TODO
distribution parts in bishop maybe

Probabilistic Artificial Intelligience

task:

1. reintepret the lecture and add important comments of Krause. Formulas in latex and give a space for detailed proof.
2. one section for reading and maybe essential questions in homework
3. yandex or berkley materials to complement

Introduction and probability

Lecture Notes

Topics Covered

- Probabilistic foundations of AI
- Bayesian learning (GPs, Bayesian deep learning, variational inference, MCMC)
- Bandits & Bayesian optimization
- Planning under uncertainty (MDPs, POMDPs)
- (Deep) Reinforcement learning

- Applications (in class and in project)

Review:Probability

- **Probability space** $(\Omega, \mathcal{F}, \mathcal{P})$
- set of **atomic events** Ω
- set of all **non-atomic events** $\mathcal{F}$
- $\mathcal{F}$is a $\sigma\text{-algebra}$(closed under complements and countable unions)
  - $\Omega \in \mathcal{F}$
  - $A \in \mathcal{F} \rightarrow \Omega \backslash A \in \mathcal{F}$
  - $A_1, ..., A_n, ... \in \mathcal{F} \rightarrow \bigcup_i A_i \in \mathcal{F}$
- **Probability measure** $\mathcal{P} : \mathcal{F} \rightarrow [0, 1]$
  - for $A \in \mathcal{F}$, $P(A)$ is the probability that event A happens

Probability Axioms

- Normalization: $P(\Omega) = 1$
- Non-negativity: $P(A) \geq 0 \text{ for all} A \in \mathcal{F}$
- $\sigma\text{-additivity}$:

$$\forall A_1, ..., A_n, ... \in \mathcal{F} \text{ disjoint:} P(\bigcup_{I=1}^{\infty} A_i) = \sum_{I=1}^{\infty} P(A_i)$$

Interpretation of Probabilities

- Frequentist interpretation
  - $P(A)$ is relative frequency of A in repeated experiments
  - Can be difficult to assess with limited data
- Bayesian interpretation
  - $P(A)$ is "degree of belief" A that will occur
  - Where does this belief come from?
  - Many different flavors (subjective, objective, pragmatic, …)

Random Variables

- Let D be some set (e.g., the integers)
- A random variable X is a mapping $X : \Omega \rightarrow D$
- For some $x \in D$, we say

$$P(X = x) = P(\omega \in \Omega : X(\omega) = x) \qquad \text{“probability that variable X assumes state x”}$$

Specifying Probability Distributions through RVs

- **Bernoulli** distribution: “(biased) coin flips”$D = \{H, T\}$
  Specify $P(X = H) = p$. Then $P(X = T) = 1 - p$.
  Note: can identify atomic ev.ω with $\{X = H\}$,$\{X = T\}$
- **Binomial** distribution counts no. heads S in n flips
- **Categorical** distribution: “(biased) m-sided dice” $D = \{1, ..., m\}$
  Specify $P(X = i) = p_i$, s.t. $p_i \geq 0, \sum p_i = 1$
- **Multinomial** distribution counts the number of
  outcomes for each side for n throws

Joint Distributions

- random vector $\mathbf{X} = [X_1(\omega), ..., X_n(\omega)]$
- can specify $P(X_1 = x_1, ..., X_n = x_n)$ directly (atomic events are assignments $x_1, ..., x_n$)
- **Joint Distribution** describes relationship among all variables

Conditional Probability

- Formal definition:

$$P(a|b) = \frac{P(a \land b)}{P(b)} \text{ if } P(b) \neq 0$$

- **Product rule** $P(a \land b) = P(a|b)P(b)$
- for distributions: $P(A, B) = P(A|B)P(B)$
  (set of equations, one for each instantiation of A, B)
  $\forall a, b : P(A = b, B = b) = P(A = a|B = b) \cdot P(B = b)$
- **Chain(product) rule** for multiple RVs:$X_1, .., X_n$
  $P(X_1, .., X_n) = P(X_{1:n}) = P(X_1) \cdot P(X_2|X_1) \cdot ... \cdot P(X_n|X_{1:n-1})$

