SME3006 Machine Learning – 2025 Fall

Uncertainty Quantification and Gaussian Process

INHA UNIVERSITY

Overview

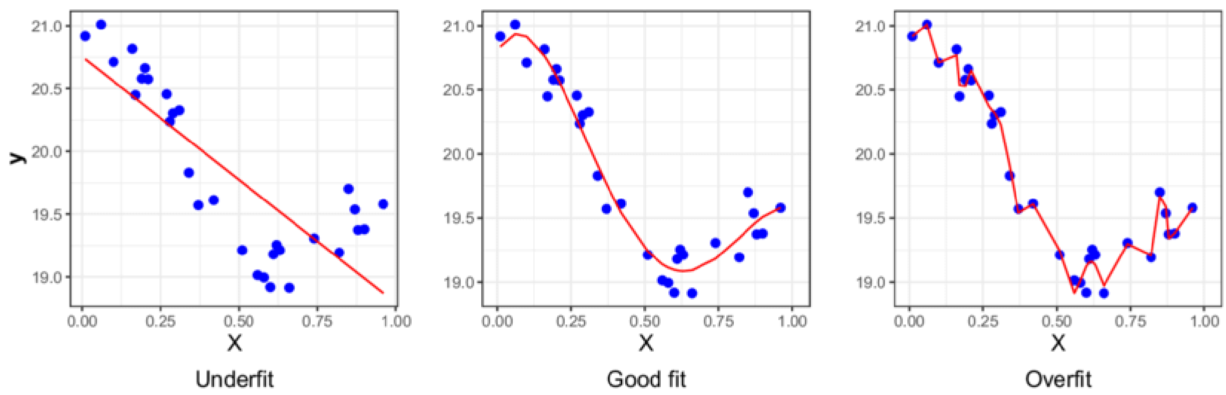
> Introduction to model uncertainty
  - why uncertainty
  - uncertainty types

> How to quantify the uncertainty
  - Bayesian approach (probabilistic uncertainty)
  - Bayesian linear regression
  - Gaussian process
  - Ensemble

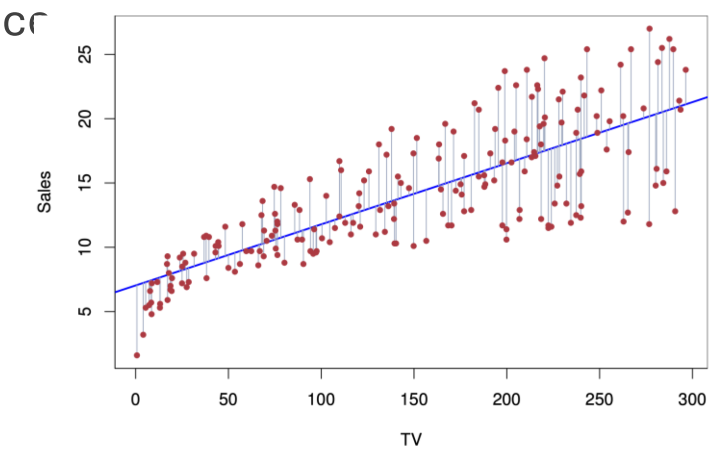
> Applications
  - exploration, active learning
  - Bayesian optimization

Regression problem

> Fitting line with reducing the mean error
  - more complex model → low mean error, but low generalizability

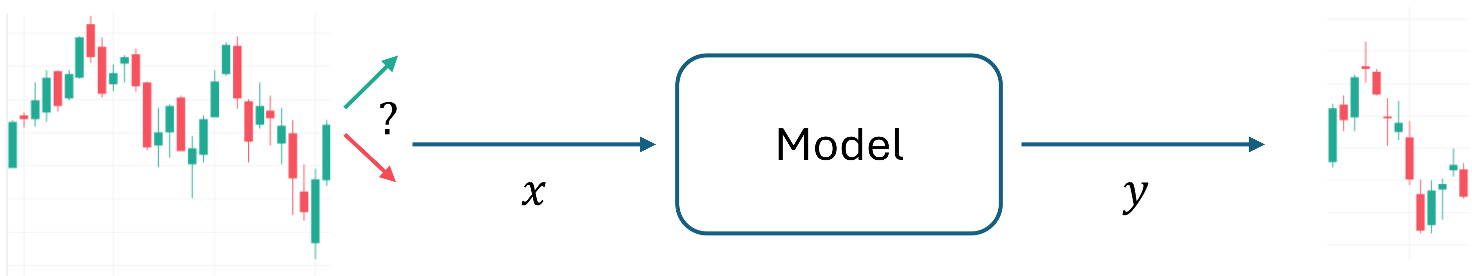


> According to data, there is a bound we can minimize the error
  - due to noise, disturbance, ignoranc



Machine learning

> Model: given an input, produce an output

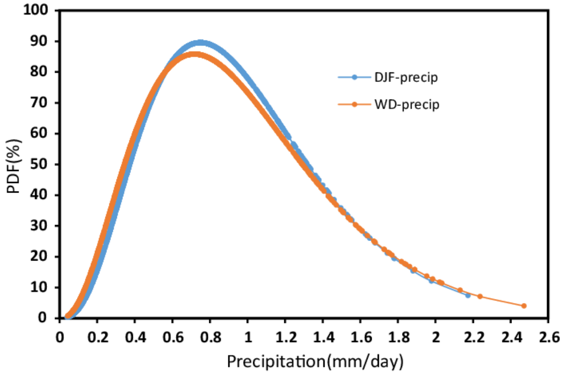


- how certain we are about the model? can you believe the model's output?
- error can be computed after we observe the result
- we should predict the uncertainty of the model before observing the result
  - predicting the stock market
  - autonomous driving
  - all practical and safety-related models

Uncertainty

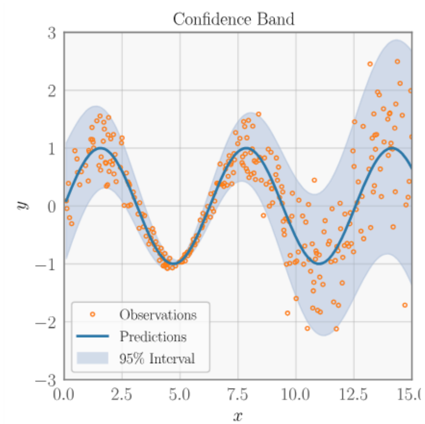
> Quantifying the certainty / uncertainty?
  - Assuming that you are going to a field trip
  - measure of uncertainty would be the probability
    - what does it mean?

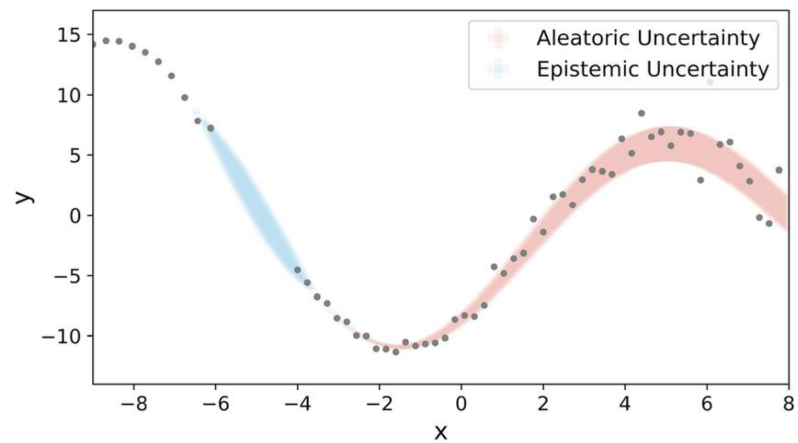
  - How about continuous variables?
    - the probability of rain amount 1mm? 1.1mm? 1-1.1mm?



