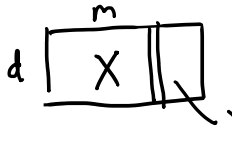
Gaussian Processes

Friday, March 27, 2020 3:05 PM

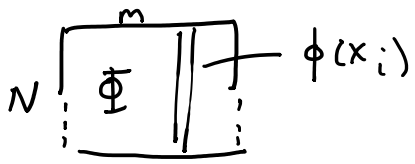
For ML and beyond - not in classical PAC framework since it's Bayesian

Notation: $x \in \mathbb{R}^d$, $S = ((x_i, y_i))_{i=1}^m$ as usual

feature map $\phi: \mathbb{R}^d \to \mathbb{R}^N$ (or, later, $\to$ Hilbert space, possibly ∞-dim)



(transposed of usual convention)



GP use kernels (like SVM), and as w/ kernel-trick, only need to know $\langle \phi(x), \phi(x') \rangle = k(x, x')$ and don't need to know ϕ explicitly

Idea of GP:

we predict w/ a function f ($f: X \to \mathbb{R}$ for regression, $f: X \to \{\pm 1\}$ for classification)

and instead of constraining $f \in \mathcal{H}$,

we'll instead encode prior knowledge via

a Bayesian prior on f

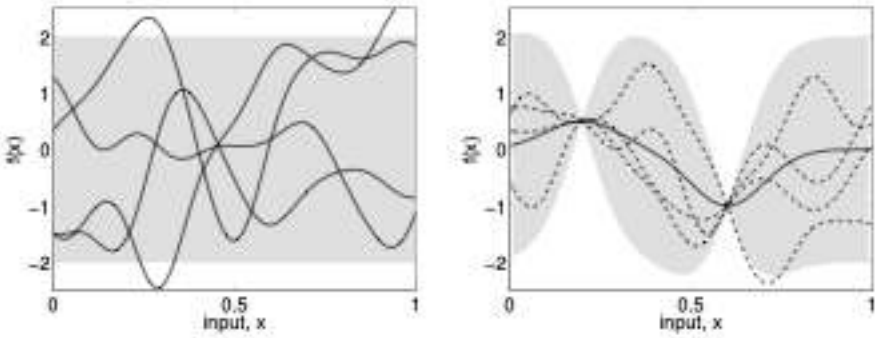
i.e. f is a random function = stochastic process

(generalizes random variable)

then, given observations S,

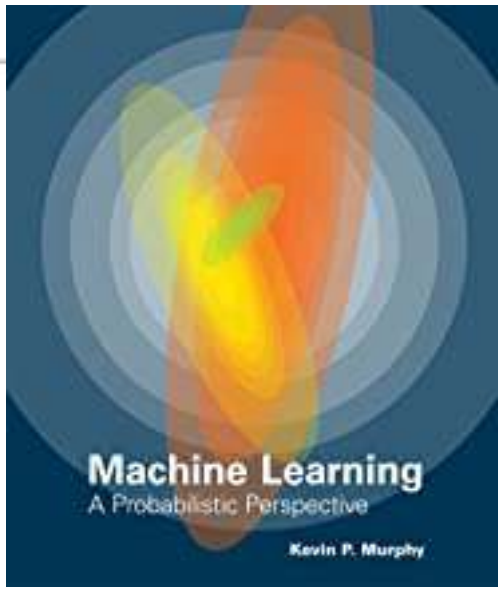
we update, ie., compute the posterior distribution using Bayes' Rule

C. E. Rasmussen & C. K. I. Williams, Gaussian Processes for Machine Learning, the MIT Press, 2006, ISBN 026218253X. www.GaussianProcess.org/gpml



(a), prior (b), posterior

Figure 1.1: Panel (a) shows four samples drawn from the prior distribution. Panel (b) shows the situation after two datapoints have been observed. The mean prediction is shown as the solid line and four samples from the posterior are shown as dashed lines. In both plots the shaded region denotes twice the standard deviation at each input value x.



Classification is a bit more complicated (we'll need to "squash" the output to the range $[0,1]$ using, eg, the logistic function, and this complicates things)

Regression is cleaner, so we'll focus on this.

