An alternative clusterwise linear model

Anonymous Author(s)
Affiliation
Address
email

Abstract

TO DO

1 Introduction

TO DO

2 Theoretical model: a cluster-wise linear regression algorithm.

Let's consider a regression problem defined by an observation data matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^\top$ and a target vector $\mathbf{y} = [y_1, \ldots, y_N]^T$, where $\mathbf{x}_i \in \Re^D$ is the i^{th} observation and $y_i \in \Re$ is its corresponding target.

The proposed model considers that the distribution of observations of $\mathbf{x}$ is approximated by a mixture of K Gaussians

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{1}$$

where $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are, respectively, the mean and covariance matrix of the k-th Gaussian.

This model assumes that a set of latent variables $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_N]^T$ exists, where each $\mathbf{z}_i = \{z_{i,k}\}_{k=1}^K$ is modeled such that only k-th entry of these vectors equals 1 and the rest is zero, indicating that $\mathbf{x}_i$ has been generated by k-th Gaussian mixture component. The priors of these variables are defined as:

$$p(z_{i,k} = 1) = \pi_k. \tag{2}$$

where $0 \leq \pi_k \leq 1$ and $\sum_1^K \pi_k = 1$.

This means that, given $z_k = 1$, the generation of y only depends of the k-th regressor, given by the following linear model:

$$y = \mathbf{w}_k^\top \mathbf{x} + \epsilon_k, \tag{3}$$

where $\mathbf{w}_k$ are the linear regression weights of the k_{th} component, including the bias term; $\mathbf{x}$ is considered to be extended with a constant term of value 1 to account for the bias term; and ϵ_k is assumed to be Gaussian noise with zero mean and variance β_k^{-1}.

Thus, given the observation $\mathbf{x}$ and the model parameters, and with $z_k = 1$, the probability distribution for the target variable y becomes:

$$p(y|z_k = 1, \mathbf{X}, \boldsymbol{\theta}) = \mathcal{N}(y|\mathbf{w}_k^\top \mathbf{x}, \beta_k^{-1}) \tag{4}$$

The mixture distribution for the target variables can therefore be stated as follows:

$$p(y|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(y|\mathbf{w}_k^\top \mathbf{x}, \beta_k^{-1}) \tag{5}$$

where $\boldsymbol{\theta}$ includes all model parameters: $\boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{w}$ and $\boldsymbol{\beta}$.

Submitted to 31st Conference on Neural Information Processing Systems (NIPS 2017). Do not distribute.

2.1 Probabilistic representation

From a probabilistic standpoint, this model can be represented graphically as in Figure 1.

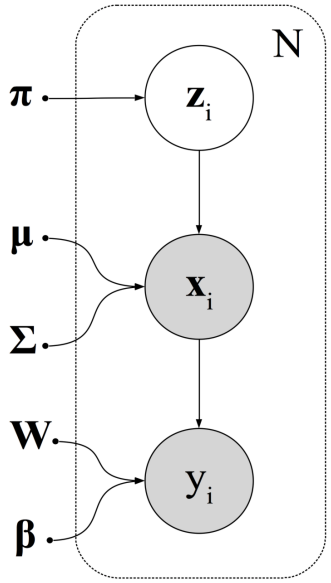

Figure 1: Graphical representation for the model in which the regression weights are treated as parameters.

This graph leads us to the following complete-data likelihood function:

$$p(\mathbf{Z}, \mathbf{X}, \mathbf{y}, \mathbf{w}|\boldsymbol{\theta}) = p(\mathbf{Z}|\boldsymbol{\theta})p(\mathbf{X}|\mathbf{Z}, \boldsymbol{\theta})p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) \tag{6}$$

where:

$$p(\mathbf{Z}|\boldsymbol{\theta}) = \prod_{i=1}^{N}\prod_{k=1}^{K} \pi_k^{z_{i,k}} \tag{7}$$

$$p(\mathbf{X}|\mathbf{Z}, \boldsymbol{\theta}) = \prod_{i=1}^{N}\prod_{k=1}^{K} \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_{i,k}} \tag{8}$$

$$p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) = \prod_{i=1}^{N}\prod_{k=1}^{K} \mathcal{N}(y_i|\mathbf{w}_k^\top \mathbf{x}_i, \beta_k^{-1})^{z_{i,k}} \tag{9}$$

Therefore:

$$p(\mathbf{Z}, \mathbf{X}, \mathbf{y}, \mathbf{w}|\boldsymbol{\theta}) = \prod_{i=1}^{N}\prod_{k=1}^{K} \left[\pi_k \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\mathcal{N}(y_i|\mathbf{w}_k^\top \mathbf{x}_i, \beta_k^{-1})\right]^{z_{i,k}} \tag{10}$$