Uncertainty

> Desirable prediction
  - underconfident / overconfident / well calibrated



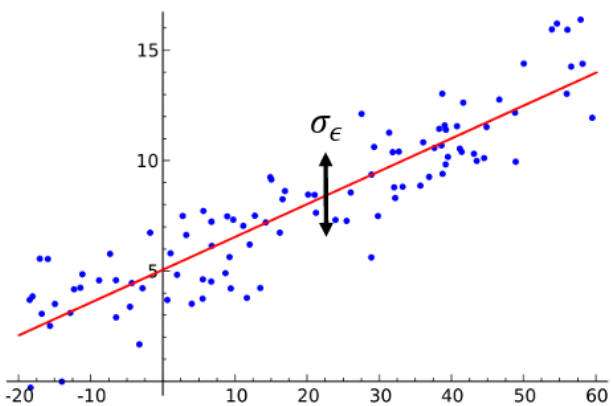


> Types of uncertainties
  - aleatoric uncertainty (random noise)
    - Inherent to data. Cannot be reduced by adding more information
  - epistemic uncertainty (lack of knowledge)
    - By the model. Can be reduced by adding more information

Uncertainty

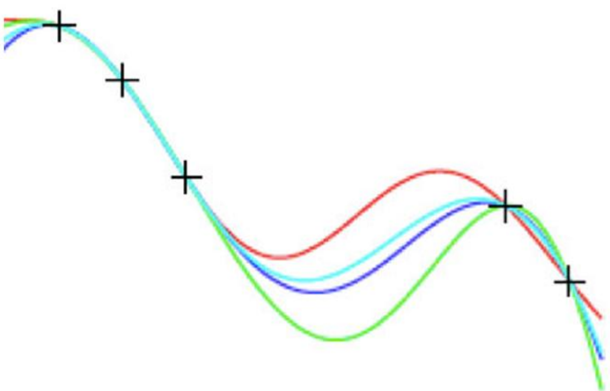
> Aleatoric uncertainty
  - most models include aleatoric parameters that capture mismatch between the model predictions and the labels
  - even with infinity data, there is ambiguity inherent in data itself

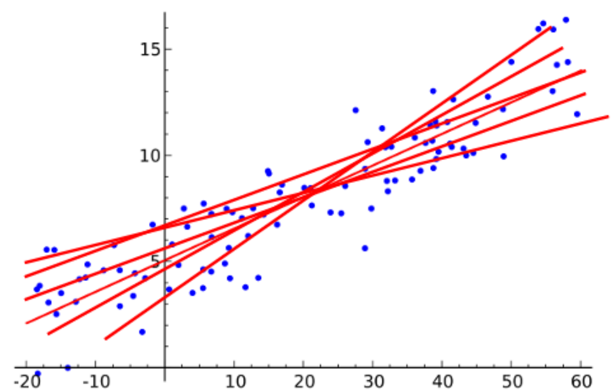
  - linear regression
    - $y = \beta^\top x + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$
    - σ_ϵ^2 estimates the amount of noise in the labels
    - it is estimated as the variance of the training data residuals



Uncertainty

> Epistemic uncertainty
  - uncertainty due to the finite amount of training data
  - large number of possible models can explain a dataset
    - uncertain which model parameters to choose to predict with
    - affect how we predict with new test points
  - in a non-stationary world, target is continually changing and there is always epistemic uncertainty





Uncertainty

- Global uncertainty, related to aleatoric or epistemic?
  - Are the measured features sufficient to make accurate predictions?
  - Is there selection bias in the features?
  - Can the model class represent a good approximation of the true decision boundary?
  - Is there measurement noise in the features?
  - Do we have enough training data so that a learning algorithm can find that good approximation?
  - Are the labels on the training data accurate, noisy, or biased?
  - Are there missing values in the features?
  - Can the learning algorithm find that good approximation?
  - Is the optimal classifier changing over time? (data shift)

Probabilistic modeling

> If we use the mathematics of probability theory to express all forms of uncertainty and noise associated with our model

- then inverse probability (i.e. Bayes rules) allows us to infer unknown quantities, adapt our models, make predictions and learn from data

> Bayes rule

$$p(h|\mathcal{D}) = \frac{p(\mathcal{D}|h)p(h)}{p(\mathcal{D})} \qquad h: hypothesis \quad \mathcal{D}: dataset$$

- it tells us how to do inference about hypotheses from data
- there always exists some underlying process that generated data
- in Bayesian probabilistic modeling we set underlying process explicit (find the distribution that generated data)

Probabilistic modeling

> Dealing with the probability
  - Bayes rule

likelihood

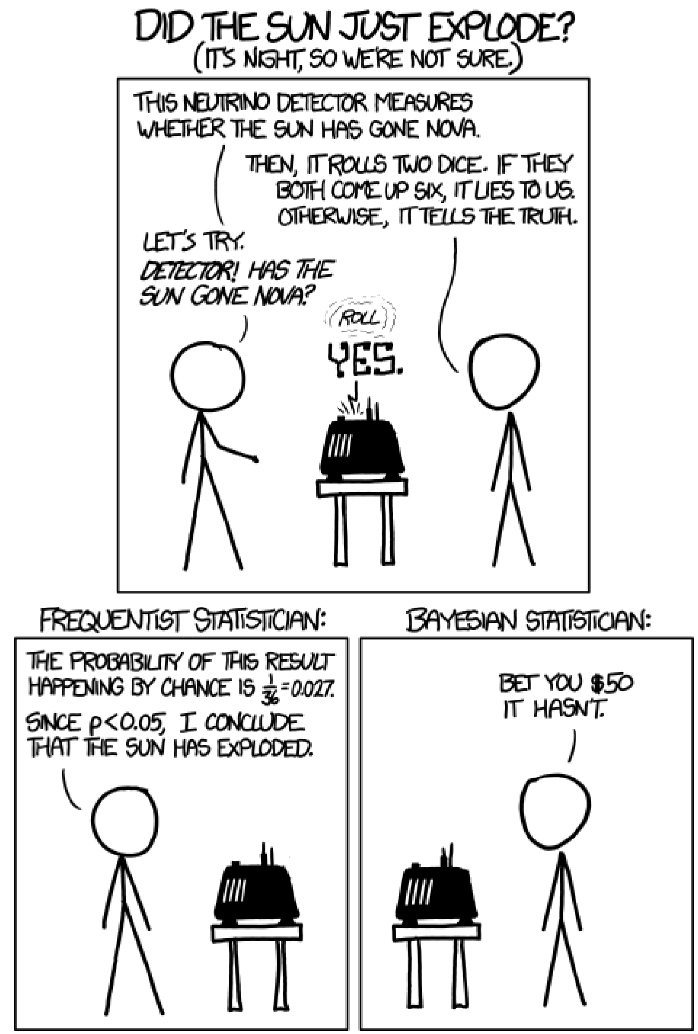
prior

$$p(B|A) = \frac{p(A|B)p(B)}{p(A)}$$

posterior

marginal likelihood

- $p(lie)$ = 1/36
- $p(exploded|yes)$

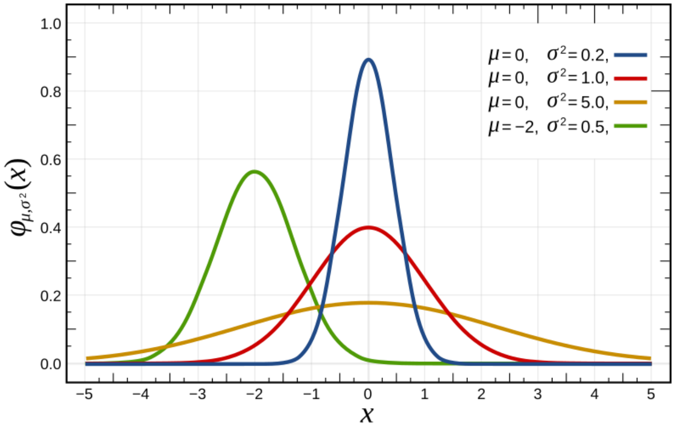


Probabilistic modeling

> ML data underlying process
  - e.g. cat vs dog classification
    - there exist some underlying rules we don't know
    - such as if has pointy ears then cat
    - we observe pairs and want to infer the underlying rules

  - e.g. Gaussian density estimation
    - Generated data follows Gaussian distribution

$$x_n \sim \mathcal{N}(\mu, \sigma^2), \qquad \sigma = 1$$

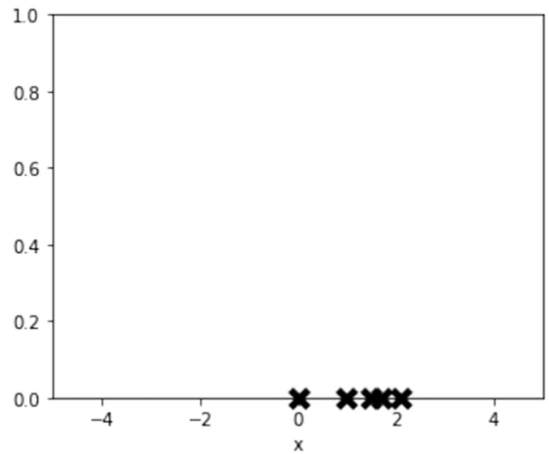


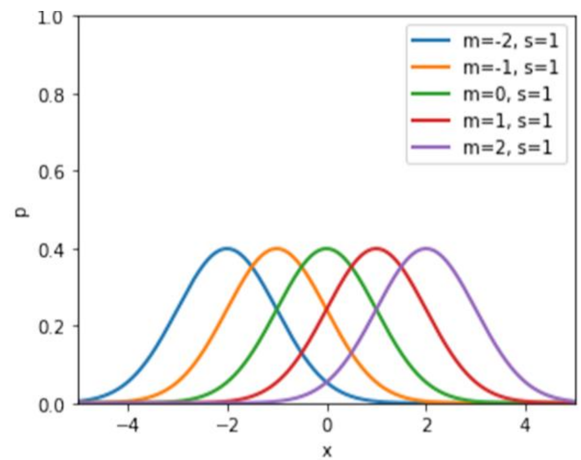
    - Gaussian density with mean μ and σ^2 is

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

Probabilistic modeling

> Gaussian density estimation example
  - we observed 5 points and want to infer μ ($\sigma = 1$ and we know it)
  - let's say we have 5 candidates Gaussians
  - what is the probability that $\mu = 1$ generated the data?