We essentially always do the kernel trick, so really focus on linear regression

our model is $f(x) = \langle w, x \rangle$ or $f(x) = \langle w, \psi(x) \rangle$

$w \in \mathbb{R}^d$ $\quad$ $w \in \mathcal{H}$ ← Hilbert space, $\psi : \mathbb{R}^d \to \mathcal{H}$

So a prior on f really means a prior on w

For least-squares, we assume error term is $\varepsilon \sim N(0, \sigma_m^2)$

$y = f(x) + \varepsilon$ for some true f.

ie., $y = \langle w, x \rangle + \varepsilon$ for some true but unknown w

If we know w, x, then

$y \mid w, x \sim N(\langle w, x \rangle, \sigma_m^2)$

Gaussian facts:
1) Sum of independent Gaussians is Gaussian (i.e. convolve PDFs)

Thus, given a dataset $X = d\boxed{\quad}^{m}$,

Adding PDF's is different — that is a mixture model
2) product of Gaussian r.v. is a Gaussian r.v
3) conditional, marginal of multivariate normal is still normal
cf. A.2 [RW]

the likelihood is $p(\vec{y} \mid X, w) \sim N(X^T w, \sigma_n^2 I)$

Normal distr. aka Gaussian $N(\mu, \Sigma)$

We encode the prior on w as $w \sim N(0, \Sigma_p)$ → today, "Σ" is a matrix usually except $\sum_{i=1}^{m}$ is a sum

Given data (y, X), how to update the distribution on w? (known Gaussian, known Gaussian)

Bayes' Rule

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}, \quad p(w \mid y, X) = \frac{p(y \mid X, w)\, p(w)}{p(y \mid X)}$$

← independent of w, so can ignore

product of two Gaussians is a Gaussian
so with some computation, the posterior of w is

$$p(w \mid X, y) \propto N(\bar{w}, A^{-1})$$

$$\text{with } A := \frac{1}{\sigma_n^2} X X^T + \Sigma_p^{-1}$$

$$\bar{w} := \frac{1}{\sigma_n^2} A^{-1} X y$$

So we have a full distribution on w (more than usual PAC learning)!

If we need to pick out a single w to get a "point estimate",
then usual idea is to pick the median,
ie. maximum a posteriori (MAP) estimate.

For a Gaussian, median = mean, so MAP is $\bar{w}$

To make a prediction at a new point X_*, again we get a full distribution

$$p(f(X_*) \mid X_*, X, y) = \int p(f(X_*) \mid X_*, w) \cdot p(w \mid X, y)\, dw$$

$$\sim N(\langle \bar{w}, X_* \rangle, \langle X_*, A^{-1} X_* \rangle)$$

and again, to get a point estimate, take the median/mean,
so $\langle \bar{w}, X_* \rangle$

alternative derivation: if we posit that $f(X_*)$ is jointly Gaussian

with y, $\begin{bmatrix} y \\ f(X_*) \end{bmatrix} \sim N\left(\begin{bmatrix} \mu \\ \mu_* \end{bmatrix}, \begin{bmatrix} \Sigma & \Sigma_* \\ \Sigma_*^T & \Sigma_{**} \end{bmatrix} \right)$

(μ_*: unknown)

then $f(X_*) \mid X_*, X, y \sim N(\mu_*, \tilde{\Sigma})$

w/ expressions for μ_* and for $\tilde{\Sigma}$ via Schur complement (see eq. 15.7 to 15.9 Murphy)

repeating this for kernel version

replace X  with Φ 

$f(x) = \langle w, \phi(x) \rangle$ is model

almost everything stays the same, just replace X with Φ

i.e., now $A = \frac{1}{\sigma_m^2} \Phi \Phi^T + \Sigma_p^{-1}$

$\Phi \Phi^T \to \dim(\mathcal{H}) \times \dim(\mathcal{H})$

$\bar{w} = \frac{1}{\sigma_m^2} A^{-1} \Phi y \in \mathcal{H}$

$f(X_*) \mid X_*, X, y \sim N(\langle \bar{w}, \phi(X_*) \rangle, \langle \phi(X_*), A^{-1} \phi(X_*) \rangle)$

