LOGISTIC REGRESSION and KERNEL LOGISTIC REGRESSION

Ezukwoke K.I[1], Samaneh M.J[2]

[1, 2] Department of Computer Science
Machine Learning and Data Mining
{ifeanyi.ezukwoke, samaneh}@etu.univ-st-etienne.fr
University Jean Monnet, Saint-Etienne, France

Abstract

Kernel principal component analysis (kernel-PCA) is a prominent non-linear extension of one of the most used classiical dimensionality reduction algorithms (PCA-Principal Component Analysis). In this paper, we present an inhaustive comparison between the classical PCA and the kernel-PCA. We use the moon dataset

Keywords
Dimensionality reduction, PCA, kernel-PCA

1 INTRODUCTION

Logistic regression is a binary classification algorithm. The algorithm is capable of classifying linearly separable dataset. The linear version of logistic regression is however not able to accurately classify non-linear data, hence its kernel version.

Using the Iterative Reweighted Least Square method (IRLS) proposed by Zhu et al.[2]. an extension of the Support vector machine was introduced to logistic regression (Import vector machines). This method uses much more smaller data points as support vectors than SVMs and hence, outperforms SVMs for classification purposes.

2 CLASSIFICATION

2.1 LOGISTIC REGRESSION

Classical logistic regression is define using the binomial distribution as

$$l(\beta_0, \beta) = \Pi_{i=1}^{n} p(x_i)^y (1 - p(x_i)^{(1-y_i)}) \tag{1}$$

and the log-likelihood follows as

$$L(\beta_0, \beta) = \sum_{i=1}^{n} y_i logp(x_i) + (1 - y_i) \\ log(1 - p(x_i))$$

where

$$p(x; \beta) = \frac{1}{1 + e^{-x \cdot \beta}} \tag{2}$$

and

$$1 - p(x; \beta) = \frac{1}{1 + e^{x \cdot \beta}} \tag{3}$$

2.2 KERNEL LOGISTIC REGRESSION

However, classical logistic regression will fail to classify accurately non-linearly separable data, hence its kernel version.

The vector space can be expressed as a linear combination of the input vectors such that

$$\beta = \sum_{i=1}^{N} \alpha_i \phi(\mathbf{x_i}) \tag{4}$$

where $\alpha \in \mathbb{R}^{nx1}$ is the dual variable. The function $\phi(x_i)$ maps the data points from lower dimension to higher dimension.

$$\phi : \mathbf{x} \in \mathbb{R}^{\mathbb{D}} \rightarrow \phi(x) \in \mathbb{F} \subset \mathbb{R}^{\mathbb{D}'} \tag{5}$$

Let $\kappa(x_i, x)$ be a kernel function resulting from the inner product of $\phi(x_i)$ and $\phi(\mathbf{x_j})$, such that

$$\kappa(x_i, x) = \langle \phi(x_i)\phi(x_j) \rangle \tag{6}$$

From **representer theorem** we know that

$$\begin{aligned} F = \beta^T \phi(x) &= \alpha \left\langle \phi(\mathbf{x_i})\phi(\mathbf{x_j}) \right\rangle \\ &= \alpha\kappa(x_i, x_j) \end{aligned}$$

We can now express $p(x;\beta)$ is subspace of input vectors only such that

$$p(\phi;\alpha) = \frac{1}{1 + e^{-\alpha_i \kappa(x_i, x_j)}} \tag{7}$$

and

$$1 - p(\phi;\alpha) = \frac{1}{1 + e^{\alpha_i \kappa(x_i, x_j)}} \tag{8}$$

The logit function is mapped into the kernel space as

$$logit(\frac{p(\phi;\alpha)}{1 - p(\phi;\alpha)}) = \alpha\kappa(x_i, x) \tag{9}$$

Deriving the equation of kernel logistic regression requires the regularized logistic regression, precisely the regularized $l2 - norm$ of the log-likelihood. This is in comparison to the SVM objective function.

$$\begin{aligned} L_\alpha = \sum_{i=1}^{n} y_i log p(x_i) + (1 - y_i) log(1 - p(x_i)) \\ - \frac{\lambda}{2} \alpha^{\mathbf{T}} \kappa(\mathbf{x_i}, \mathbf{x})\alpha \end{aligned}$$

2.3 Learning kernel logistic regression

As mentioned earlier, some of the methods for finding the maximum likelihood estimate include gradient descent (**GD**), iterative reweighted least sqaures (**IRLS**) method. Here we employ the use of IRLS which is based on the Newton-Ralphson algorithm.

- Optimization function:

$$\begin{aligned} L_\alpha = \sum_{i=1}^{n} y_i log p(x_i) + (1 - y_i) log(1- \\ p(x_i)) - \frac{\lambda}{2} \alpha^{\mathbf{T}} \kappa(\mathbf{x_i}, \mathbf{x})\alpha \end{aligned}$$

We can expand the objective function as follows

$$\begin{aligned} L_\alpha = ylog\left(\frac{p}{1-p}\right) + log(1 - p(x_i)) \\ - \frac{\lambda}{2} \alpha^{\mathbf{T}} \kappa(\mathbf{x_i}, \mathbf{x})\alpha \\ = ylog\left(\frac{p}{1-p}\right) + log\left(\frac{1}{1 + e^{\alpha\kappa(x_i, x)}}\right) \\ - \frac{\lambda}{2} \alpha^{\mathbf{T}} \kappa(\mathbf{x_i}, \mathbf{x})\alpha \\ = y\alpha\kappa(x_i, x) - log(1 + e^{\alpha\kappa(x_i, x)}) \\ - \frac{\lambda}{2} \alpha^{\mathbf{T}} \kappa(\mathbf{x_i}, \mathbf{x})\alpha \end{aligned}$$