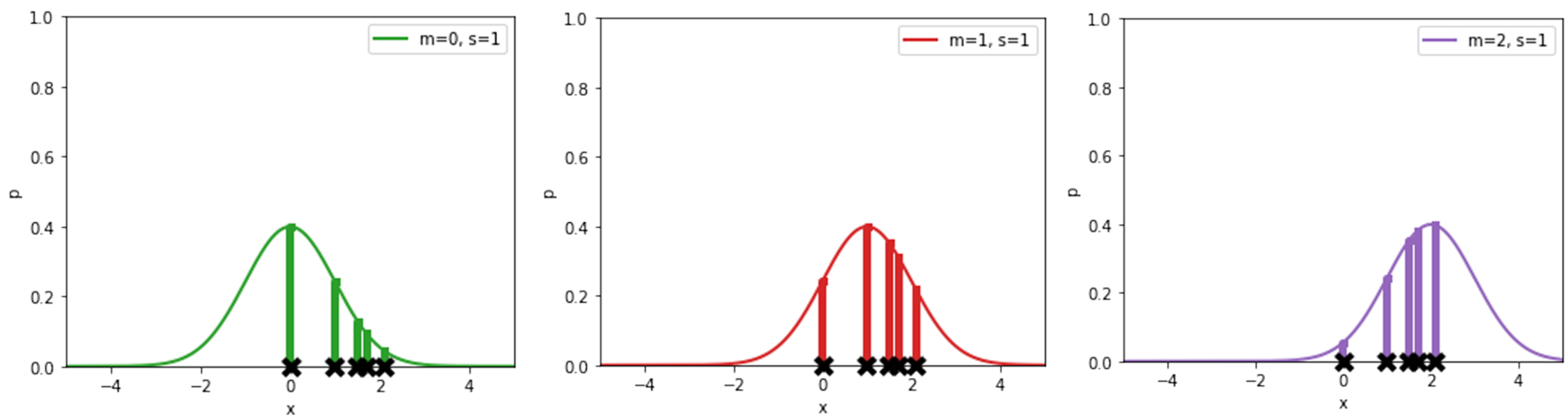




- $p(h|\mathcal{D}) = \frac{p(\mathcal{D}|h)p(h)}{p(\mathcal{D})} \quad \rightarrow \quad p(\mu = 1|\mathcal{D}) = \frac{p(\mathcal{D}|\mu=1)p(\mu=1)}{p(\mathcal{D})}$

Probabilistic modeling

> Gaussian density estimation example
  - $p(\mu = 1|\mathcal{D}) = \frac{p(\mathcal{D}|\mu=1)p(\mu=1)}{p(\mathcal{D})}$
  - Likelihood



  - prior: we believe data is equally likely to have come from 5 Gaussians
  - marginal likelihood: normalizer (sum of likelihood)

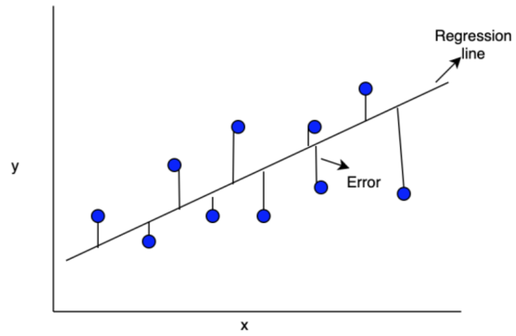
Bayesian linear regression

> Recap: linear regression
  - given a training set of inputs and targets $\{(x_i, y_i)\}_{i=1}^N$
  - linear model: $\hat{y}_i = w^\top x_i$
  - squared error loss: $L = \frac{1}{2}\sum(y_i - \hat{y}_i)^2$
  - solution 1: solve analytically by setting the gradient to 0

$$w = (X^\top X)^{-1} X^\top y$$

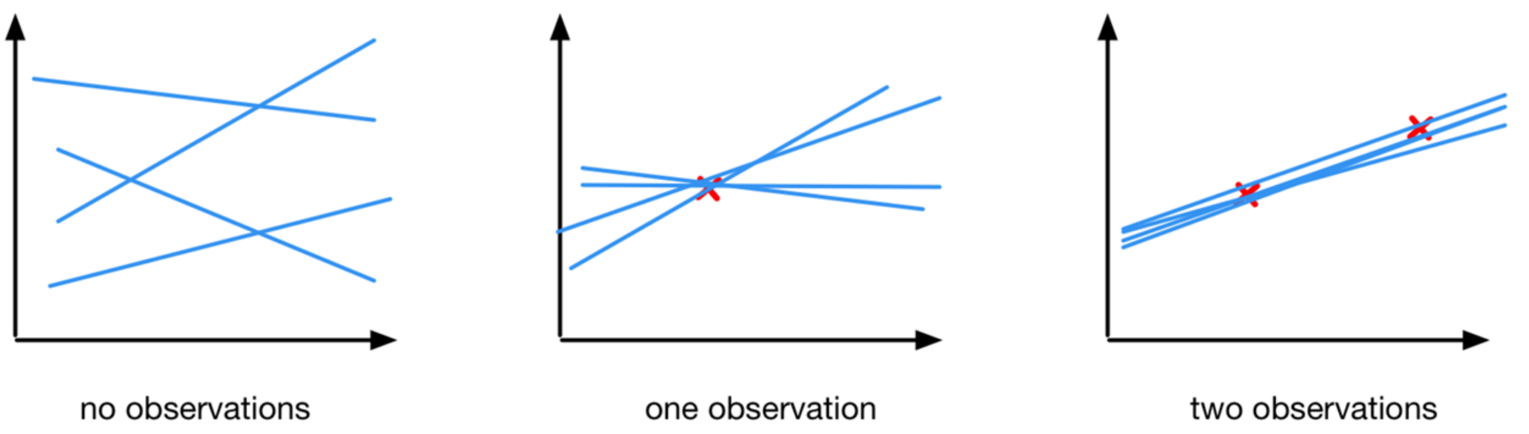
  - solution 2: solve approximately using gradient descent

$$w \leftarrow (1-\alpha)w - \alpha X^\top (y - \hat{y})$$



Bayesian linear regression

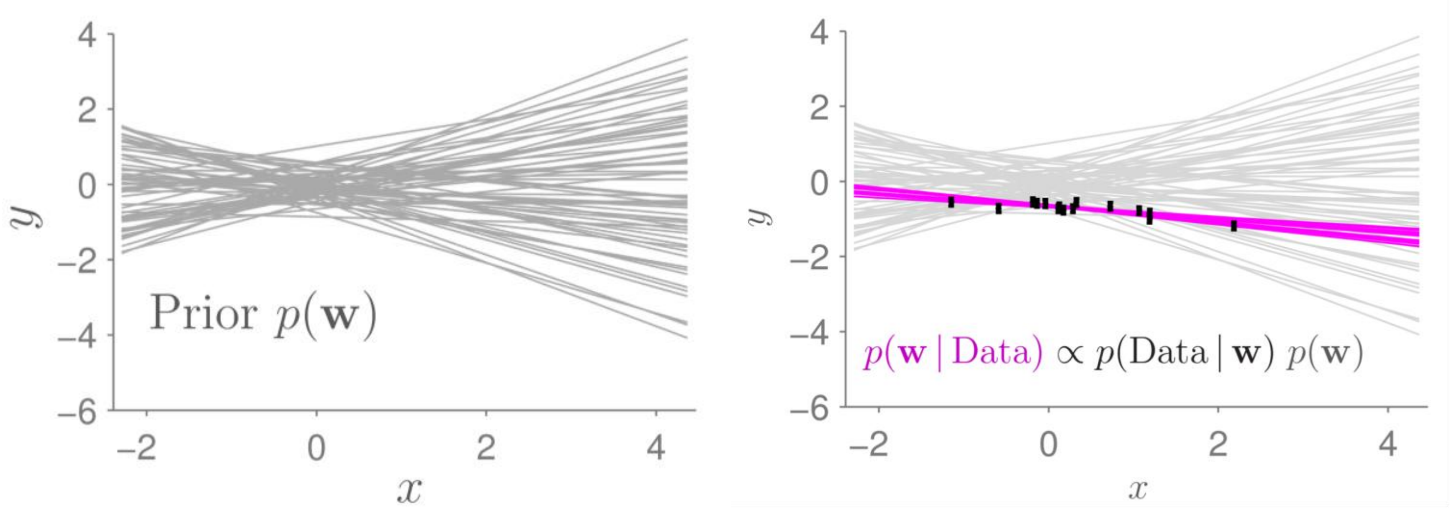
> Full Bayesian inference
  - compute posterior using Bayes' rule:
  $$p(w|\mathcal{D}) \propto p(w)p(\mathcal{D}|w)$$
  - make prediction by averaging over all likely explanations under the posterior distribution
  $$p(y|x,\mathcal{D}) = \int p(w|\mathcal{D})p(y|x,w)dw$$
  - we can quantify the model uncertainty