The Two Rules for Joint Distributions

- **Sum rule (Marginalization)**
  $P(X_{1:i-1}, X_{i+1:n}) = \sum_{x_i} P(X_{1:i-1}, X_i = x_i, X_{i+1:n})$
- **Product rule (chain rule)**

Bayes' Rule

Given:

- **Prior** $P(X)$
- **Likelihood** $P(X|Y) = \frac{P(X,Y)}{P(Y)}$

Then:

- **Posterior**
  $P(X|Y) = \frac{P(X)P(Y|X)}{\sum_{X=x} P(X=x)P(Y|X=x)}$

Independent RVs

- Random variables$X_1, ..., X_n$ are called **independent** if
  $P(X_1 = x_1, ..., X_n = x_n) = P(x_1)P(x_2)\dots P(x_n)$

Conditional Independence

- Rand. vars. X and Y conditionally independent given Z **iff** for all x, y, z:
  $P(X = x, Y = y|Z = z) = P(X = x|Z = z)P(Y = y|Z = z)$
- If $P(Y = y|Z = z) > 0$, that is equivalent to
  $P(X = x|Y = y, Z = z) = P(X = x|Z = z)$
  Similar for sets of random variables$\mathbf{X}, \mathbf{Y}, \mathbf{Z}$
  we write:$\mathbf{X} \perp \mathbf{Y} | \mathbf{Z}$

Problems with High-dim. Distributions

- Suppose we have n binary variables, then we have 2^{n-1} variables to specify
  $P(X_1 = x_1, .., X_n = x_n)$
- Computing marginals:
  - Suppose we have joint distribution $P(X_1, .., X_n)$
  - Then (acc. to sum rule)

$$P(X_i = x_i) = \sum_{x_{1:i-1}, x_{i+1:n}} P(x_1, ..., x_n)$$

  - If all X_i are binary: this sum has 2^{n-1} terms
- Conditional queries
  - Suppose we have joint distribution $P(X_1, .., X_n)$
  - Compute distribution of some variables given values for others:
    $P(X_1 = \cdot|X_7 = x_7) = \frac{P(X_1=\cdot, X_7=x_7)}{P(X_7=x_7)} = \frac{1}{Z}P(X_1 = \cdot, X_7 = x_7)$
    where, $Z = \sum_{x_1} P(X_1 = x_1, X_7 = x_7)$
    where, $P(X_1 = x_1, X_7 = x_7) = \sum_{x_{2:6}} \sum_{x_{8:n}} P(X_{1:n} = x_{1:n})$, 2^{n-2} terms for binomial X_i
- Representation (parametrization)
- Learning (estimation)
- Inference (prediction)

Gaussian Distribution

- univariate :

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} exp(-\frac{(x-\mu)^2}{2\sigma^2})$$

  σ: Std. dev., μ: mean
- multivaraite:

$$p(\mathbf{x}) = \frac{1}{2\pi\sqrt{|\Sigma|}} exp(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu))$$

  where $\Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{21} & \sigma_2^2 \end{pmatrix}$, $\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}$.
- **Multivariate Gaussian distribution**

$$\mathcal{N}(y;\Sigma,\mu) = \frac{1}{((2\pi)^{n/2}\sqrt{|\Sigma|}} exp(-\frac{1}{2}(y-\mu)^T\Sigma^{-1}(y-\mu))$$

  where $\Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & ... & \sigma_{1n} \\ \vdots & & & \vdots \\ \sigma_{n1} & \sigma_{n2} & ... & \sigma_n^2 \end{pmatrix}$,
  $\sigma_{ij} = \mathbb{E}((x_i-\mu_i)(x_j-\mu_j))$,
  $\sigma_i^2 = \mathbb{E}((x_i-\mu_i)^2) = Var(x_i)$.
  The joint distribution over n variables requires **only $O(n^2)$ parameters**.
- **Fact:Gaussians are independent iff they are uncorrelated:**
  $X_i \perp X_j \Leftrightarrow \sigma_{ij} = 0$
- Multivariate Gaussians have important properties:
  - **Compact representation** of high-dimensional joint distributions
  - **Closed form inference**

