Lecture 9 - Experimental Design with Gaussian Processes

OBJECTIVE: In this lecture we introduce gaussian processes (GPs) for regression. These nonparametric models allow us to generate nonlinear predictions. Yet, the models are very tractable and permit the application of experimental design ideas introduced on the previous lecture. We will also study the problem of classification and construct ingenious kernels using graphs. By relaxing the classification problem with relational inputs, we will end up with a simple GP classifier that is amenable to active learning.

◇ GAUSSIAN PROCESSES

A Gaussian process, denoted $\mathbf{z}(\cdot) \sim GP(m(\cdot), K(\cdot,\cdot))$, is an infinite random process indexed by a variable $\mathbf{x}$ such that any realization $\mathbf{z}(\mathbf{x}_i)$ is Gaussian with mean $m(\mathbf{x}_i)$ and covari-

ance (symmetric positive definite kernel) $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$.

Since the mean function is typically unknown, one can construct a GP for prediction in two stages:

$$\begin{aligned} \widehat{\mathbf{y}}(\mathbf{x}) &= \sum_{j=1}^{d} f_j(\mathbf{x})\theta_j + \mathbf{z}(\mathbf{x}) \\ \mathbf{z}(\cdot) &\sim GP(0, \sigma^2\mathbf{K}) \end{aligned}$$

On observing data $\{\mathbf{x}_{1:n}, \mathbf{y}_{1:n}\}$ and a test point $\mathbf{x}_{n+1}$ (or more test points), the vector of predictions on the test and training data is jointly Gaussian:

$$\begin{bmatrix} \widehat{y}_{n+1} \\ \widehat{\mathbf{y}}_{1:n} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \widehat{\mathbf{f}} \\ \mathbf{F} \end{bmatrix} \boldsymbol{\theta}, \sigma^2 \begin{bmatrix} k & \mathbf{k}^T \\ \mathbf{k} & \mathbf{K} \end{bmatrix} \right)$$

where $k = k(\mathbf{x}_{n+1}, \mathbf{x}_{n+1})$,

$$\begin{aligned}
\widehat{\mathbf{k}}^T &= \begin{bmatrix} k(\mathbf{x}_{n+1}, \mathbf{x}_1) & \cdots & k(\mathbf{x}_{n+1}, \mathbf{x}_n) \end{bmatrix} \\
\mathbf{K} &= \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \cdots & k(\mathbf{x}_1, \mathbf{x}_n) \\ \vdots & \vdots & \vdots \\ k(\mathbf{x}_n, \mathbf{x}_1) & \cdots & k(\mathbf{x}_n, \mathbf{x}_n) \end{bmatrix} \\
\widehat{\mathbf{f}} &= \begin{bmatrix} f_1(\mathbf{x}_{n+1}) & \cdots & f_d(\mathbf{x}_{n+1}) \end{bmatrix} \\
\mathbf{F} &= \begin{bmatrix} f_1(\mathbf{x}_1) & \cdots & f_d(\mathbf{x}_1) \\ \vdots & \vdots & \vdots \\ f_1(\mathbf{x}_n) & \cdots & f_d(\mathbf{x}_n) \end{bmatrix}
\end{aligned}$$

◇ GP WITH KNOWN MEAN FUNCTION

If $\boldsymbol{\theta}$ is known, then one can use simple conditioning arguments for multivariate Gaussian distributions to obtain the conditional distribution of $\widehat{\mathbf{y}}_{n+1}$ given $\widehat{\mathbf{y}}_{1:n}$. This distribution

is Gaussian with mean and covariance:

$$\begin{aligned}
\mathbb{E}(\widehat{\mathbf{y}}_{n+1}|\widehat{\mathbf{y}}_{1:n}) &= \widehat{\mathbf{f}}\boldsymbol{\theta} + \mathbf{k}^T\mathbf{K}^{-1}(\widehat{\mathbf{y}}_{1:n} - \mathbf{F}\boldsymbol{\theta}) \\
cov(\widehat{\mathbf{y}}_{n+1}|\widehat{\mathbf{y}}_{1:n}) &= k - \mathbf{k}^T\mathbf{K}^{-1}\mathbf{k}
\end{aligned}$$

◇ GP WITH UNKNOWN MEAN FUNCTION

As in the linear model, we can assign a Gaussian prior $\boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{\theta}_0, \sigma^2 R^{-1})$. This prior results in a Gaussian posterior with mean and covariance:

$$\begin{aligned}
\boldsymbol{\mu} = \mathbb{E}(\boldsymbol{\theta}|\mathbf{F}, \mathbf{y}) &= (\mathbf{F}^T\mathbf{K}^{-1}\mathbf{F} + R)^{-1}(\mathbf{F}^T\mathbf{K}^{-1}\mathbf{y} + R\boldsymbol{\theta}_0) \\
cov(\boldsymbol{\theta}|\mathbf{F}, \mathbf{y}) &= (\mathbf{F}^T\mathbf{K}^{-1}\mathbf{F} + R)^{-1}\sigma^2 = \Sigma\sigma^2
\end{aligned}$$