Bayesian linear regression

> Bayes' rule
  - $p(w|\mathcal{D}) = p(w|X, y, \sigma^2) = \frac{p(y|X,w,\sigma^2)p(w)}{p(y|X,\sigma^2)} \propto p(y|X, w, \sigma^2)p(w)$
  - we want to infer w given (X, y, σ^2) assuming that σ^2 is fixed/known

> We also assume that noise follows the Gaussian distribution
  - $y_i = w^\top x_i + \epsilon_i, \quad \epsilon \sim \mathcal{N}(0, \sigma^2)$
  - $y = Xw + \epsilon, \ \epsilon \sim \mathcal{N}(0, \sigma^2 I)$, likelihood $p(y|X, w, \sigma^2) = \mathcal{N}(Xw, \sigma^2 I)$
  - we set prior for parameter w as Gaussian $p(w) = \mathcal{N}(0, \sigma_w^2 I)$



Bayesian linear regression

> Posterior
  - $p(w|X, y, \sigma^2) \propto p(y|X, w, \sigma^2)p(w) \propto \mathcal{N}(Xw, \sigma^2)\mathcal{N}(0, \sigma_w^2 I) = \mathcal{N}(\mu, \Sigma^2)$ (Gaussian × Gaussian = Gaussian)
  - $\Sigma = (\sigma_w^{-2} I + \sigma^{-2} X^\top X)^{-1}$, $\mu = \sigma^{-2} \Sigma X^\top y$
  - Derivation:
    - $$p(y|X, w, \sigma^2) \propto \exp\left(-\frac{1}{2\sigma^2}(y - Xw)^\top (y - Xw)\right)$$
      $$= \exp\left(-\frac{1}{2\sigma^2}(w^\top X^\top X w - 2w^\top X^\top y + constant)\right)$$
    - $$p(w) \propto \exp\left(-\frac{1}{2\sigma_w^2} w^\top w\right)$$
    - $$p(y|X, w, \sigma^2)p(w) \propto \exp\left(-\frac{1}{2} w^\top \underbrace{(\sigma^{-2} X^\top X + \sigma_w^{-2} I)}_{A} w + w^\top \underbrace{(\sigma^{-2} X^\top y)}_{b} + constant\right)$$
      $$\propto \exp(-\frac{1}{2}(w - A^{-1}b)^\top A(w - A^{-1}b))$$
    - $\Sigma = \mathrm{A}^{-1},\ \mu = A^{-1}b$

Bayesian linear regression

> Posterior predictive

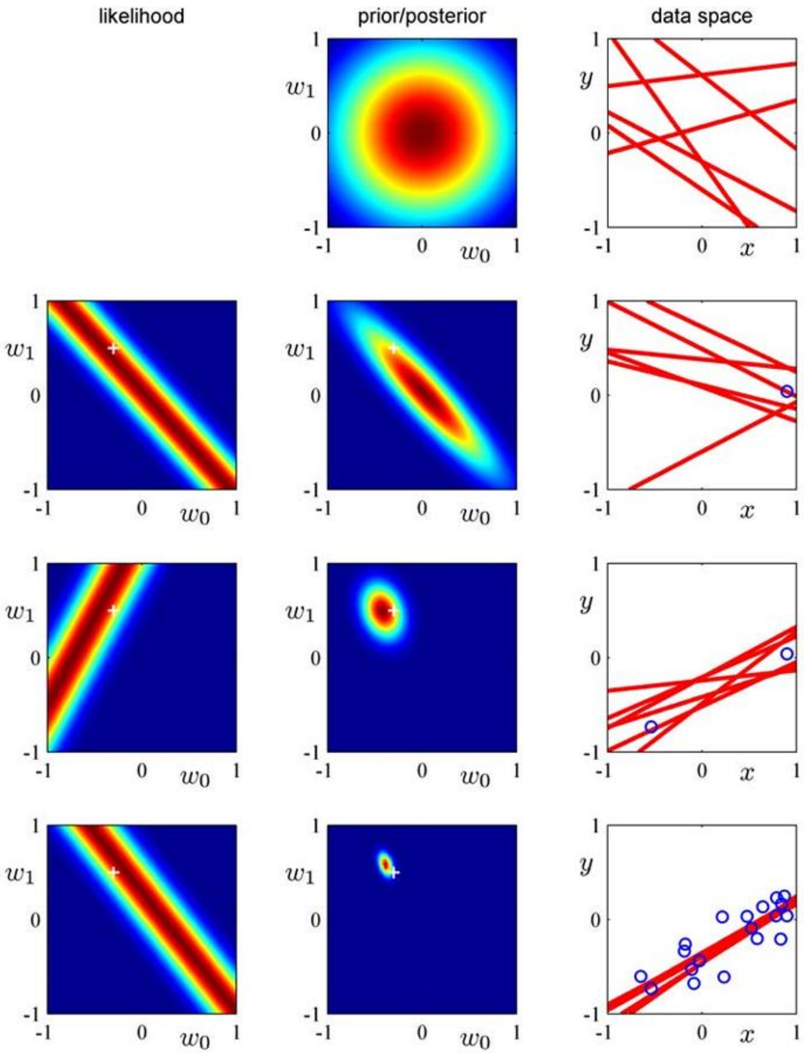
$$p(y|x, \mathcal{D}) = \int p(w|\mathcal{D}) p(y|x, w) dw$$

- with new input x_* $\rightarrow$ $y_* = w^\top x_* + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma^2)$
- $p(y_*|x_*, X, y, \sigma^2) = \int p(w|X, y, \sigma^2) p(y_*|x_*, w, \sigma^2) dw$
  $= \mathcal{N}(\mu^\top x_*, \underbrace{\sigma^2}_{\text{aleatoric uncertainty}} + \underbrace{x_*^\top \Sigma x_*}_{\text{epistemic uncertainty}})$ derivation?

aleatoric uncertainty epistemic uncertainty

  - we can't reduce aleatoric uncertainty
  - epistemic uncertainty decreases as more data are collected proof?

Bayesian linear regression

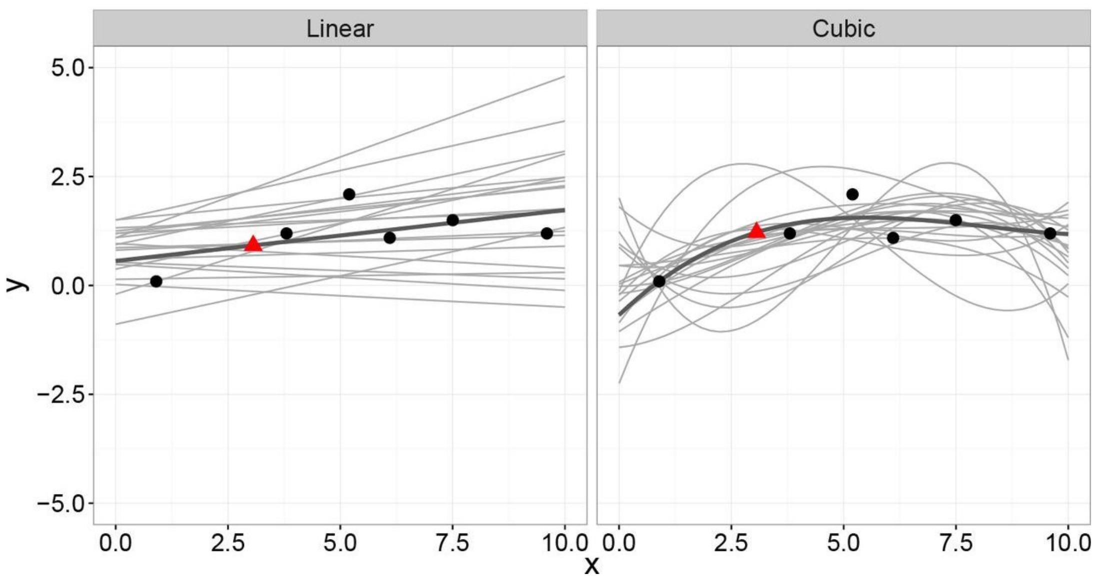


Gaussian process

> Instead of using linear functions, we want to use nonlinear functions

- BLR prior: $y = w^\top x + \epsilon,\ \ w \sim \mathcal{N}(0, \sigma_w^2 I)$
  Linear equations with parameters given by Gaussian random weights