From here we can now compute the complete-data log likelihood:

$$\begin{aligned} &\ln p(\mathbf{X}, \mathbf{y}, \mathbf{Z}|\boldsymbol{\theta}) = \\ &= \sum_{i=1}^{N}\sum_{k=1}^{K} z_{i,k}\left[\ln \pi_k + \ln \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) + \ln \mathcal{N}(y_i|\mathbf{w}_k^\top \mathbf{x}_i, \beta_k^{-1})\right] \end{aligned} \tag{11}$$

2.2 E-Step: updating the posterior expectations for the latent variables

To find the optimum value of the parameters $\boldsymbol{\theta}$, we can apply the EM algorithm. So, we will start with the E-step by computing the posterior value of the latent variables, known as responsabilities:

$$\begin{aligned}\gamma(z_{i,k}) = p(z_{i,k}|\mathbf{x}_i, \mathbf{y}_i, \boldsymbol{\theta}) = \frac{p(\mathbf{x}_i, \mathbf{y}_i, z_{i,k}|\boldsymbol{\theta})}{p(\mathbf{x}_i, \mathbf{y}_i|\boldsymbol{\theta})} = \\ = \frac{\pi_k \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\mathcal{N}(y_i|\mathbf{w}_k^\top \mathbf{x}_i, \boldsymbol{\beta}_k^{-1})}{\sum_{k'=1}^{K} \pi_{k'} \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'})\mathcal{N}(y_i|\mathbf{w}_{k'}^\top \mathbf{x}_i, \boldsymbol{\beta}_{k'}^{-1})}\end{aligned} \tag{12}$$

2.3 M-Step: updating the model parameters

We can now update the parameter values in the M-step by computing the maximum value of the expected value of the complete log-likelihood under the posterior of the latent variables, i.e.,

$$\boldsymbol{\theta}^{\text{new}} = \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{Z}} \left\{ \ln p(\mathbf{Z}, \mathbf{X}, \mathbf{y}|\boldsymbol{\theta}) \right\} \tag{13}$$

where

$$\begin{aligned}&\mathbb{E}_{\mathbf{Z}} \left\{ \ln p(\mathbf{Z}, \mathbf{X}, \mathbf{y}|\boldsymbol{\theta}) \right\} = \\ &= \sum_{i=1}^{N} \sum_{k=1}^{K} \gamma(z_{i,k}) \left[\ln \pi_k + \ln \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) + \ln \mathcal{N}(y_i|\mathbf{w}_k^\top \mathbf{x}_i, \boldsymbol{\beta}_k^{-1}) | \right] |\end{aligned} \tag{14}$$

Then, computing the derivatives of (14) with respect to the different parameters, we can update the model parameters:

$$\pi_k = \frac{1}{N} \sum_{i=1}^{N} \gamma(z_{n,k}) \tag{15}$$

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{i=1}^{N} \gamma(z_{i,k}) \mathbf{x}_i \tag{16}$$

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{i=1}^{N} \gamma(z_{i,k}) (\mathbf{x}_i \mathbf{x}_i^\top - \boldsymbol{\mu}_k \boldsymbol{\mu}_k^\top) \tag{17}$$

$$\mathbf{w}_k = (\mathbf{X}^\top \boldsymbol{\Gamma}_k \mathbf{X})^{-1} \mathbf{X}^\top \boldsymbol{\Gamma}_k \mathbf{y} \tag{18}$$

$$\boldsymbol{\beta}_k^{-1} = \frac{1}{N \pi_k} \sum_{i=1}^{N} \gamma(z_{i,k}) \left(y_i - \mathbf{w}_k^\top \mathbf{x}_i \right)^2 \tag{19}$$

where $N_k = \sum_{\forall i} \gamma(z_{i,k})$ and $\boldsymbol{\Gamma}_k$ is defined as:

$$\boldsymbol{\Gamma}_k = diag\big(\{\gamma(z_{1,k}), \gamma(z_{2,k}), \ldots, \gamma(z_{N,k})\}\big) \tag{20}$$

Note that, contrary to the standard GMM model and the expert mixture model proposed in [1], in this model the responsibilities depend on both $p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $p(\mathbf{y}|\mathbf{w}^\top \mathbf{X}, \beta^{-1})$.

