Gaussian Processes (Continued)

Scribe: Mishari Alarfaj

---

1. Kernel Function

- The kernel function can sometimes be called the 'Covariance Function'
- In this lecture, we will use $k(x_1, x_2) = e^{\frac{distance(x_1, x_2)}{l^2}}$, where l is the 'rate of decay', or in other words, how much x_1 and x_2 influence each other.
- If l is too small, $\hat{f}$ will appear to look like a series of delta functions.
- If l is too large, $\hat{f}$ will be over-fitted.

2. Deriving the algorithm

- We have:
  - $k(x_1, x_2)$, the kernel function
  - $x_1, x_2 \ldots x_{10}$, the feature vectors
  - $f(x_1), f(x_2) \ldots f(x_{10})$, the result vector
- We now derive $p(f|x) = \frac{1}{z} e^{(\vec{f})^T K^{-1} (\vec{f})}$
- Where:
  - f = The vector of results
  - $K = \begin{bmatrix} k(x_1, x_1) & \ldots & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \ldots & k(x_n, x_n) \end{bmatrix}$
  - $z = \sqrt{(2\pi)^n * det(K)}$
- Keep in mind that $\mu = 0$ for now

Aside: Why is this useful?

- Consider the kernel function above with values of l = 1.0 and l = 0.5
  - $k_1(x_1, x_2) = e^{\frac{-distance(x_1, x_2)^2}{1.0}}$ $\quad k_2(x_1, x_2) = e^{\frac{-distance(x_1, x_2)^2}{0.5}}$
  - Which do we choose?
- We want l by solving: $\max_l \log(p(f|l)) + \log(p(l))$
- Which is the same as: $\min_l \frac{1}{2} \vec{f}^T K^{-1} \vec{f} + \log|K| + C$
  - The first term is the error in prediction, since it is not 0-mean
  - The second term is large if there is little overlap, and small if there is a lot of overlap
- k_1, k_2 above are isotropic kernels, and are uniform over the space

Aside 2: 0-Mean does not matter... Why?

- Let mean $\mu = 0$
  $f \sim GP(\mu(x), k(x, x'))$
- $$f' = f(x) - \mu(x)$$
  $$f' = GP(0, k(x, x'))$$
- Also, μ is independent of covariance since it is only a bias.

3. Back to derivation

- Up until now, we have not considered noise.
- We attach the next result, f^* to the result vector and calculate its parameters as before:
  $$\begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \\ f^* \end{bmatrix} = N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} K_{xx} & K_{xx^*} \\ K_{x^*x} & K_{x^*x^*} \end{bmatrix}\right)$$
- When we condition, we get: $f^* | \vec{f} = N\left(K_{x^*x} * K_{xx}^{-1} * \vec{f}, ...\right)$
  - Where we set $\alpha = K_{xx}^{-1} * \vec{f}$
  - posterior: $\mu(x|\vec{f}) = \sum_i \alpha_i * K(x_i, x)$
- Another way of looking at it is: $\beta = K_{x^*x} * K_{xx}^{-1}$
  - posterior: $\mu(x^*|\vec{f}) = \sum_i \beta(x^*) * \vec{f}$
  - Where β is the 'equivalent kernel'
- However, we never see f's, we see 'y'
  - $y_i = f_i + \epsilon$ where $\epsilon \sim N(0, \sigma^2)$
  - So we get: $\vec{y} = \vec{f} + \epsilon$ where $\epsilon \sim N(0, \sigma^2 \mathrm{I})$
  - Therefore: $p(\vec{y}) = N(0, K + \sigma^2 \mathrm{I})$ since $Var(X+Y) = Var(X) + Var(Y)$ if X and Y are independent of each other.
  - Finally: $p(y^* | y_{1,} y_{2,} ... y_n) = N\left(K_{x,x^*} * [K + \sigma^2 \mathrm{I}]^{-1} * \vec{y}, ...\right)$

4. Computational complexity:

- Learning
  - $BLR = O(d^3)$
  - $GP = O(n^3)$
- Prediction (per point)
  - $BLR = O(d)$
  - $GP = O(n * \hat{d})$ where $\hat{d}$ is dependent on the kernel used
  -

5.Conclusion

- To Predict:
  - $\alpha = K_{x,x^*} * [K + \sigma^2 \mathrm{I}]^{-1}$
  - $\mu = \sum_i \alpha_i k(x_i, x^*)$
- To prevent outfitting:
  - More noise = less correlation
  - Cross-validate with a certain set of data points.