- general prior: $y = w^\top \psi(x) + \epsilon, w \sim \mathcal{N}(0, \sigma_w^2 I)$
  Nonlinear equations with parameters given by Gaussian random weights
  $\phi(x)$ is a feature mapping function (basis functions of the feature space)



Gaussian process

> Instead of using linear functions, we want to use nonlinear functions

- BLR prior: $y = w^\top x + \epsilon,\ \ w \sim \mathcal{N}(0, \sigma_w^2 I)$

  posterior: $p(w|X, y, \sigma^2) \propto \mathcal{N}(\mu, \Sigma^2)$
  $\Sigma = (\sigma_w^{-2} I + \sigma^{-2} X^\top X)^{-1}, \mu = \sigma^{-2} \Sigma X^\top y$

  prediction distribution: $\mathcal{N}(x_*^\top \mu,\ \ \sigma^2 + x_*^\top \Sigma x_*)$

- general prior: $y = w^\top \psi(x) + \epsilon, w \sim \mathcal{N}(0, \sigma_w^2 I)$

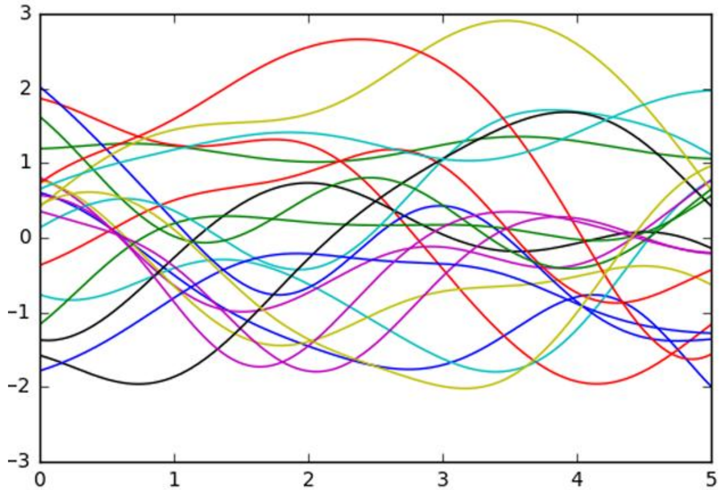
  posterior: $p(w|X, y, \sigma^2) \propto \mathcal{N}(\mu, \Sigma^2)$
  $\Sigma = (\sigma_w^{-2} I + \sigma^{-2} \Psi^\top \Psi)^{-1}, \mu = \sigma^{-2} \Sigma \Psi^\top y$

  prediction distribution: $\mathcal{N}(\psi(x_*)^\top \mu,\ \ \sigma^2 + \psi(x_*)^\top \Sigma \psi(x_*))$

- with some math, prediction distribution is represented by kernel

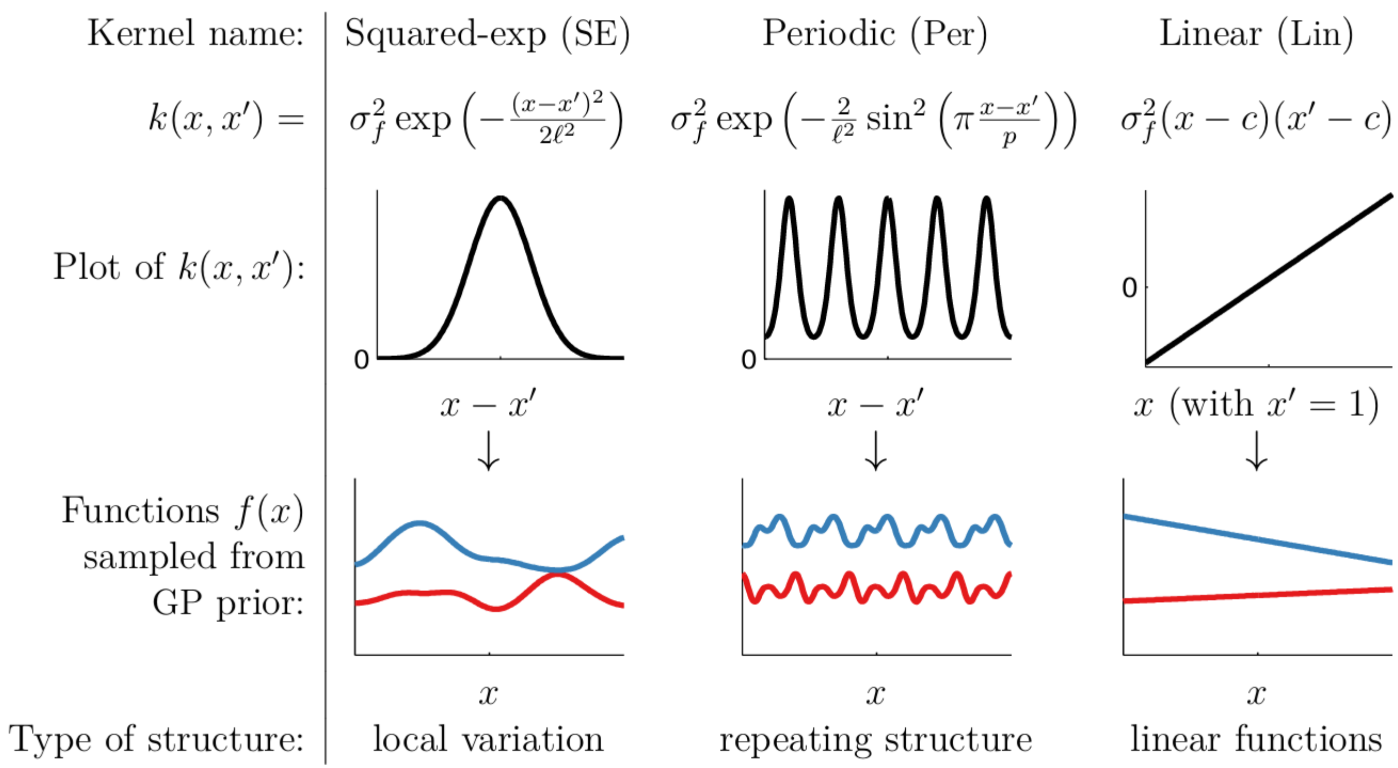
Gaussian process

> Recall: kernel $K(x_i, x_j) = \psi(x_i)^\top \psi(x_j)$
  - we don't have to know the $\psi(x_i)$, we only need $K(x_i, x_j)$
  - Kernel implicitly defines an infinite-dimensional mapping function
  - example: RBF (Gaussian): $K(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$
    - for $d = 1$, $\psi(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right)\left[1, \frac{x}{\sigma\sqrt{1!}}, \frac{x^2}{\sigma\sqrt{2!}}, \frac{x^3}{\sigma\sqrt{3!}}, \ldots\right]^\top$
    - this is an infinite vector, nobody actually uses this value
    - generated function (prior) $f(x) = w^\top \psi(x)$



Gaussian process

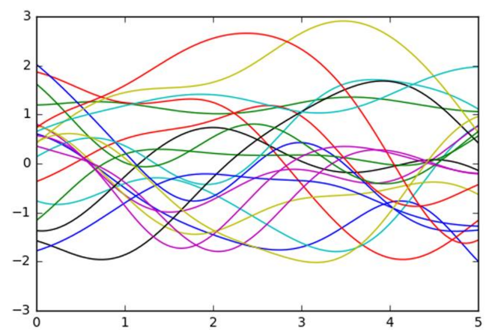
- Recall: kernel $K(x_i, x_j) = \psi(x_i)^\top \psi(x_j)$
  - generated function (prior) $f(x) = w^\top \psi(x)$



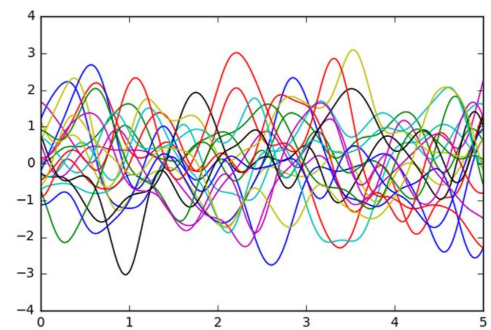
Gaussian process

> Prior - RBF kernel with hyperparameters σ

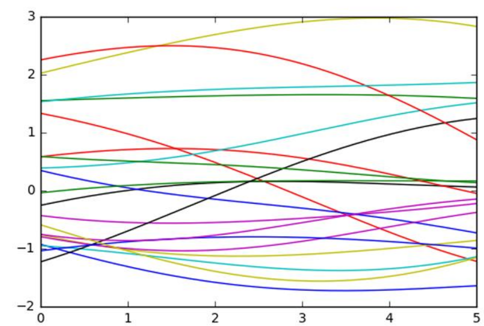
$$k(x,x') = \exp\left(-\frac{1}{2}(x-x')^2\right)$$