Under this posterior distribution, the predictive distribution of the Gaussian process becomes:

$$\begin{aligned}
\mathbb{E}(\widehat{\mathbf{y}}_{n+1}|\widehat{\mathbf{y}}_{1:n}) &= \widehat{\mathbf{f}}\boldsymbol{\mu} + \mathbf{k}^T\mathbf{K}^{-1}(\widehat{\mathbf{y}}_{1:n} - \mathbf{F}\boldsymbol{\mu}) \\
cov(\widehat{\mathbf{y}}_{n+1}|\widehat{\mathbf{y}}_{1:n}) &= \sigma^2 \left\{ k - \mathbf{k}^T\mathbf{K}^{-1}\mathbf{k} + (\widehat{\mathbf{f}} - \mathbf{F}^T\mathbf{K}^{-1}\mathbf{k})^T\Sigma(\widehat{\mathbf{f}} - \mathbf{F}^T\mathbf{K}^{-1}\mathbf{k}) \right\}
\end{aligned}$$

◇ EXPERIMENTAL DESIGN WITH GAUSSIAN PROCESSES

As in the linear case, we should choose an experiment (point x) so as to minimize the variance of the prediction. That is,

$$u^\star(\mathbf{e}) = \min_{\mathbf{e}} cov(\widehat{\mathbf{y}}_{n+1}|\widehat{\mathbf{y}}_{1:n})$$

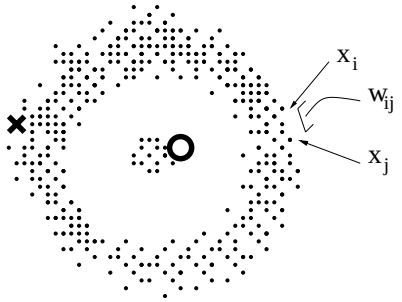
That is, we should choose to experiment in parts of the space where there is high uncertainty.

In this case the covariance is really a univariate variance as we only have a single test point. In general we can try to minimize either the determinant or trace (better) of the predictive covariance for a block of test points.

Note that the design with unknown mean function depends on the prediction $\widehat{\mathbf{f}}$.

◇ KERNELS ON GRAPHS AND SEMI-SUPERVISED LEARNING

We are given N feature vectors $\mathbf{x} \in \mathbb{R}^d$ as shown for $d = 2$ in Fig. 2. Some of the points have labels. In the figure, two points have the labels $y^l = 1$ and $y^l = 0$. The rest of the points have unknown labels y^u. The goal is to compute the unknown labels.



Input data: Two points $(\times)$ *and* (o) *have labels* $y^l = 1$ *and* $y^l = 0$ *respectively. The remaining points are unlabelled* $y^u =?$*, but their topology is essential to the construction of a good classifier.*

A human would classify all the points in the outer ring as 1 and the points in the inner circle as 0. We want an algorithm that reproduces this perceptual grouping.

We do this by considering each point $\mathbf{x}_i$ as a node in a fully connected undirected graph. The edges of the graph have weights corresponding to a similarity kernel. In our case, the weight between points $\mathbf{x}_i$ and $\mathbf{x}_j$ is

$$w_{ij} = \exp\left(-\frac{1}{\sigma}\|\mathbf{x}_i - \mathbf{x}_j\|^2\right),$$

where $\|\cdot\|$ denotes the L_2 norm. Hence, points that are close together will have high similarity (high w), whereas points that are far apart will have low similarity.
It is sensible to minimize the following error function to compute the unknown labels y^u:

$$E(y^u) = \frac{1}{2}\left(\sum_{i\in L, j\in L} w_{ij}(y_i^l - y_j^l)^2 + 2\sum_{i\in U, j\in L} w_{ij}(y_i^u - y_j^l)^2 + \sum_{i\in U, j\in U} w_{ij}(y_i^u - y_j^u)^2\right),$$

where L is the set of labelled points and U is the set of unlabelled points. If two points are close then w will be large. Hence, the only way to minimize the error function is to make these two nearby points have the same label y.

Let us introduce the adjacency matrix $\mathbf{W}$ with entries w_{ij} and the following block structure:

$$\mathbf{W} = \begin{pmatrix} \mathbf{W}_{ll} & \mathbf{W}_{lu} \\ \mathbf{W}_{ul} & \mathbf{W}_{uu} \end{pmatrix}.$$

where $\mathbf{W}_{uu}$ denotes the entries w_{ij} with $i, j \in U$.
Let $\mathbf{D} = diag(d_i)$ where $d_i = \sum_j w_{ij}$. That is,