First order derivative of the log-likelihood

$$\begin{aligned} \nabla_\alpha L = y\kappa(x_i, x) - \frac{\kappa(x_i, x) e^{\alpha\kappa(x_i, x)}}{1 + e^{\alpha\kappa(x_i, x)}} \\ - \lambda\alpha\kappa(x_i, x) \\ = y\kappa(x_i, x) - p\kappa(x_i, x) - \lambda\alpha\kappa(x_i, x) \\ \nabla_\alpha L = \kappa(x_i, x)(y - p) - \lambda\alpha\kappa(x_i, x) \end{aligned}$$

Now we deduce the second derivative from the result of the first. We know from the first that

$$\begin{aligned} \nabla_\alpha L = \kappa(x_i, x)\left(y - \frac{1}{1 + e^{\alpha\kappa(x_i, x)}}\right) \\ - \lambda\alpha\kappa(x_i, x) \end{aligned}$$

$$\begin{aligned} \nabla_\alpha^2 L = -\kappa(x_i, x)^T \left(\frac{e^{\alpha\kappa(x_i, x)}}{(1 + e^{\alpha\kappa(x_i, x)})^2}\right) \\ \times\kappa(x_i, x) - \lambda\kappa(x_i, x) \\ = -\kappa(x_i, x)^T \left(\frac{e^{\alpha\kappa(x_i, x)}}{1 + e^{\alpha\kappa(x_i, x)}} \cdot \frac{1}{1 + e^{\alpha\kappa(x_i, x)}}\right) \\ \times\kappa(x_i, x) - \lambda\kappa(x_i, x) \end{aligned}$$

resulting

$$\nabla^2_\alpha L = -\kappa(x_i,x)^T(p(1-p))\kappa(x_i,x) - \lambda\kappa(x_i,x)$$

The update for α is

$$\alpha_{j+1} = \alpha_j - (\nabla^2_\alpha L)^{-1}\nabla_\alpha L \qquad (10)$$

$$\alpha_{j+1} = \alpha_j + (\kappa(x_i,x)^T W\kappa(x_i,x) + \lambda\kappa(x_i,x))^{-1}(\kappa(x_i,x)^T(y-p) - \lambda\kappa(x_i,x)\alpha)$$

Where W is the diagonal matrix corresponding to $p(1-p)$. For simplification let $\alpha_j = (\kappa(x_i,x)^T W\kappa(x_i,x) + \lambda\kappa(x_i,x))^{-1}(\kappa(x_i,x)^T W\kappa(x_i,x) + \lambda\kappa(x_i,x))\alpha_j$ and $\kappa(x_i,x) = K$.

so that

$$\begin{aligned}\alpha_{j+1} &= (K^TWK+\lambda K)^{-1}(K^TWK+\lambda K)\alpha_j + (K^TWK+\lambda K)^{-1}(K^T(y-p) - \lambda K\alpha)\\ &= (K^TWK+\lambda K)^{-1}((K^TWK+\lambda K)\alpha_j + K^T(y-p) - \lambda K\alpha)\end{aligned}$$

After expanding the above term we have that

$$\begin{aligned}\alpha_{j+1} &= (K^TWK+\lambda K)^{-1}(K^TWK\alpha_j + K^T(y-p))\\ &= (K^TWK+\lambda K)^{-1}K^TW(K\alpha_j + W^{-1}(y-p))\end{aligned}$$

if $z_j = (K\alpha_j + W^{-1}(y-p))$ we can summarize the solution of α_{j+1} with a shorthand equation thus.

$$\alpha_{j+1} = (K^TWK+\lambda K)^{-1}K^TWz_j \qquad (11)$$

z_j is the adjusted response.

- Prediction
  Still using the representer theorem, we compute the posterior probability of a new data point such that

$$y = sign\left(\frac{1}{1+\exp^{-\alpha\kappa(x_i,x)}}\right) \qquad (12)$$

Here, the prediction is dependent only on α and the inner product of the training and test data.

Algorithm 1: KLR using **IRLS**

```
Input  : $\kappa, y, \alpha_j$
Output : $\alpha_{j+1}$
begin
    $c = 0$;
    while $\left|\frac{DEV^c - DEV^{c+1}}{DEV^{c+1}}\right| \geq \epsilon_1$ and $c \leq Max\ IRLS\ Iterations$ do
        for $i \leftarrow 1$ to $N$ do
            $\hat{p} = \frac{1}{1+\exp^{-\alpha\kappa(x_i,x)}}$;
            $v_i = \hat{p}(1-\hat{p})$;
            $z_i = K\alpha^c + \frac{y_i - \hat{p}}{\hat{p}(1-\hat{p})}$;
        end
        $\mathbf{V} = diag(v_1, ..., v_N)$;
        $\hat{\alpha}^{c+1} = (K^TWK + \lambda K)^{-1}K^TWz^c$;
        $c = c + 1$
    end
end
```

3 EXPERIMENTAL RESULT

3.1 Dataset

3.2 Performance analysis

4 CONCLUSION

References

[1] Pearson K. On lineas and Planes of closest fit to systems of points in space. *Philos Mag A*, 6:559–572, 1901.

[2] Zhu J, Hastie T. Kernel logistic regression and import vector machine. *J Comput Graphic Stat*, 14:185–205, 2005.

[3] Saunders C., Gammerman A., Vovk V., Ridge Regression Learning Algorithm in Dual Variables. *ICML 15th Internationl Conforence on Machine Learning.*, 1998.