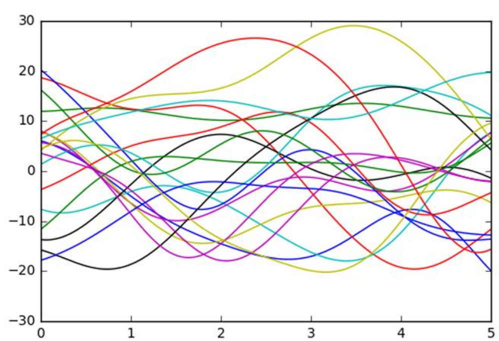
$$k(x,x') = \exp\left(-\frac{1}{2}\frac{(x-x')^2}{0.25^2}\right)$$



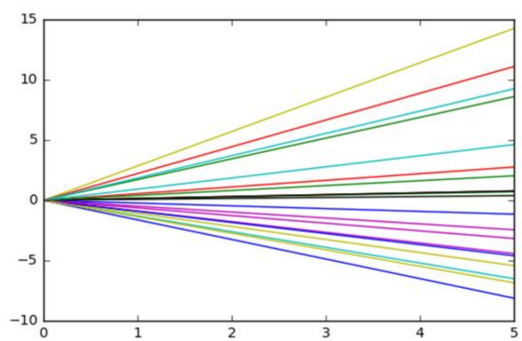
$$k(x,x') = \exp\left(-\frac{1}{2}\frac{(x-x')^2}{4^2}\right)$$



$$k(x,x') = 100\exp\left(-\frac{1}{2}(x-x')^2\right)$$

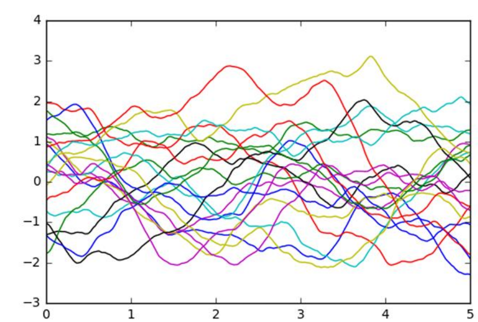


$$k(x,x') = x^\top x'$$



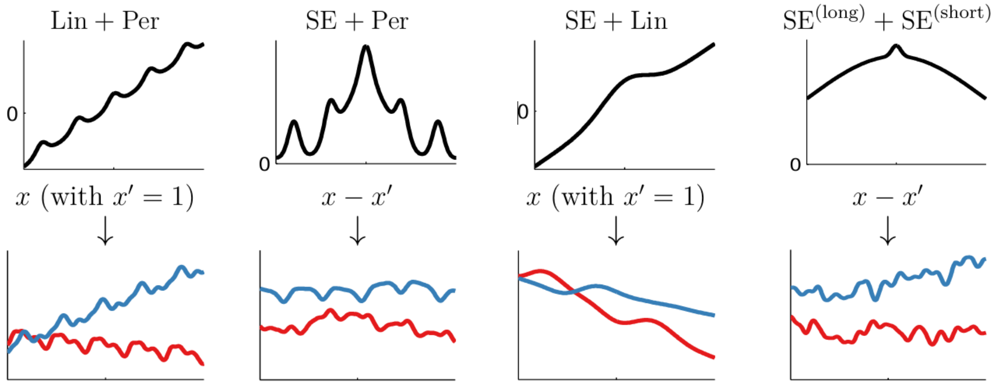
Matern 3/2

$$k(x,x') \sim (1+|x-x'|)\exp\left(-|x-x'|\right)$$

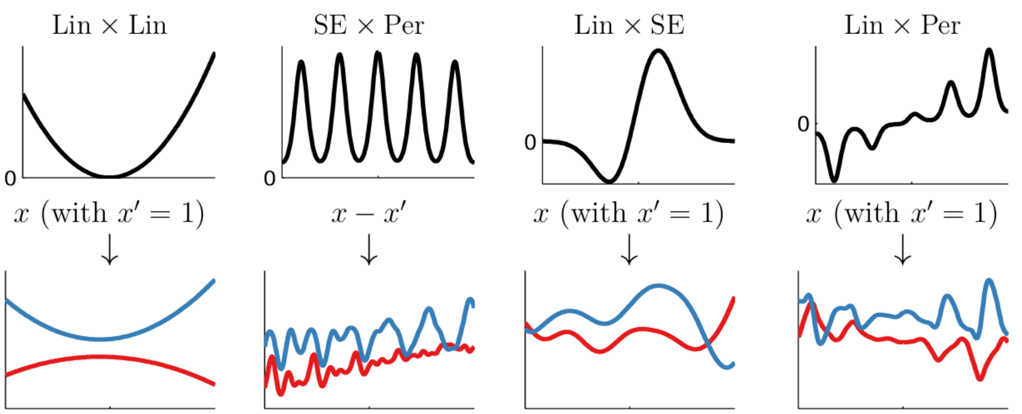


Gaussian process

- recall: Creating more complicated kernels
  - $K(x_i, x_j) = K_1(x_i, x_j) + K_2(x_i, x_j)$

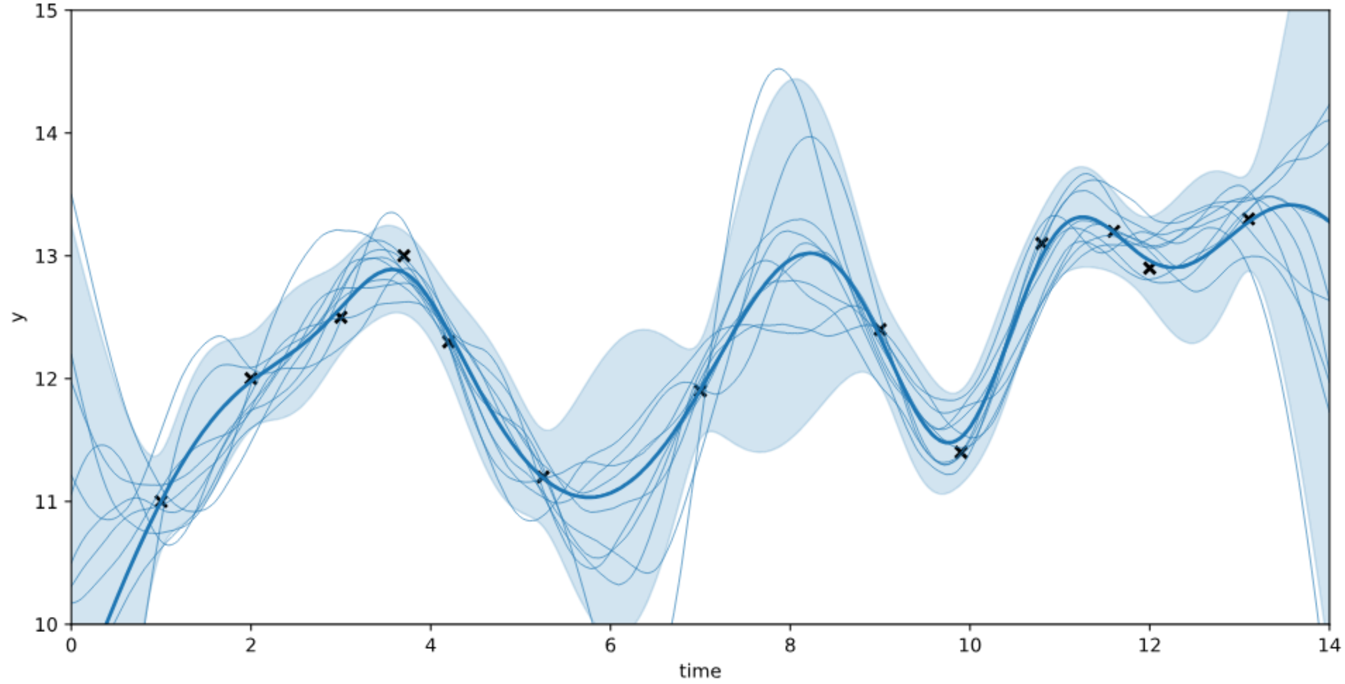


  - $K(x_i, x_j) = K_1(x_i, x_j) K_2(x_i, x_j)$



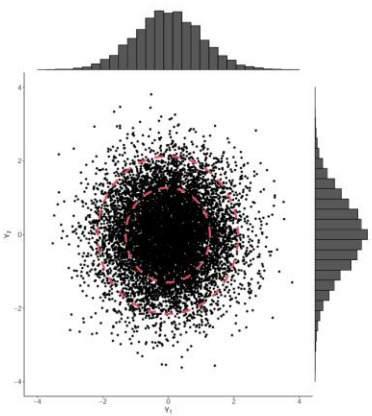
Gaussian process

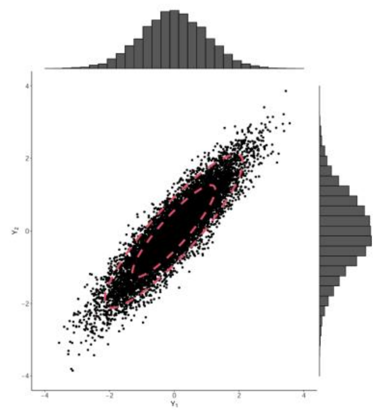
> A GP is a collection of random variables (random function values), any finite number of which have a joint Gaussian distribution
  - the kernel defines the covariance between function values
  - given training data, GP produces a predictive distribution for new inputs