2.4 Predictive function

The goal of the prediction function is to estimate the output, $f^\star$, given a new test observation $\mathbf{x}^\star$. In this case, the output of the k-th regressor is given by $f_k^\star = \mathbf{w}_k^\top \mathbf{x}^\star$. Therefore, the probability distribution of $f^\star$ given the test observation and the training data is given by:

$$p(f^\star|x^\star, \mathbf{X}, \mathbf{y}, \boldsymbol{\theta}) = \sum_{k=1}^{K} \gamma_k^\star p(f_k^\star|z_k^\star = 1, \mathbf{x}^\star, \mathbf{X}, \mathbf{y}, \boldsymbol{\theta}) \tag{21}$$

where we can define the responsibility $\gamma_k^\star$ as:

$$\begin{aligned} \gamma_k^\star &= p(z_k^\star = 1|\mathbf{x}^\star, \mathbf{X}, \mathbf{y}, \boldsymbol{\theta}) = p(z_k^\star = 1|\mathbf{x}^\star, \boldsymbol{\theta}) = \\ &= \frac{p(z_k^\star = 1, \mathbf{x}^\star, \boldsymbol{\theta})}{\sum_{k'=1}^{K} p(z_{k'}^\star = 1, \mathbf{x}^\star, \boldsymbol{\theta})} = \frac{\pi_k p(\mathbf{x}^\star|\mu_k, \Sigma_k)}{\sum_{k'=1}^{K} \pi_{k'} p(\mathbf{x}^\star|\mu_{k'}, \Sigma_{k'})} \end{aligned} \tag{22}$$

and

$$p(f_k^\star|z_k^\star = 1, \mathbf{x}^\star, \mathbf{X}, \mathbf{y}, \boldsymbol{\theta}) = p(f_k^\star|z_k^\star = 1, \mathbf{w}_k) = \mathcal{N}(f_k^\star|\mathbf{w}_k^\top \mathbf{x}^\star, 0) = \delta(f_k^\star - \mathbf{w}_k^\top \mathbf{x}^\star) \tag{23}$$

Therefore:

$$p(f^\star|\mathbf{x}^\star, \mathbf{X}, \mathbf{y}, \boldsymbol{\theta}) = \sum_{k=1}^{K} \gamma_k^\star \delta(f_k^\star - \mathbf{w}_k^\top \mathbf{x}^\star) \tag{24}$$

We can now calculate the predictive value of $f^\star$ as the expected value of $p(f^\star|\mathbf{x}^\star, \mathbf{X}, \mathbf{y}, \boldsymbol{\theta})$:

$$\mathbb{E}\{f^\star|\mathbf{x}^\star, \mathbf{X}, \mathbf{y}, \boldsymbol{\theta}\} = \sum_{k=1}^{K} \gamma_k^\star \mathbf{w}_k^\top \mathbf{x}^\star \tag{25}$$

2.5 Regularization term

An L^2 regularization on the regression weights can be included in the model with barely any modifications to the algorithm. As we will see in the following sections, this result is akin to assuming a Gaussian prior on the regression weights, where $p(\mathbf{w}|\eta) = \mathcal{N}(\mathbf{w}|0, \eta^{-1}\mathbf{I})$. It is important to note, however, that we are not yet giving the regression weights a full probabilistic treatment. For now, the free parameter η acts simply as a regularization constant, and must be cross-validated to determine its optimal value without over-fitting the training data.

The introduction of this regularization term appears in the model in equation 18, which now takes the following form:

$$\mathbf{w}_k = (\mathbf{X}^\top \boldsymbol{\Gamma}_k \mathbf{X} \beta_k + \eta \mathbf{I})^{-1} \mathbf{X}^\top \boldsymbol{\Gamma}_k \beta_k \mathbf{y} \tag{26}$$

2.6 Performance example

Figures 2.6 - 8 showcase the performance of the regularized version of the algorithm described in Section 2, pitted against two baseline methods, when applied to an artificially generated toy dataset:

- A simple but widely used strategy that first performs K-means clustering on the input data and then trains K separate linear regressors.
- A Mixture of Linear Regressors as described in [**?**].

For clustering purposes on the training set, the clusterwise linear model offers three different interpretations for each training sample, $\mathbf{x}_i$:

- The full likelihood of each cluster for a given training sample, reflected in the responsibilities, $\gamma(z_{i,k})$.
- The component of the responsibilities that corresponds to the input Gaussian mixture model, $\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$.
- The component of the responsibilities that corresponds to the output linear regression model, $\mathcal{N}(y_i|\mathbf{w}_k^\top \mathbf{x}_i, \boldsymbol{\beta}_k^{-1})$.