Bayesian Inference in Gaussian Distributions

- Suppose we have a Gaussian random vector
  $\mathbf{X} = \mathbf{X}_V = [X_1, ..., X_d] \sim \mathcal{N}(\mu_V, \Sigma_{VV})$
- Hereby $V = \{1, ..., d\}$ is an index set.
- Suppose we consider a subset of the variables
  $A = \{i_1, ..., i_k\}, \quad i_j \in V$
- The **marginal distribution** of variables indexed by A is:
  $\mathbf{X}_A = [X_{i_1}, ..., X_{i_k}] \sim \mathcal{N}(\mu_A, \Sigma_{AA})$

where $\mu_A = [\mu_{i_1}, ..., \mu_{i_k}]$, $\Sigma_{AA} = \begin{pmatrix} \sigma_{i_1 i_1} & ... & \sigma_{i_1 i_k} \\ \vdots & \ddots & \vdots \\ \sigma_{i_k i_1} & ... & \sigma_{i_k i_k} \end{pmatrix}$

Conditional Distributions

- Suppose we have a Gaussian random vector $\mathbf{X} = \mathbf{X}_V = [X_1, ..., X_d] \sim \mathcal{N}(\mu_V, \Sigma_{VV})$
- Further, suppose we take two disjoint subsets of V $A = \{i_1, ..., i_k\} \quad B = \{j_1, ..., j_m\}$
- The **conditional distribution** $p(\mathbf{X}_A | \mathbf{X}_B = \mathbf{x}_B) = \mathcal{N}(\mu_{A|B}, \Sigma_{A|B})$ is Gaussian, **where**

$$\mu_{A|B} = \mu_A + \Sigma_{AB}\Sigma_{BB}^{-1}(\mathbf{x}_B - \mu_B)$$

$$\Sigma_{A|B} = \Sigma_{AA} - \Sigma_{AB}\Sigma_{BB}^{-1}\Sigma_{BA}$$

where $\Sigma_{AB} = \begin{pmatrix} \sigma_{i_1 j_1} & ... & \sigma_{i_1 j_m} \\ \vdots & \ddots & \vdots \\ \sigma_{i_k j_1} & ... & \sigma_{i_k j_m} \end{pmatrix} \in \mathbb{R}^{k \times m}$

Multiples of Gaussians are Gaussian

- Suppose we have a Gaussian random vector $\mathbf{X} = \mathbf{X}_V = [X_1, ..., X_d] \sim \mathcal{N}(\mu_V, \Sigma_{VV})$
- Take a matrix $M \in \mathbb{R}^{m \times d}$
- Then the random vector $\mathbf{Y} = \mathbf{MX}$ is Gaussian:

$$\mathbf{Y} \sim \mathcal{N}(\mathbf{M}_{\mu_V}, \mathbf{M}\Sigma_{VV}\mathbf{M}^T$$

Sums of Gaussians are Gaussian

- Suppose we have independent two Gaussian random vectors $\mathbf{X} = \mathbf{X}_V = [X_1, ..., X_d] \sim \mathcal{N}(\mu_V, \Sigma_{VV})$ $\mathbf{X}' = \mathbf{X}'_V = [X'_1, ..., X'_d] \sim \mathcal{N}(\mu'_V, \Sigma'_{VV})$

Bayesian Learning

Lecture Notes

Recall: linear regression

- $y \approx \mathbf{w}^T \mathbf{x} = f(\mathbf{x})$

Recall: ridge regression

- Regularized optimization problem: $\min_{\mathbf{w}} \sum_i (y_i - \mathbf{w}^T \mathbf{x}_i)^2 + \lambda \|\mathbf{w}\|_2^2$
- Can optimize using (stochastic) gradient descent, or still find **analytical solution**: $\hat{\mathbf{w}} = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y}$

