

## なぜ Local Regression ?

なぜ、Inverse PM で Local regression を使おうと思うのか？

俺の直感では、Inverse PM では、high-level indicator と PM parameter の関係は一般的に非線形であり、その関係を quadratic だとか、polynomial とかで表現できるとは限らない。そもそも、general なアプローチを提案したいので、quadratic だとか polynomial のような前提はしたくない。

Artificial Neural Network や Deep Learning は、そのような前提なしに、input と output の関係を学習できる。しかし、関係が非常に複雑な場合は、新しい値に対する interpolation、extrapolation は、やはり難しい。※実際に deep learning を実装して試す必要アリ。

俺の直感では、local regression が最も最適な solution ではないかと思う。非常にたくさんの sample を用意し、locally linear (もしくは、locally quadratic) を前提として output の推定を行う。非常に dense な sample があれば、local regression は一般的に効果的みたい。しかも、dense な sample があれば、どのような関係も表現できる。

つまり、感覚的には、deep learning と同様に表現力があり、しかも、deep learning よりも robust というイメージかな。

## Local Regression

データ  $X = \{x_1, x_2, \dots, x_N\}$ 、そのアウトプット  $Y = \{y_1, y_2, \dots, y_N\}$  が与えられた時、データ  $x$  のアウトプット  $\mu(x)$  を予測したい。 **※アウトプットはスカラー値とする！**

Local linear regression の場合、各アウトプットを以下のように線形で表現する。

$$\mu(x_i) \approx a_0 + a_1(x_i - x) \quad (1)$$

式(1)を、多次元データに一般化すると、

$$\mu(x_i) \approx [(x_i - x)^T \quad 1] \mathbf{a} \quad (2)$$

この時、以下の誤差を最小化するような  $a_0$ 、 $a_1$  を求める。

$$\sum_{i=1}^N W_i [y_i - \mu(x_i)]^2 \quad (3)$$

$W_i$  は、距離  $\|x_i - x\|$  に基づいた重みだ。以下のようなガウス分布などが良く使われる。

$$W_i = \exp \left[ -\frac{\|x_i - x\|^2}{2\sigma^2} \right]$$

他にも、squared や cubic など良く使われるみたい。

$$W_i = (1 - \|x_i - x\|^3)^3$$

結局、上の式は、 $x$ が $x_i$ に近ければ近いほど、誤差の重みが大きくなる。つまり、これを最小化することで、 $x$ が $x_i$ に近ければ、アウトプット $\mu(x)$ は $y_i$ に近づくわけだ。すごくリーズナブルなアイデアだね。

この時、予測される値は、式(1)に $x_i = x$ を代入して、 $\mu(x) = a_0$ となる。

式(3)は、

$$W_1(y_1 - \mu_1)^2 + W_2(y_2 - \mu_2)^2 + \dots + W_N(y_N - \mu_N)^2$$

と表せる。これって、例の「二次形式」を使うと、以下のように行列で表現できる。

$$[y_1 - \mu_1 \quad y_2 - \mu_2 \quad \dots \quad y_N - \mu_N] \begin{bmatrix} W_1 & & & \\ & W_2 & & \\ & & \ddots & \\ & & & W_N \end{bmatrix} \begin{bmatrix} y_1 - \mu_1 \\ y_2 - \mu_2 \\ \vdots \\ y_N - \mu_N \end{bmatrix}$$

つまり、

$$(\mathbf{y} - \boldsymbol{\mu})^T W (\mathbf{y} - \boldsymbol{\mu}) \quad (4)$$

ただし、

$$\mu_i = [(\mathbf{x}_i - \mathbf{x})^T \quad 1] \mathbf{a}$$

よって、

$$\boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_N \end{bmatrix} = \begin{bmatrix} [(\mathbf{x}_1 - \mathbf{x})^T \quad 1] \mathbf{a} \\ \vdots \\ [(\mathbf{x}_N - \mathbf{x})^T \quad 1] \mathbf{a} \end{bmatrix} = X \mathbf{a}$$

ただし、

$$X = \begin{bmatrix} (\mathbf{x}_1 - \mathbf{x})^T & 1 \\ \vdots & \vdots \\ (\mathbf{x}_N - \mathbf{x})^T & 1 \end{bmatrix} \quad (5)$$

従って、式(4)は、

$$(\mathbf{y} - X\mathbf{a})^T W (\mathbf{y} - X\mathbf{a}) \quad (6)$$

つまり、weighted least squares は、式(6)の形で表現できる。

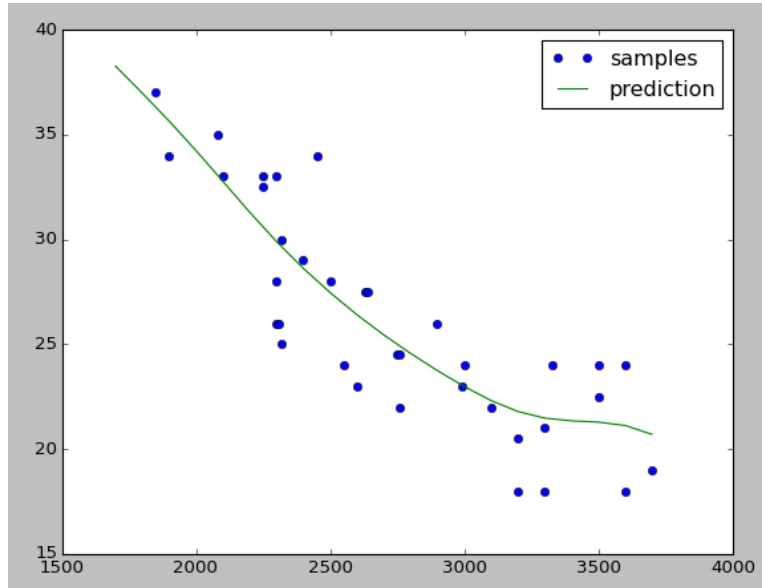
この解は、以下のようになるらしい（なんで??）

$$\mathbf{a} = \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = (X^T W X)^{-1} X^T W Y \quad (7)$$

■まとめ：

ある点 $x$ が与えられた時、重み行列 $W$ を計算し、式(5)で行列 $X$ を作成し、式(7)を使って $\mathbf{a}$ を求めたら、その最後の要素（ $a_1$ ）が、予想されるアウトプットだ。

うまくいけば、こんな結果が得られるはず。



**Figure 1.** ガウスカーネルを使用した場合の、local regression の結果。  $\sigma = 250$  を使用した。

## 多次元アウトプット

ここまでの議論では、アウトプットはスカラー値と見なしてきた。つまり、

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

多次元アウトプットの場合は、式(7)で、 $Y$ の列数が1から複数に増えるだけ。しかも、 $Y$ は、式(7)の最後に出てくるだけなので、 $Y$ の列数が増えても、そのままOKってことだ！つまり、

$$A = (X^T W X)^{-1} X^T W Y$$

で $A$ を求めたら、 $A$ の最後の行が答えだ！