatrix} \frac{\sigma_1}{\sigma_1^2+\alpha} & 0 & \cdots & 0 \\ 0 & \frac{\sigma_2}{\sigma_2^2+\alpha} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{\sigma_n}{\sigma_n^2+\alpha} \end{bmatrix} Q^Td \\
&= Q\begin{bmatrix} \frac{\sigma_1^2}{\sigma_1^2+\alpha} & 0 & \cdots & 0 \\ 0 & \frac{\sigma_2^2}{\sigma_2^2+\alpha} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{\sigma_n^2}{\sigma_n^2+\alpha} \end{bmatrix} Q^Td
\end{aligned}
$$

Let $\sigma_i' = \frac{\sigma_i^2}{\sigma_i^2+\alpha}$, and let Σ' be defined as follows:

$$\begin{aligned}\Sigma' &= \begin{bmatrix} \frac{\sigma_1^2}{\sigma_1^2+\alpha} & 0 & \cdots & 0 \\ 0 & \frac{\sigma_2^2}{\sigma_2^2+\alpha} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{\sigma_n^2}{\sigma_n^2+\alpha} \end{bmatrix} \\ &= \begin{bmatrix} \sigma_1' & 0 & \cdots & 0 \\ 0 & \sigma_2' & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n' \end{bmatrix}\end{aligned}$$

Thus we get:

$$\boxed{y = Q\Sigma' Q^T d}$$

The selection algorithm

We iteratively select the first k columns of Q and Σ', which correspond to the k largest singular values. This gives us a reduced representation of the data, focusing on the most important features. We then calculate the corresponding error and stop when the error is below a certain threshold.

$$\begin{aligned}\text{Error} &= \|d - P\hat{\nu}\|_2^2 \\ &= \|d - Q\Sigma' Q^T d\|_2^2\end{aligned}$$

Let us define $Q^{(k)}$ as the matrix formed by the first k columns of Q and zeros elsewhere, and $\Sigma^{(k)}$ as the matrix formed by the first k rows and columns of Σ' and zeros in the rest. Then we can rewrite the error as follows:

$$\text{Error}_k = \|d - Q^{(k)}\Sigma'^{(k)}Q^{(k)T}d\|_2^2$$

Our stopping criterion is based on the error:

$$\text{Error}_k < \epsilon$$

where ϵ is a predefined threshold. When this condition is met, we stop the selection process and use the current $Q^{(k)}$ and $\Sigma'^{(k)}$ for our reduced representation.

We can refine this process by optimizing the calculation of Error_k. We can achieve this by only calculating the differences of the errors.

$$\begin{aligned}
\text{Error}_0 &= \|d\|_2^2 \\
\text{Error}_k - \text{Error}_{k-1} &= \|d - Q^{(k)}\Sigma'^{(k)}Q^{(k)T}d\|_2^2 - \|d - Q^{(k-1)}\Sigma'^{(k-1)}Q^{(k-1)T}d\|_2^2 \\
&= \left(\|d\|_2^2 - 2\sum_{i=1}^{k} d^T q_i \sigma_i' q_i^T d + \sum_{i=1}^{k} d^T q_i^T \sigma_i' q_i q_i \sigma_i' q_i^T d\right) \\
&\quad - \left(\|d\|_2^2 - 2\sum_{i=1}^{k-1} d^T q_i \sigma_i' q_i^T d + \sum_{i=1}^{k-1} d^T q_i^T \sigma_i' q_i q_i \sigma_i' q_i^T d\right) \\
&= \left(\|d\|_2^2 - 2\sum_{i=1}^{k} \sigma_i' \left(q_i^T d\right)^2 + \sum_{i=1}^{k} \sigma_i'^2 \left(q_i^T d\right)^2\right) \\
&\quad - \left(\|d\|_2^2 - 2\sum_{i=1}^{k-1} \sigma_i' \left(q_i^T d\right)^2 + \sum_{i=1}^{k-1} \sigma_i'^2 \left(q_i^T d\right)^2\right) \\
&= \left(\|d\|_2^2 + \sum_{i=1}^{k} \sigma_i' \left(\sigma_i' - 2\right) \left(q_i^T d\right)^2\right) \\
&\quad - \left(\|d\|_2^2 + \sum_{i=1}^{k-1} \sigma_i' \left(\sigma_i' - 2\right) \left(q_i^T d\right)^2\right) \\
&= \sigma_k' \left(\sigma_k' - 2\right) \left(q_k^T d\right)^2
\end{aligned}$$

$$\boxed{\text{Error}_k - \text{Error}_{k-1} = \sigma_k' \left(\sigma_k' - 2\right) \left(q_k^T d\right)^2}$$

To be able to consistently use the same sort of ϵ values, we can normalize the error by dividing it by the initial error:

$$\text{Normalized Error}_k = \frac{\text{Error}_k}{\text{Error}_0} = \frac{\text{Error}_k}{\|d\|_2^2}$$

Thus, we get our optimized centre-selection algorithm:

```
1. Normalized $\text{Error}_0 := 1$
2. While k $< |C_{\text{candidates}}|$:
    1. Normalized $\text{Error}_k :=$ Normalized $\text{Error}_{k-1} + \sigma_k' \left(\sigma_k' - 2\right) \frac{\left(q_k^T d\right)^2}{\|d\|_2^2}$
    2. If Normalized $\text{Error}_k < \epsilon$ then:
        1. Stop
```

TODO

- regularizations
    - L1 regularization
- non global sigma estimation
- multistep forecast