Ridge regression as Bayesian inference

♿ proof to do

Assume $p(\mathbf{w}) = \mathcal{N}(0, \sigma_p^2 \cdot \mathbf{I}))$ independent of $\mathbf{x}_{1:n}$
conditional iid. $\Rightarrow p(y_{1:n} | \mathbf{w}, \mathbf{x}_{1:n}) = \prod_{i=1}^n p(y_i | \mathbf{w}, \mathbf{x}_i)$
In particular: $p(y_i | \mathbf{w}, \mathbf{x}_i) = \mathcal{N}(y_i; \mathbf{w}^T \mathbf{x}_i, \sigma_n^2) \Leftrightarrow y_i = \mathbf{w}^T \mathbf{x}_i + \varepsilon_i \quad \varepsilon_i \sim \mathcal{N}(0, \sigma_n^2)$
ParseError: KaTeX parse error: \cr valid only within a tabular/array environment

Ridge regression = MAP estimation

- Ridge regression can be understood as finding the **Maximum A Posteriori (MAP) parameter estimate** for a linear regression problem, assuming that
- The **noise** $P(y | \mathbf{x}, \mathbf{w})$ is **(cond.) iid Gaussian** and
- The **prior** $P(\mathbf{w})$ on the model parameters $\mathbf{w}$ is **Gaussian**
- However, ridge regression returns a single model
- Such a **point estimate** does not quantify **uncertainty**

Bayesian Linear Regression (BLR)

- Key idea: Reason about full posterior of $\mathbf{w}$, not only its mode
- For Bayesian linear regression with Gaussian prior and Gaussian likelihood, posterior has **closed form**

Posterior distributions in BLR

- Prior: $p(\mathbf{w} = \mathcal{N}(0, \mathbf{I})$
- Likelihood: $p(y | \mathbf{x}, \mathbf{w}, \sigma_n) = \mathcal{N}(y; \mathbf{w}^T \mathbf{x}, \sigma_n^2)$
- Posterior: ParseError: KaTeX parse error: \cr valid only within a tabular/array environment
- $\bar{\mu}$ is ridge regression solution!
- Precision matrix: $\bar{\Lambda} = \bar{\Sigma}^{-1} = \sigma_n^{-2} \mathbf{X}^T \mathbf{X} + \mathbf{I}$

Making predictions in BLR

- For test point $\mathbf{x}^*$, define $f^* = \mathbf{w}^T\mathbf{x}^*$. Then:
  ParseError: KaTeX parse error: \cr valid only within a tabular/array environment

♿ ***proof to do***

Aleatoric vs. epistemic uncertainty

- Uncertainty about f^* : $\bar{\Sigma} \leftarrow \text{(epistemic)}$
- Noise/Uncertainty about y^* given f^* : $\sigma_n^2 \leftarrow \text{(aleatoric)}$
- Can distinguish two forms of uncertainty:
  - **Epistemic uncertainty**: Uncertainty about the model due to the lack of data
  - **Aleatoric uncertainty**: Irreducible noise

Bayesian Linear Regression(cont'd)

Lecture Notes

- Observations: Conditional Linear Gaussians
- If X, Y are jointly Gaussian, then $p(X|Y=y)$ is Gaussian, with mean linearly dependent on y:
  $p(X=x|Y=y) = \mathcal{N}(x; \mu_{X|Y}\sigma^2_{X|Y})$
  $\mu_{X|Y} = \mu_X + \sigma_{XY}\sigma_Y^2(y - \mu_Y)$
- Thus random variable X can be viewed as a linear function of Y with independent Gaussian noise added
  $X = a \cdot Y + b + \varepsilon$, where $a = \sigma_{XY}\sigma_Y^2$, $b = \mu_X - \sigma_{XY}\sigma_Y^2\mu_Y$
- The converse also holds.

