6 Kernel Methods

Definition. ***Kernel Methods***

1. ***Memory-based Method*** *Training data is not discarded during prediction (unlike neural nets, or bayesian regression). Requires a metric that defines measures of similarity of any two vector in the input space*

2. ***Kernel*** *Given a fixed nonlinear feature space mapping $\phi(\mathbf{x})$, the kernel function measures similarity*

$$k(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}(\mathbf{x})^T \boldsymbol{\phi}(\mathbf{x}') = \sum_{i=1}^{M} \boldsymbol{\phi}_i(\mathbf{x}) \boldsymbol{\phi}(\mathbf{x}')$$

   Linear kernel $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}'$

3. ***Valid Kernel*** *the Gram matrix $\mathbf{K}$ where $(\mathbf{K})_{nm} = k(\mathbf{x}_n, \mathbf{x}_m)$ is positive semidefinite for all possible $\mathbf{x}_n, \mathbf{x}_m$*

4. ***Dual Representation*** *Idea is for some linear models, i.e. perceptron, minimization of least squared loss often has an equivalent formulation comprised of entirely kernel functions*

5. ***Kernel Substitution (trick)*** *Replace scalar product of input vectors $\mathbf{x}$ with some other choice of kernel (i.e. nonlinear PCA)*

6. ***Gaussian Processes*** *Dual of probabilistic discriminative models gives rise to Gaussian processses*

6.4 Gaussian Processes

Definition. ***GP***

1. ***Bayesian Linear Regression Revisit***

$$y(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}) \qquad \textit{where} \quad p(\mathbf{w}) = \mathcal{N}\left(\mathbf{w} | \mathbf{0}, \alpha^{-1} \mathbf{I}\right)$$

   Given design matrix Φ over $\mathbf{x}_1, \cdots, \mathbf{x}_N$, we can write

$$\mathbf{y}(\mathbf{x}) = \Phi \mathbf{w}$$

$$\mathbb{E}\left\{\mathbf{y}\right\} = \Phi \mathbb{E}\left\{\mathbf{w}\right\} = \mathbf{0} \quad \textit{var}\left\{\mathbf{y}\right\} = \mathbb{E}\left\{\mathbf{y}\mathbf{y}^T\right\} = \Phi \mathbb{E}\left\{\mathbf{w}\mathbf{w}^T\right\} \Phi^T = \mathbf{K}$$

   where $\mathbf{K}$ defined with $\mathbf{K}_{nm} = k(\mathbf{x}_n, \mathbf{x}_m) = \frac{1}{\alpha} \boldsymbol{\phi}(\mathbf{x}_n)^T \boldsymbol{\phi}(\mathbf{x}_n)$. This is a special case of Gaussian process in that distributinon over a function $y(\mathbf{x})$ evaluated over some indexed set of input values yield a jointly multivariate Gaussian distribution

2. ***Motivation*** *Gaussian processes is a probability distribution over functions* $y(\mathbf{x})$ *such that the set of values of* $y(\mathbf{x})$ *evaluated at an arbitrary set of points* $\mathbf{x}_1, \cdots, \mathbf{x}_N$ *jointly have a Gaussian distribution. Usually set prior mean be zero, and the Gaussian process is completely determined by the variance, or the kernel function*

$$var\left\{\mathbf{x}_n, \mathbf{x}_m\right\} = k(\mathbf{x}_n, \mathbf{x}_m)$$

3. ***Gaussian Process*** *A stochastic process is a collection of random variables (functions), i.e.* $\{f(\mathbf{x}) : \mathbf{x} \in \mathcal{X}\}$ *for some index set* $\mathcal{X}$*. A Gaussian Process is a stochastic process such that any finite subcollection of random variables has multivariate Gaussian distribution. A collection of random variables* $\{f(\mathbf{x}) : \mathbf{x} \in \mathcal{X}\}$ *are said to be drawn from Gaussian processes with* ***mean function*** $m(\cdot)$ *and covariance function* $k(\cdot, \cdot)$ *if for any finite set of elements* $\mathbf{x}_1, \cdots, \mathbf{x}_m \in \mathcal{X}$*, the associated set of random variables* $f(\mathbf{x}_1), \cdots, f(\mathbf{x}_m)$ *has distribution*