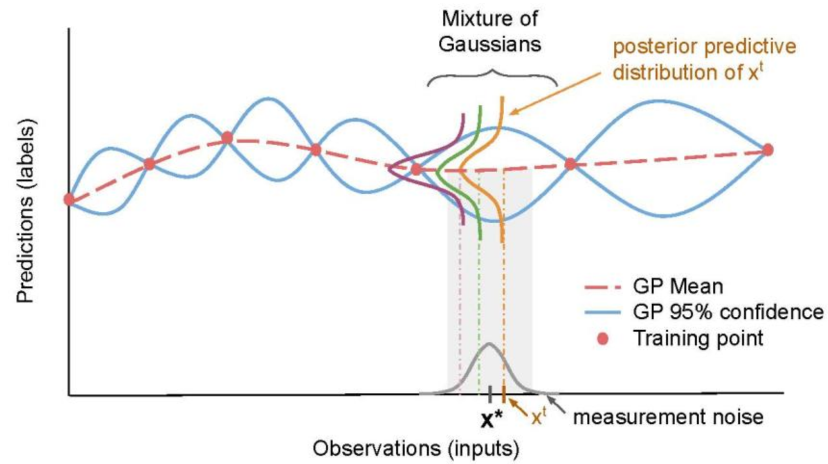


Gaussian process

> It follows a multivariate Gaussian distributions
  - multivariate = two or more random variables







- In theory: GP = infinite-dimensional Gaussian distribution prior over functions (all possible x)

- In practice: choose a finite set of inputs corresponding outputs follow a multivariate Gaussian

Gaussian process

- Tuning GP hyperparameters
  - maximizing the log-likelihood of y after integrating out possible $f(\cdot)$'s

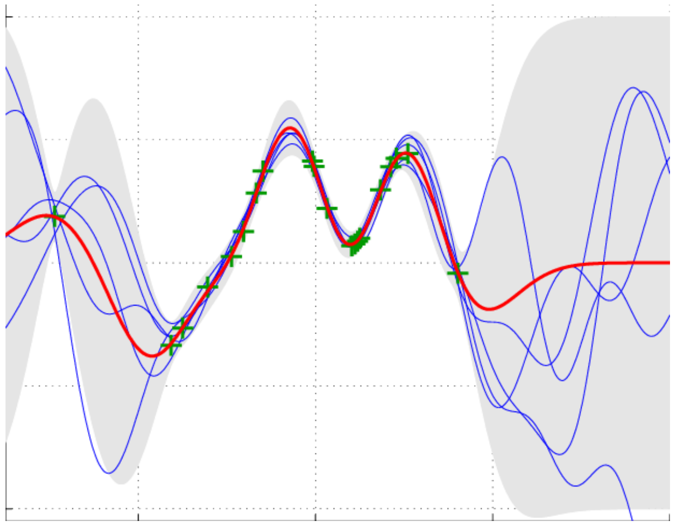
$$\log p(y|X, w, \sigma^2) = \log \int p(y|f, \sigma^2) p(f|X, w) df = \log \mathcal{N}(0, K(X, X))$$

  - it is differentiable, thus optimizing is convenient
- Disadvantages
  - Inverting kernel matrix takes $O(n^3)$ operations
    - Nowadays, use parallel computation (e.g. GPyTorch) for high-dimensional data
  - Designing the kernel may require considerable work
  - Mainly applied to regression tasks
  - Some kernel may break down in high dimensions (e.g., images)

Ensemble methods

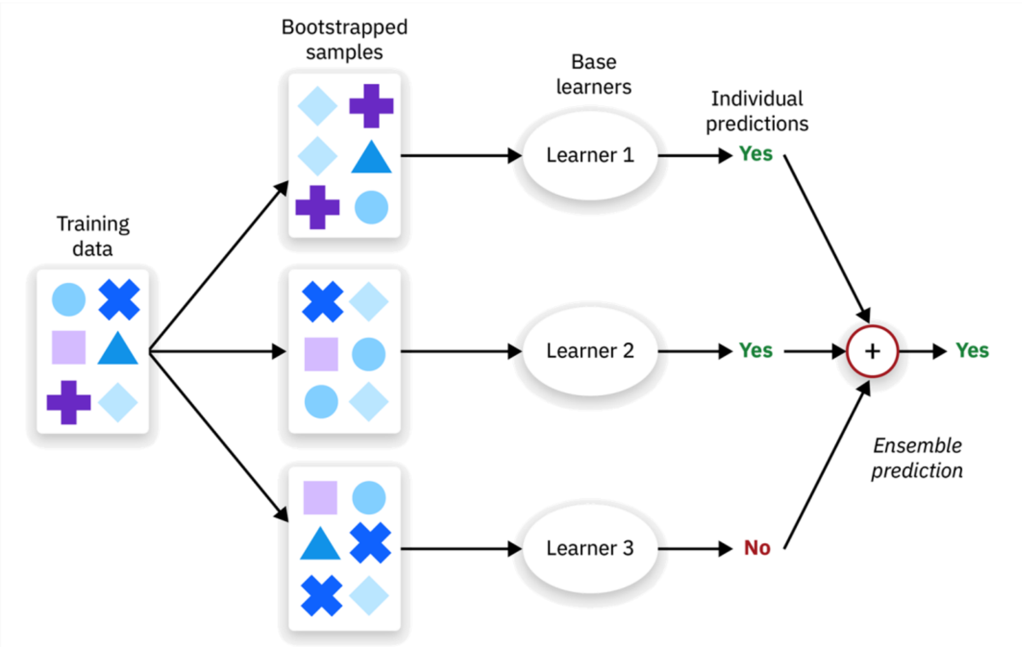
> There are many machine learning algorithms
  - Dimension reduction, PCA, SVM, clustering, decision trees …
  - how can we measure an uncertainty of each machine learning model?

> Idea: train an ensemble of models, measure the degree of disagreement
  - high disagreement → high epistemic uncertainty
    - under the assumption that the learning algorithms would converge to a unique answer given infinite data



Ensemble methods

- Recall: bagging
  - from a single dataset $\mathcal{D}$, generating m new datasets, each by sampling n training examples from $\mathcal{D}$
  - predict with models trained on each of these datasets
- To measure the uncertainty, use disagreement (variance of prediction)
  - if the model is stochastic, then disagreement includes aleatoric uncertainty



Bayesian optimization

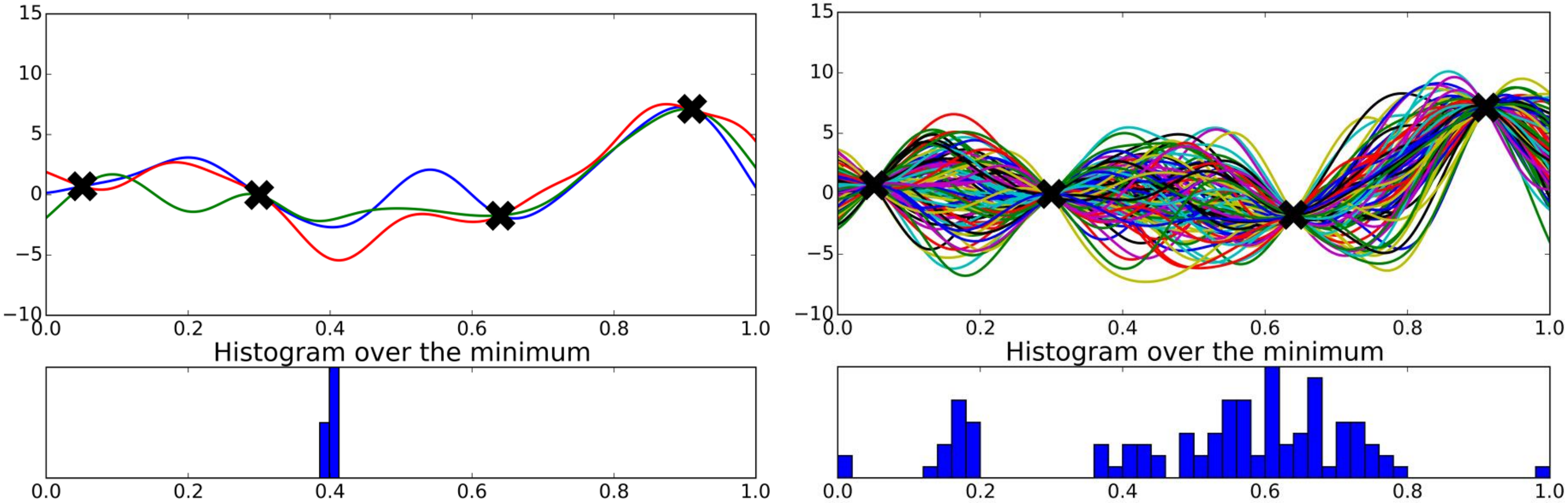
> Bayesian optimization
  - for expensive black-box optimizations, the goal is to find the next point to evaluate

> Active learning
  - learn efficiently by choosing the most informative samples

> Role of uncertainty
  - uncertainty highlights regions with potential rewards
  - we also consider the function value (cost)
  - acquisition function = uncertainty + cost

Bayesian optimization

> Where is the minimum of f?
  - where should the take the next evaluation?