Ridge regression vs Bayesian lin. regression

- Ridge regression: predict using *MAP estimate* for weights
  $\hat{\mathbf{w}} = \arg\max_{\mathbf{w}} p(\mathbf{w}|\mathbf{x}_{1:n}, y_{1:n})$
  $p(y^*|\mathbf{x}^*, \hat{\mathbf{w}}) = \mathcal{N}(y^*; \hat{\mathbf{w}}^T\mathbf{x}^*, \sigma_n^2)$
- BLR: predict by averaging all $\mathbf{w}$ acc. to posterior:
  $p(y^*|\mathbf{X}, \mathbf{y}, \mathbf{x}^*) = \int p(y^*|\mathbf{x}^*, \mathbf{w})p(\mathbf{w}|\mathbf{x}_{1:n}, \mathbf{y}_{1:n})d\mathbf{w} = \mathcal{N}(\bar{\mu}^T\mathbf{x}^*, \mathbf{x}^{*T}\bar{\Sigma}\mathbf{x}^* + \sigma_n^2)$
- Thus, ridge regression can be viewed as approximating the full posterior by **(placing all mass on) its mode**
  ParseError: KaTeX parse error: \cr valid only within a tabular/array environment
- *Note*: $\delta_{\hat{\mathbf{w}}}(\cdot)$ is such that $\int f(\mathbf{w})\delta_{\hat{\mathbf{w}}}(\mathbf{w})d\mathbf{w} = f(\hat{\mathbf{w}})$

Choosing hyperparameters

- In BLR, need to specify the (co-)variance of the prior σ_p and the variance of the noise σ_n
- These are **hyperparameters** of the model (governing the distribution of the parameters $\mathbf{w}$)
- How to choose? One option:
  - Choose $\hat{\lambda} = \frac{\hat{\sigma}_n^2}{\hat{\sigma}_p^2}$ via cross-validation
  - Then estimate $\hat{\sigma}_n^2 = \frac{1}{n}\sum_{i=1}^n (y_i - \hat{\mathbf{w}}^T \mathbf{x}_i)^2$ as the empirical variance of the residual, and solve for $\hat{\sigma}_p^2 = \frac{\hat{\sigma}_n^2}{\hat{\lambda}}$
- Another option: marginal likelihood of the data, see <>

Side note: Graphical models

- Have seen: Can represent arbitrary joint distributions as product of conditionals via chain rule
- Often, factors only depend on subsets of variables
- Can represent the resulting product as a directed acyclic graph
- Graphical model for BLR (see lecture notes)

Recursive Bayesian updates

- "Today's posterior is tomorrow's prior"
- Surpose that:
  Prior: $p(\theta)$, observe $y_{1:n}$, s.t. $p(y_{1:n}|\theta) = \prod_{i=1}^n p_i(y_i|\theta)$
  for BLR: $\theta \equiv \mathbf{w}$, $p_i(y_i|\theta) \equiv p(y_i|\mathbf{w}, \mathbf{x}_i)$
  Define $p^{(j)}(\theta)$ to be the posterior afer recurring the first j observation. $p^{(j)}(\theta) = p(\theta|y_{1:j})$
- $p^{(0)}(\theta) = p(\theta) = \mathcal{N}(0, \sigma_p \cdot \mathbf{I})$
  Surpose we have cumputed $p^{(j)}(\theta) \equiv \mathcal{N}(\mu^{(j)}, \Sigma^{(j)}) \leftarrow$ posterior $\theta^{(j)} = \{\mu^{(j)}, \Sigma^{(j)}\}$
  and observed y_j.
- $p^{(j+1)}(\theta) = p(\theta|y_{1:j+1}) = \frac{1}{Z}p(\theta|y_{1:j})p(y_{j+1}|\theta, y_{1:j}) = \mathcal{N}(\mu^{(j+1)}, \Sigma^{(j+1)})$
  where, $\theta^{(j+1)} = \{\mu^{(j+1)}, \Sigma^{(j+1)}\}, \quad p(\theta|y_{1:j}) = p^{(j)}(\theta), \quad p(y_{j+1}|\theta, y_{1:j}) = p_{j+1}(y_{j+1}|\theta)$