$$\begin{pmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_N) \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} m(\mathbf{x}_1) \\ \vdots \\ m(\mathbf{x}_N) \end{pmatrix}, \begin{pmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \cdots & k(\mathbf{x}_1, \mathbf{x}_N) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_N, \mathbf{x}_1) & \cdots & k(\mathbf{x}_N, \mathbf{x}_N) \end{pmatrix}\right)$$

$$f(\cdot) = \mathcal{GP}(m(\cdot), k(\cdot, \cdot))$$

4. ***GP for Regression***

$$t_n = y_n + \epsilon \qquad where p(t_n|y_n) = \mathcal{N}\left(t_n|y_n, \beta^{-1}\right)$$

We can derive the conditional and marginal, as we want to get $p(\mathbf{t}, \mathbf{y})$

$$p(\mathbf{t}|\mathbf{y}) = \mathcal{N}\left(\mathbf{t}|\mathbf{y}, \beta^{-1}\mathbf{I}_N\right) \qquad p(\mathbf{y}) = \mathcal{N}\left(\mathbf{y}|\mathbf{0}, \mathbf{K}\right)$$

where $p(\mathbf{y})$ *is Gaussian as* $\mathbf{y}$ *drawn from a Gaussian process. We then compute marginal distribution of training data* $\mathbf{t}$

$$p(\mathbf{t}) = \int p(\mathbf{t}|\mathbf{y})p(\mathbf{y})d\mathbf{y} = \mathcal{N}\left(\mathbf{t}|\mathbf{0}, \mathbf{C}\right) \qquad \mathbf{C}(\mathbf{x}_n, \mathbf{x}_m) = k(\mathbf{x}_n, \mathbf{x}_m) + \beta^{-1}\delta_{nm}$$

$\mathbf{C}$ *captures both randomness in picking* $\mathbf{y}$ *and randomness in data noise* ϵ*. We want to predict target value* t_{N+1} *for a new input vector* $\mathbf{x}_{N+1}$ *given training* $(\mathbf{x}, \mathbf{t})$ *dataset, i.e.* $p(t_{N+1}|\mathbf{t}_N)$*. We first determine joint distribution of* $\mathbf{t}_{N+1}$

$$p(\mathbf{t}_{N+1}) = p(\mathbf{t}_1, \cdots, \mathbf{t}_{N+1}) = \mathcal{N}\left(\mathbf{t}_{N+1}|\mathbf{0}, \mathbf{C}_{N+1}\right) \qquad \mathbf{C}_{N+1} = \begin{pmatrix} \mathbf{C}_N & \mathbf{k} \\ \mathbf{k}^T & c \end{pmatrix}$$

we get

$$p(t_{N+1}|\mathbf{t}) = \mathcal{N}\left(m(\mathbf{x}_{N+1}), \sigma^2(\mathbf{x}_{N+1})\right) \qquad m(\mathbf{x}_{N+1}) = \mathbf{k}^T\mathbf{C}_N^{-1}\mathbf{t} \quad \sigma^2(\mathbf{x}_{N+1}) = c - \mathbf{k}^T\mathbf{C}_N^{-1}\mathbf{k}$$

where the mean and variance depends on $\mathbf{x}_{N+1}$

5. ***Characteristic of a kernel function*** *squared exponential with constant term and linear term*

$$k(\mathbf{x}_n, \mathbf{x}_m) = \theta_0 \exp -\frac{\theta_1}{2} \|\mathbf{x}_n - \mathbf{x}_m\|^2 + \theta_2 + \theta_3 \mathbf{x}_n^T \mathbf{x}_m$$

   θ_0 simply scales the curve vertically without changing the shape of the curve. As θ_1 increase, points far away will have higher correlations than before, so sampled function thend to be smoother overall. θ_2 influences variance regardless of value of x, θ_3 influences variance depending on values of x

6. ***Learning hyperparameters*** *Evaluaate likelihood function $p(\mathbf{t}|\mathbf{\Theta})$ where $\mathbf{\Theta}$ denotes hyperparameters of GP model, i.e. parameters in kernel function*