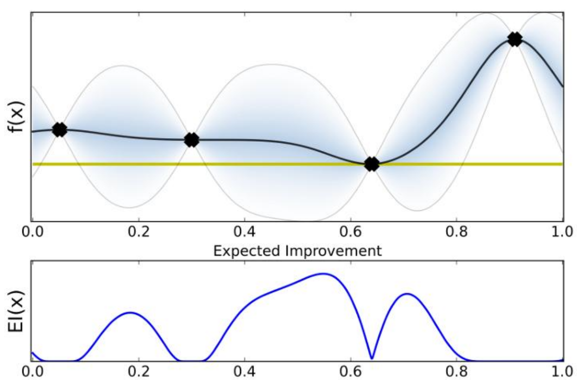


  - acquisition function: combines the predicted mean (cost) and uncertainty
  - There are many options: expected improvement, upper confidence bound, Thompson sampling, ...

Bayesian optimization

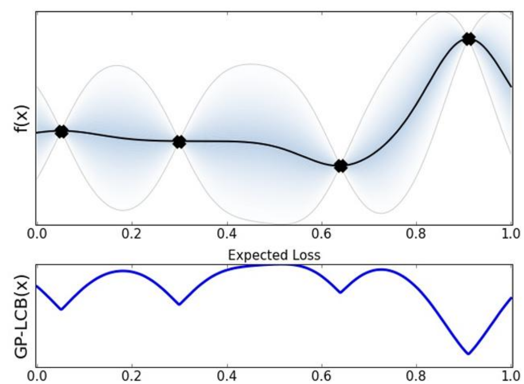
> Acquisition function
  - There are many options: expected improvement, upper confidence bound, Thompson sampling, ...

$$\alpha_{EI}(\mathbf{x};\theta,\mathcal{D}) = \int_y \max(0, y_{best} - y)p(y|\mathbf{x};\theta,\mathcal{D})dy$$



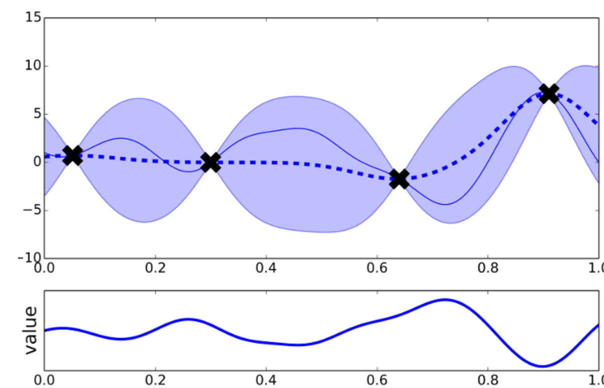
Expected improvement

$$\alpha_{LCB}(\mathbf{x};\theta,\mathcal{D}) = -\mu(\mathbf{x};\theta,\mathcal{D}) + \beta_t\sigma(\mathbf{x};\theta,\mathcal{D})$$



Confidence bound

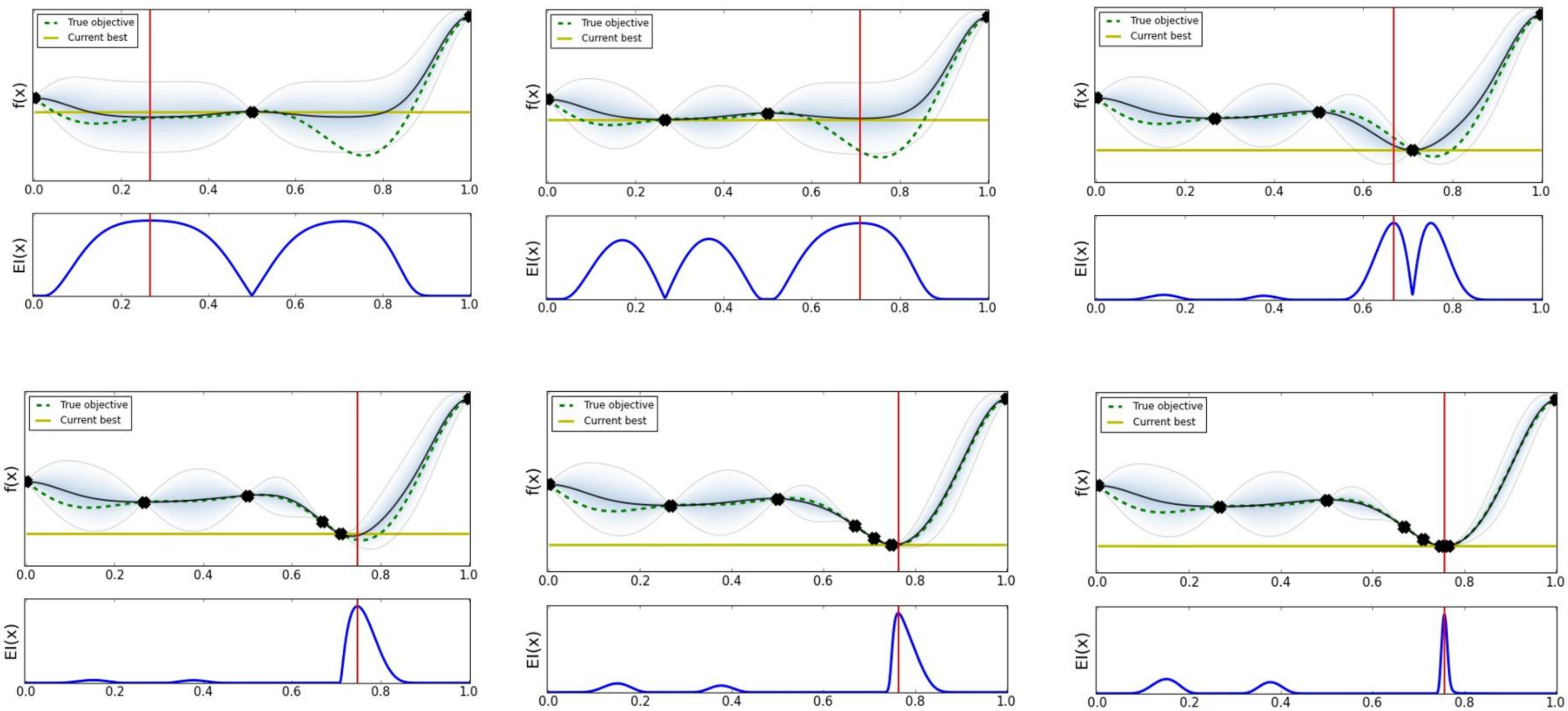
$$\alpha_{THOMSON}(\mathbf{x};\theta,\mathcal{D}) = g(\mathbf{x})$$
$$g(\mathbf{x}) \text{ is sampled form } \mathcal{GP}(\mu(x), k(x,x'))$$



Thompson sampling

Bayesian optimization

> Procedure



More on uncertainty quantification

> Many applications: selective prediction, active learning, system integration, uncertainty aware learning

> Good to investigate area: uncertainty calibration, propagation, out-of-distribution (OOD) detection, anomaly detection, conformal prediction

Reference

- Bayesian linear regression
  - https://www.su.se/polopoly_fs/1.484397.1581424529!/menu/standard/file/GuestLectureKTH2020%281%29.pdf
  - https://www.cs.toronto.edu/~rgrosse/courses/csc411_f18/slides/lec19-slides.pdf
- Uncertainty quantification
  - https://web.engr.oregonstate.edu/~tgd/talks/dietterich-uncertainty-quantification-in-machine-learning-final.pdf
  - https://www.cs.ox.ac.uk/people/yarin.gal/website/bdl101/MLSS_2019_BDL_1.pdf
- Gaussian Process
  - https://www.cs.toronto.edu/~rgrosse/courses/csc411_f18/slides/lec20-slides.pdf
  - https://gpss.cc/gpss20/slides/Wilkinson2020.pdf
  - https://www.comp.nus.edu.sg/~scarlett/gp_slides/GP_Slides00_Background.pdf
  - https://www.youtube.com/watch?v=UBDgSHPxVME
- Bayesian optimization
  - https://gpss.cc/gpmc17/slides/LancasterMasterclass_1.pdf