

PRML 本の 6 章に詳細あり。式番号は、PRML 本に合わせている。

## 1 Gaussian process

とりあえず難しい数式の話は後回しね。データポイント  $x_1, \dots, x_N$  に対して、真の値を  $y = y_1, \dots, y_N$  とすると、各値は以下の確率で与えられるとする。

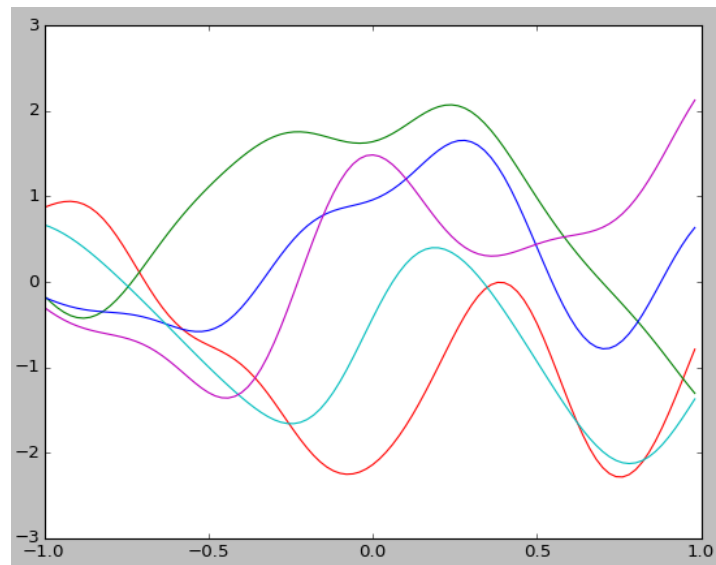
$$p\left(\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}\right) \sim N\left(\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, Cov\right)$$

ただし、

$$Cov = \begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_N) \\ \vdots & \ddots & \vdots \\ k(x_N, x_1) & \cdots & k(x_N, x_N) \end{bmatrix}$$

※なんで、平均が 0 なの？ってところがすごい混乱するんだけど、そういうものと考えられないみたい。逆に言うと、元の値が偏っていたりする場合は、平均が 0 になるよう調整する必要がある。

下図に、いくつかの曲線が表示されている。1つの曲線が、1つの  $y$  に相当する。ガウス分布に従った乱数なので、平均 0 から少し離れたりしているわけだよね。



ガウス過程を使った regression では、与えられた観測値に対して、最も近い曲線を当てはめるというイメージだ。※ハッキリ言って、Gaussian process そのものは良く分らんけど、次章の regression さえ理解しておけば問題ないと思う。

## 2 Gaussian process for regression

データポイント $x_1, \dots, x_N$ に対して、真の値を $y_1, \dots, y_N$ 、観測値 $\mathbf{t} = t_1, \dots, t_N$ とする。観測値は、真の値に対してノイズが加味されると考える。つまり、