... though computing A^{-1} is awkward since it is $\dim(\mathcal{H}) \times \dim(\mathcal{H})$

trick is, again, the matrix inversion lemma

$\Rightarrow \bar{w} = \Sigma_p \Phi (K + \sigma_m^2 I)^{-1} y$

$A^{-1} = \Sigma_p - \Sigma_p \Phi (K + \sigma_m^2 I)^{-1} \Phi^T \Sigma_p$

where $K := \Phi^T \Sigma_p \Phi \in \mathbb{R}^{m \times m}$

and again we can use the kernel trick: all the $\mathcal{H}$ or ϕ terms combine to be in the form

$\Phi^T \Sigma_p \Phi$ or $\phi(X_*)^T \Sigma_p \Phi$ or $\phi(X_*)^T \Sigma_p \phi(X_*)$

So... define the kernel / covariance function

$$K(x,x') = \langle \phi(x), \Sigma_p \phi(x') \rangle$$
$$= \langle \psi(x), \psi(x') \rangle \text{ if } \psi(x) = \Sigma_p^{1/2} \phi(x)$$

Recall that our main choices are

① prior distribution on w
(and we've fixed this to be a centered normal, so main choice is Σ_p)

② nonlinear map ϕ

... so you can see that equivalently, just specify the kernel

Function Space pt. of view

Def A Gaussian Process (GP) is a collection (ex: in time-series, indexed by $\mathbb{R}$; in spatial-stat, indexed by $\mathbb{R}^2$ or $\mathbb{R}^3$)

of random variables, any finite number of which have a joint Gaussian distribution

(implicitly imposes a consistency requirement or "marginalization property")

$f \sim GP(m, K)$ $\quad f: X \to \mathbb{R}$ a function
$m: X \to \mathbb{R}$ the mean function
$K: X \times X \to \mathbb{R}$ is a pos. semidef kernel (covariance)

$\Rightarrow$ ① $\forall x, \; \mathbb{E} f(x) = m(x)$
(typically set $m(x) \equiv 0$)

② $Cov(f(x), f(x')) = K(x, x')$

③ $\forall m, \forall \{x_1, \dots, x_m\} \subseteq X$,

$(f(x_1), \dots, f(x_m)) \sim N(\vec{\mu}, K)$

$\vec{\mu}_i := m(x_i), \; K_{ij} = k(x_i, x_j)$

Given K, we saw above we can characterize the prediction, just need some $O(m^3)$ computation (invert a matrix ...)

In practice, usually want to tune hyperparameters of the kernel

eg, interpretation as smoothness of f,
or decorrelation length-scale

You can solve for hyperparameters in a principled way (but issues w/ nonconvexity) in $O(m^3)$ too

Comparison to Kernel-SVM

for simplicity, use trivial kernel, $\Psi = I$,
$K(x,x') = \langle x, x' \rangle$

(Soft-)SVM solves $\min_w \frac{1}{2}\|w\|^2 + \lambda^{-1} \sum_{i=1}^m \lfloor 1 - y_i \langle w, x_i \rangle \rfloor_+$

GP does more than just give a single pt. estimate, but if you used GP to get the MAP estimate, it is

$$\operatorname{argmin}_w \frac{1}{2}\|w\|^2 + \sum_{i=1}^m -\log\left(p(y_i \mid \langle w, x_i \rangle)\right)$$

where $p(y \mid \langle w, x \rangle)$ is the likelihood that we already discussed

and usual choice is $y \mid w, x \sim N(\langle w, x \rangle, \sigma_m^2)$

You can change this, though there is no likelihood that recovers the hinge loss (and hence SVM)

So, both GP & SVM can work w/ kernels, and both give estimates of the form $\operatorname{argmin}_w \frac{1}{2}\|w\|^2 + \sum \text{loss}(w, y_i, x_i)$

- SVM is based on minimization
- GP minimization can be done in closed form, and you recover an entire posterior distribution
  (but using GP for classification is messier)

Both Rasmussen & Williams' book and Murphy's book discuss more connections

eg., - GP vs neural networks
- Smoothing splines as special case of GP, usually for 1D or 2D data