Summary Bayesian Linear Regression

- **Bayesian linear regression** makes same modeling assumptions as ridge regression (Gaussian prior on weights, Gaussian noise)
- BLR computes / uses **full posterior distribution** over the weights rather than the mode only
- Thus, it captures **uncertainty in weights**, and allows to separate epistemic from aleatoric uncertainty
- Due to independence of the noise, can do **recursive updates** on the weights

Kalman Filters

Lecture Notes

Kalman filters

- Track objects over time using noisy observations
  - E.g., robots moving, industrial processes,...
- State described using **Gaussian variables**
  - E.g., location, velocity, acceleration in 3D
- Assume conditional linear Gaussian dependencies for states and observations

Kalman Filters: The Model

- $X_1, ..., X_T$: Location of object being tracked
- $Y_1, ..., Y_T$: Observations
- $P(X_1)$: **Prior** belief about location at time 1 (Gaussian)
- $P(X_{t+1}|X_t)$: **Motion Model**
  - How do I expect my target to move in the environment?
    $\mathbf{X}_{t+1} = \mathbf{F}\mathbf{X}_t + \varepsilon_t$, where $\varepsilon_t \in \mathcal{N}(0, \Sigma_x)$
- $P(Y_t|X_t)$: **Sensor model**
  - What do I observe if target is at location X_t?
    $\mathbf{Y}_t = \mathbf{H}\mathbf{X}_t + \eta_t$, where $\eta_t \in \mathcal{N}(0, \Sigma_y)$
- Assumptions:
  Known: $\mathbf{X}_{t+1} = \mathbf{F}\mathbf{X}_t + \varepsilon_t$, $\mathbf{Y}_t = \mathbf{H}\mathbf{X}_t + \eta_t, \qquad \varepsilon_{1:t}, \eta_{1:t}$ independent
  implies that: $X_{t+1} \perp X_{1:t-1} | X_t$, and $Y_{t+1} \perp Y_{1:t-1}, X_{1:t-1} | X_t$
  ParseError: KaTeX parse error: \cr valid only within a tabular/array environment

Bayesian filtering

- Start with $P(X_1) = \mathcal{N}(\mu, \Sigma)$
- At time t
  - Assume we have $P(X_t|Y_{1,\ldots,t-1})$
  - **Conditioning**: $P(X_t|Y_{1,\ldots t}) = \frac{1}{Z} P(X_t|Y_{1:t-1}) P(Y_t|X_t, Y_{1:t-1})$, where $P(Y_t|X_t, Y_{1:t-1}) = P(Y_t|X_t)$, so that $Z = \int P(X_t|Y_{1:t-1}) P(Y_t|X_t) dX_t$
  - **Prediction**: $P(X_{t+1}|Y_{1,\ldots t}) = \int P(X_{t+1}, X_t|Y_{1:t}) dX_t = \int P(X_{t+1}|X_t, Y_{1:t}) P(X_t|Y_{1:t}) dX_t = \int P(X_{t+1}|X_t) P(X_t|Y_{1:t}) dX_t$
  - For Gaussians, can compute these integrals in closed form!
- Example: Random walk in 1D
  - Transition / motion model: $P(x_{t+1}|x_t) = \mathcal{N}(x_t, \sigma_x^2)$
    $x_{t+1} = x_t + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, \sigma_x^2)$
  - Sensor model: $P(y_t|x_t) = \mathcal{N}(x_t, \sigma_y^2)$
    $y_t = x_t + \eta_t, \quad \eta_t \sim \mathcal{N}(0, \sigma_y^2)$