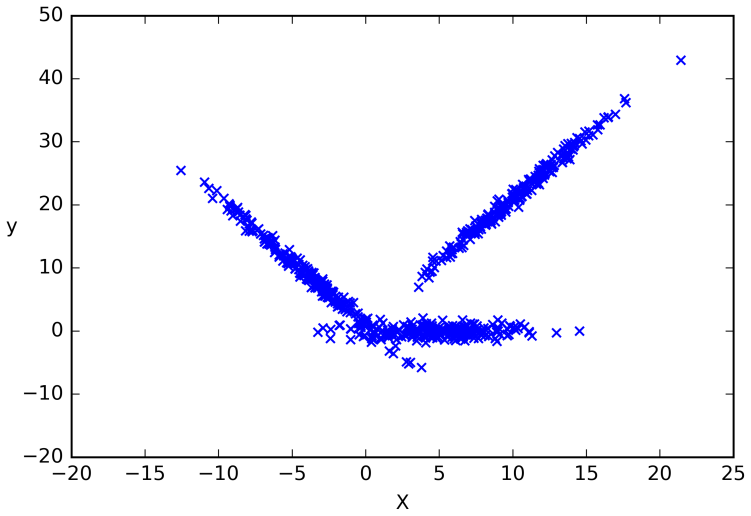

Figure 2: Example dataset.

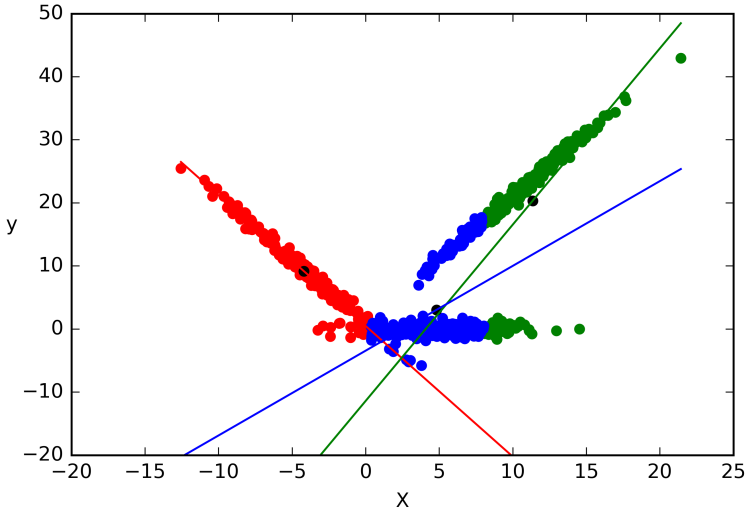

Figure 3: Training a K-means + linear regression model.

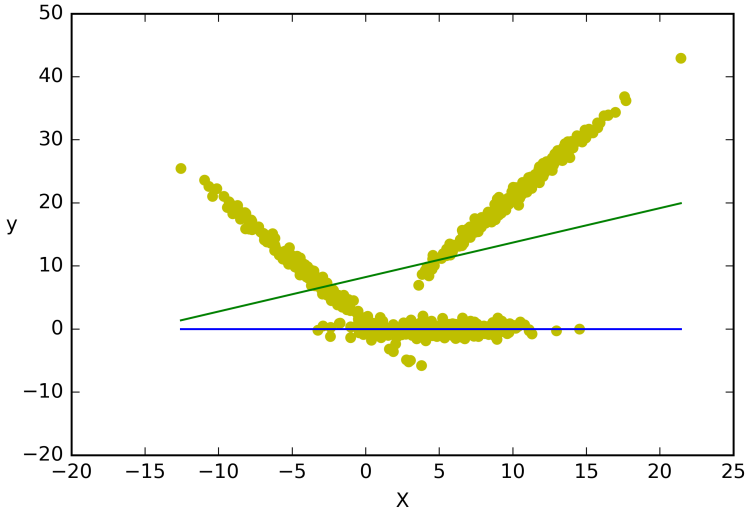

Figure 4: Training a linear regression mixture model.

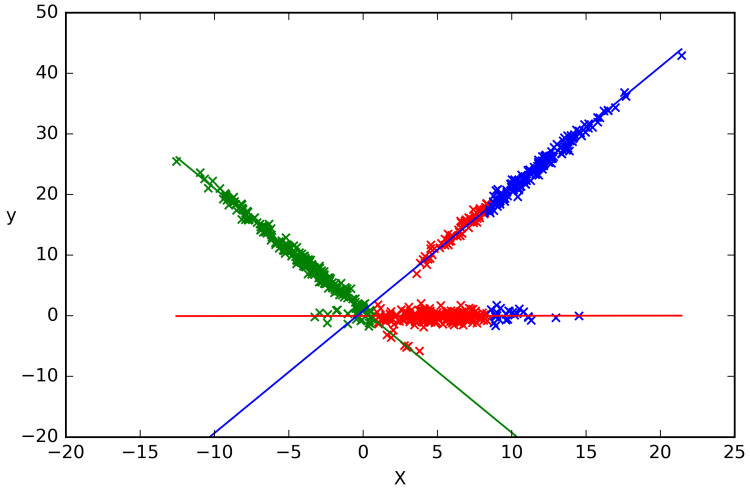

Figure 5: Training a clusterwise linear regression model. Clustering of the training set according to the responsibilities.