$$t_n = y_n + \epsilon \quad (6.57)$$

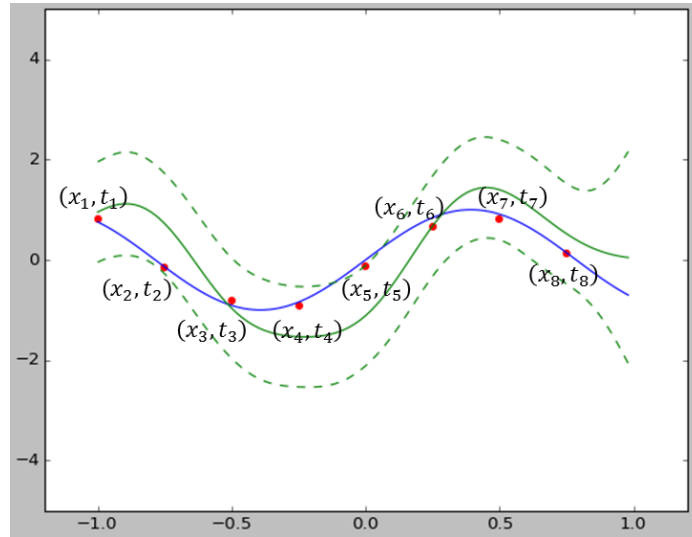
ガウスノイズと仮定し、分散 $\beta^{-1}$ とすると、

$$p(t_n|y_n) = N(t_n|y_n, \beta^{-1}) \quad (6.58)$$

また、データポイントの共分散を以下のように定義する。

$$k(x_n, x_m) = \theta_0 \exp \left\{ -\frac{\theta_1}{2} \|x_n - x_m\|^2 \right\} + \theta_2 + \theta_3 x_n^T x_m \quad (6.63)$$

なんでこんな式を共分散として使用するのか？よく使われる式みたいなので、とりあえず使っとけて感じ。難しい数式はここまで。後は実際にどう使うかの話だ。



この時、任意の点 $x_{N+1}$ の値 $\hat{y}_{N+1}$ は、次のようにして推定される。

$$\hat{y}_{N+1} = \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{t} \quad (6.66)$$

ただし、

$$\mathbf{k} = \begin{bmatrix} k(x_1, x_{N+1}) \\ k(x_2, x_{N+1}) \\ k(x_3, x_{N+1}) \\ \vdots \\ k(x_N, x_{N+1}) \end{bmatrix}$$

また、

$$C_N = \begin{bmatrix} k(x_1, x_1) + \beta^{-1} & k(x_1, x_2) & \cdots & k(x_1, x_N) \\ k(x_2, x_1) & k(x_2, x_2) + \beta^{-1} & \cdots & k(x_2, x_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_N, x_1) & k(x_N, x_2) & \cdots & k(x_N, x_N) + \beta^{-1} \end{bmatrix}$$

※式(6.66)は、こう考えると理解できる。点 $x_{N+1}$ の値は、既に観測したデータ群 $\mathbf{t}$ の加重平均的な感じで計算され、その重みは、点 $x_{N+1}$ に近い点の観測値をより大きい重みにする。すごく自然な考え方だね。

推定された値の精度は、以下のように偏差を計算して推定できる。

$$\sigma^2 = c - k^T C_N^{-1} k \quad (6.67)$$

ただし、

$$c = k(x_{N+1}, x_{N+1}) + \beta^{-1}$$

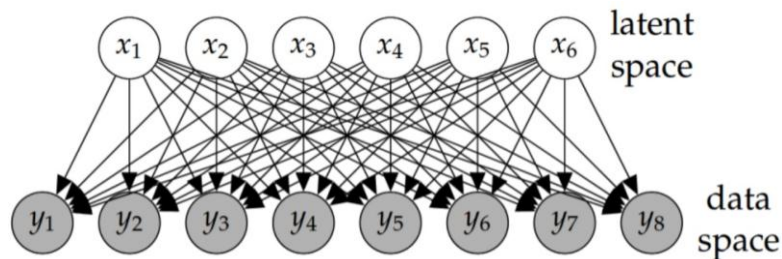
とりあえず、以上の式を使うことで、regression は行えるのだ。とりあえず、「使う」分にはこれで困らないよね。

### 3 Gaussian Process Latent Variable Model (GP-LVM)

そもそも、Latent variable model って、Machine Learning の授業でもやったけど、なんらかの隠れ変数 $x$ を想定し、実際に観測される値 $y$ は、その隠れ変数のマッピングだという考え方だね。

$$y = g(x) + \epsilon$$

Graphical model で表現すると、以下のイメージだ。



では、どうやって隠れ変数を抽出するのか？ 1つの方法はPCAだ。しかし、PCAは線形変換にしか対応しない。これに対して、GP-LVMは線形である必要がない。

欠点としては、non-convex 関数に対するパラメータ最適化を行うため、必ずしも global optimum を得られるとは限らない。また、大規模データに対して、計算コストが大きくなり、実用的でなくなる。