  - State at time t: $P(x_t|y_{1:t}) = \mathcal{N}(\mu_t, \sigma_t^2)$
  - $\rightarrow \mu_{t+1} = \frac{\sigma_y^2\mu_t+(\sigma_t^2+\sigma_x^2)y_{t+1}}{\sigma_t^2+\sigma_x^2+\sigma_y^2} \quad \sigma_{t+1} = \frac{(\sigma_t^2+\sigma_x^2)\sigma_y^2}{\sigma_t^2+\sigma_x^2+\sigma_y^2}$

General Kalman update

- Transition model: $P(\mathbf{x}_{t+1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t+1}; \mathbf{F}\mathbf{x}_t, \Sigma_x)$
- Sensor model: $P(\mathbf{y}_t|\mathbf{x}_t) = \mathcal{N}(\mathbf{y}_t; \mathbf{H}\mathbf{x}_t, \Sigma_y)$
- **Kalman Update**: ParseError: KaTeX parse error: \cr valid only within a tabular/array environment
- **Kalman Gain**:

$$\mathbf{K}_{t+1} = (\mathbf{F}\boldsymbol{\Sigma}_t\mathbf{F}^T + \boldsymbol{\Sigma}_\mathbf{x})\mathbf{H}^T(\mathbf{H}(\mathbf{F}\boldsymbol{\Sigma}_t\mathbf{F}^T + \boldsymbol{\Sigma}_\mathbf{x})\mathbf{H}^T + \boldsymbol{\Sigma}_y)^{-1}$$

- Can compute $\boldsymbol{\Sigma}_t$ and $\mathbf{H}_t$ **offline**

BLR vs Kalman Filtering

- Can view Bayesian linear regression as a form of a Kalman filter!
  - Hidden variables are the weights
  - Forward model is constant (identity)
  - Observation model at time t is determined by data point x_t

Gaussian Process

Lecture Notes

What about nonlinear functions?

- Recall: Can apply linear method (like BLR) on nonlinearly transformed data. However, computational cost increases with dimensionality of the feature space!
  $f(\mathbf{x}) = \sum_{i=1}^d w_i\phi_i(\mathbf{x})$
  In d-dim,: $\mathbf{x} = [x_1, ..., x_d]$, $\Phi(\mathbf{x}) = [1, x_1, ..., x_d, x_1^2, ..., x_d^2, x_1x_2, ..., x_{d-1}x_d, ..., x_1 \cdot ... \cdot x_m, ..., x_{d-m+1} \cdot ... \cdot x_d] \leftarrow O(d^m)$ monomials of deg m

The "Kernel Trick"

- Express problem s.t. it only depends on inner products
- Replace inner products by kernels
- $\mathbf{x}_i^T\mathbf{x}_j \Rightarrow k(\mathbf{x}_i, \mathbf{x}_j)$
- $\Phi(\mathbf{x}) = [\text{all monomials of deg } \leq m]$
  $\Rightarrow k(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T\mathbf{x}' + 1)^m \quad \text{implicitly represents all monimials o f degree up to } m$

Weight vs Function Space View

- Assume **Gaussian prior** on the weights: $\mathbf{w} \in \mathbb{R}^d \sim \mathcal{N}(0, \sigma_p^2 \mathbf{I})$
- This imply **Gaussian distribution on the predictions**
- Suppose we consider an arbitrary (finite) set of inputs $\mathbf{X} = \begin{pmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_n \end{pmatrix} \in \mathbb{R}^{n \times d}$
- The predictive distribution is given by:
  - $f \sim \mathcal{N}(0, \sigma_p^2 \mathbf{X}\mathbf{X}^T) \leftarrow \text{let } \mathbf{K}_{ij} = \mathbf{x}_i^T \mathbf{x}_j, \mathbf{K} \in (R)^{n \times n}$
  - where $f = [f_1, ..., f_n], f_i = \mathbf{x}_i^T \mathbf{w} \rightarrow f = \mathbf{X}\mathbf{w}$

Predictions in "function space"