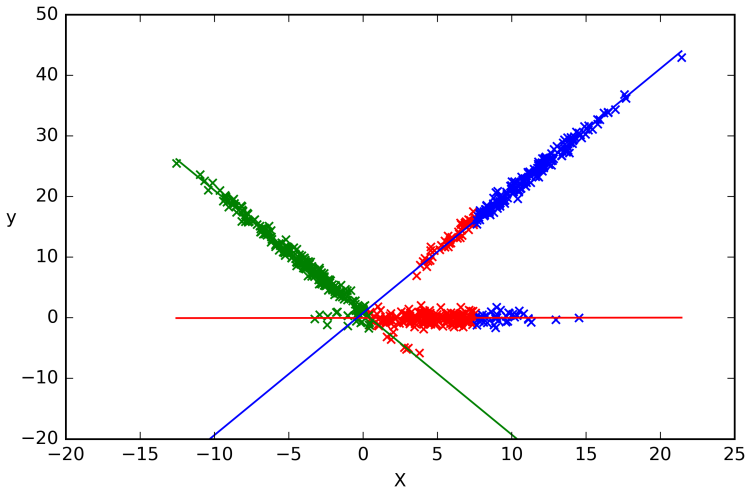

Figure 6: Training a clusterwise linear regression model. Clustering of the training set according to the input Gaussian mixture model.

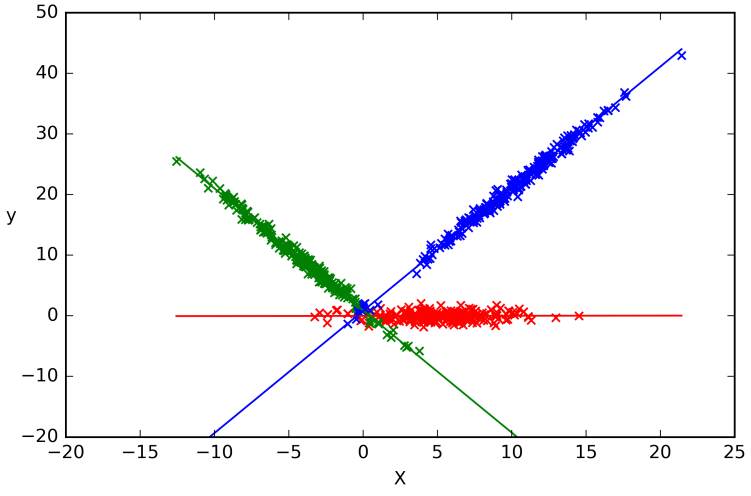

Figure 7: Training a clusterwise linear regression model. Clustering of the training set according to the output linear regression model.

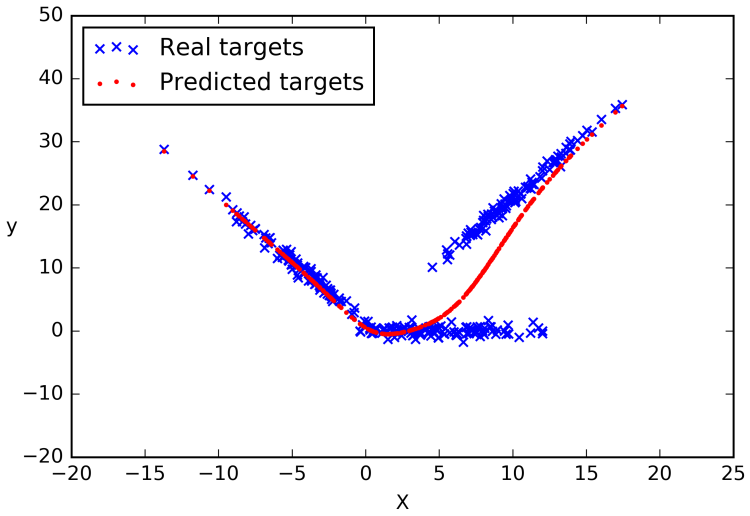

Figure 8: Clusterwise linear regression prediction on a test set.

References

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.