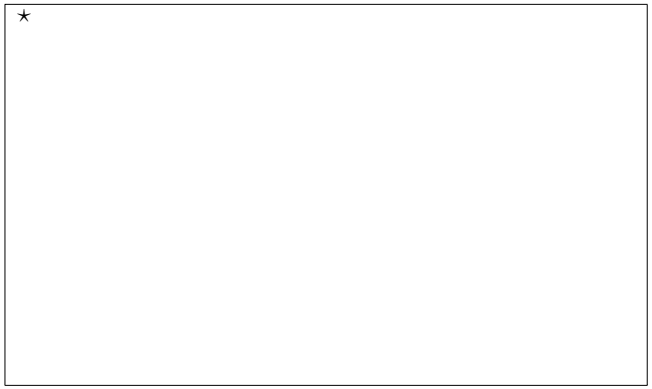
$$\mathbf{D} = \begin{pmatrix} \sum_j w_{1j} & 0 & 0 & \cdots & 0 \\ 0 & \sum_j w_{2j} & 0 & \cdots & 0 \\ & & \vdots & & \\ 0 & \cdots & 0 & 0 & \sum_j w_{Nj} \end{pmatrix}$$

$\mathbf{D}$ is a diagonal matrix whose i-th diagonal entry is the sum of the entries of row i of $\mathbf{W}$. Let the vector $\mathbf{y}_u$ contain all the unknown labels y^u and similarly let $\mathbf{y}_l$ contain all the labels y^l. Then, the error function can be written in matrix notation as follows:

$$E(\mathbf{y}_u) = \mathbf{y}_u^T(\mathbf{D}_{uu} - \mathbf{W}_{uu})\mathbf{y}_u - 2\mathbf{y}_l^T\mathbf{W}_{ul}\mathbf{y}_u + \mathbf{y}_l^T(\mathbf{D}_{ll} - \mathbf{W}_{ll})\mathbf{y}_l,$$

$$E(\mathbf{y}_u) = \begin{pmatrix} \mathbf{y}_l & \mathbf{y}_u \end{pmatrix} \begin{pmatrix} \mathbf{D}_{ll} - \mathbf{W}_{ll} & -\mathbf{W}_{lu} \\ -\mathbf{W}_{ul} & \mathbf{D}_{uu} - \mathbf{W}_{uu} \end{pmatrix} \begin{pmatrix} \mathbf{y}_l \\ \mathbf{y}_u \end{pmatrix}$$

Differentiating this error function and equating to zero, gives us our solution in terms of a linear system of equations:



where $0 \leq y^u \leq 1$ (**this is a relaxation of the original problem where the predictions are binary**) . A naive solution would cost $O(M^3)$, where $M = |U|$ is the number of unlabelled points, since $|L|$ is significantly smaller than $|U|$. However, using fast n-body methods in conjunction with conjugate gradients, it is possible to reduce the computation to $O(M \log M)$.

The solution that we have just obtained is actually not very different from a zero-mean function GP prediction (kernel ridge regression):

$$\mathbb{E}(\widehat{\mathbf{y}}_{n+1}|\widehat{\mathbf{y}}_{1:n}) = \mathbf{k}^T\mathbf{K}^{-1}\widehat{\mathbf{y}}_{1:n}$$

or in our current language:

$$\mathbf{y}_u = \mathbf{K}_{ul}\mathbf{K}_{ll}^{-1}\mathbf{y}_l$$

We have simply made the choice $\mathbf{K} = \mathbf{L}^{-1} = (\mathbf{D} - \mathbf{W})^{-1}$

⋆ Proof sketch:

The matrix $\mathbf{L}$ is known as the **Graph Laplacian**.

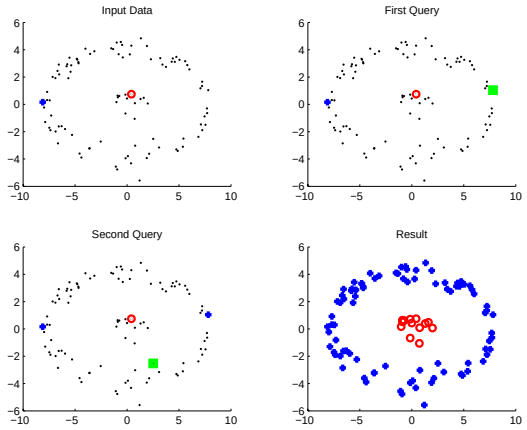
If we have many classes, we can use the same equation with voting

⋆ Proof sketch:

Once we have labels for all N points in the training data, a new point $\mathbf{x}_k$ in the test set data can be classified using the following classical kernel discriminant (Nadaraya-Watson estimate):

$$y_k = \frac{\sum_{i=1}^N w_{ik} y_i}{\sum_{i=1}^N w_{ik}}.$$

◇ ACTIVE LEARNING WITH GRAPH LAPLACIANS



The top-left plot shows the initial data $\mathbf{x}$*, where only two points have been labelled. By running the active learning algorithm, the computer asks the user to enter the label for a point that could minimize the Bayes risk the most (the square in the top-right plot). The process is then repeated in the bottom-left plot. The final classification using only these four labels is perfect.*