- Suppose we're given data $\mathbf{X}$, $\mathbf{y}$, and want to predict $\mathbf{x}^*$
  - $\widetilde{\mathbf{X}} = \begin{pmatrix} \mathbf{X} \\ \mathbf{x}^* \end{pmatrix}, \quad \tilde{\mathbf{y}} = \begin{pmatrix} \mathbf{y} \\ y^* \end{pmatrix}, \quad \tilde{\mathbf{f}} = \begin{pmatrix} \mathbf{f} \\ f^* \end{pmatrix}$
    $\rightarrow \tilde{\mathbf{f}} = \widetilde{\mathbf{X}} \cdot \mathbf{w}, \tilde{\mathbf{y}} = \tilde{\mathbf{f}} + \tilde{\varepsilon}, \tilde{\varepsilon} \sim \mathcal{N}(0, \sigma_n^2 \mathbf{I}_{n+1})$
- $\rightarrow \tilde{\mathbf{y}} \sim \mathcal{N}(0, \widetilde{\mathbf{X}}\widetilde{\mathbf{X}}^T + \sigma_n^2 \mathbf{I})$,where $\widetilde{\mathbf{K}} = \widetilde{\mathbf{X}}\widetilde{\mathbf{X}}^T$
- $\rightarrow P(y^* | \mathbf{x}_{1:n}, \mathbf{y}_{1:n}) = \mathcal{N}(\mu_{\mathbf{x}^* | \mathbf{x}_{1:n}, \mathbf{y}_{1:n}}, \sigma^2_{\mathbf{x}^* | \mathbf{x}_{1:n}})$

Key Insight

- For prior $\mathbf{w} \sim \mathcal{N}(0, \mathbf{I})$, the predictive distribution over $\mathbf{f} = \mathbf{X}\mathbf{w}$ is Gaussian $\mathbf{f} \sim \mathcal{N}(0, \mathbf{X}\mathbf{X}^T) \equiv \mathcal{N}(0, \mathbf{K})$
- Thus, data points only enter as inner products!
- Can kernelize: $\mathbf{f} \sim \mathcal{N}(0, \mathbf{K})$, where $\mathbf{K}_{\mathbf{x},\mathbf{x}'} = \phi(\mathbf{x})^T \phi(\mathbf{x}') = \mathbf{k}(\mathbf{x}, \mathbf{x}')$
  e.g. poly. kernel $(1 + \mathbf{x}^T \mathbf{x}')^m$

What about infinite domains?

- The previous construction can be generalized to **infinitely large domains $\mathbf{X}$**
- The resulting random function is called a **Gaussian process**

Bayesian learning with Gaussian processes

- c.f. Rasmussen & Williams 2006
- $Likelihood : P(data|f) \qquad Posterior : P(f|data)$
- Predictive uncertainty + tractable inference

Gaussian Processes

- ∞-dimension Gaussian
- Gaussian process (GP) = normal distribution over functions
- Finite marginals are multivariate Gaussians

- Closed form formulae for Bayesian posterior update exist
- Parameterized by covariance function $k(\mathbf{x}, \mathbf{x}') = Cov(f(\mathbf{x}), f(\mathbf{x}'))$
- A **Gaussian Process (GP)** is an:
    - (infinite) set of random variables, indexed by some set $\mathbf{X}$
      i.e., there exists functions $\mu : X \rightarrow \mathbb{R} \quad k : X \times X \rightarrow \mathbb{R}$
      such that for all $A \subseteq X, \quad A = \{x_1, ..., x_m\}$
      it holds that $Y_A = [Y_{x_1}, ..., Y_{x_m}] \sim \mathcal{N}(\mu_A, \mathbf{K}_{AA})$
      where,
      $$\mathbf{K}_{AA} = \begin{pmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_m) \\ \vdots & & \vdots & \\ k(x_m, x_1) & k(x_m, x_2) & \dots & k(x_m, x_m) \end{pmatrix}, \quad \mu_A = \begin{pmatrix} \mu(x_1) \\ \vdots \\ \mu(x_m) \end{pmatrix}$$
      k is called **covariance (kernel)** function
      μ is called **mean** function