The Unscented Transform

Bayesian Mixer

John Whitamore

14-May-26

What we'll discuss

- **What does *Bayesian* mean, actually?**
- **Sigma points and the Unscented Transform**
- **Rings of Saturn**
- **Gaussian quadrature**
- **Finite elements analysis**
- **ReLU-based deep learning**

What does *Bayesian* mean?

A machine for learning

Inference is **inductive**:

- Start with prior beliefs $p(H)$
- Evaluate data using the likelihood density $p(D|H)$
- Update beliefs . . . *obtain posterior beliefs using Bayes' Law*

And then, if useful:

- Observe more data
- Update beliefs again
- etc

cf. teaching, propaganda

So what *does* Bayesian mean, actually?

1. Inductive inference
2. Requirement for a prior distribution over parameter values:
   - To handle uncertainty, we need a distribution over parameter values.
   - We need the distribution as a **prior**, before we see the first data point.
3. Marginalisation instead of hypothesis testing:
   - Instead of representing a hypothesis by a single parameter estimate ...
   - ... and rejecting / tentatively accepting the null hypothesis
   - ... we weight the parameter values according to their posterior probability ...
   - ... and use the weighted values to make predictions.

Where is the catch?

Bayes' Law:

$$p(H|D) = \frac{p(D|H)p(H)}{p(D)}$$

Caution! Denominator is often an intractable integral

$$p(D) = \int_{\mathcal{H}} p(D|H)p(H)\, dH$$

. . . or a summation over a large number of points

$$p(D) = \sum_{\mathcal{H}} p(D|H)p(H)$$

The *shape* of a Bayesian model

A frequentist regression model

Notation

$\mathbf{y}$: Observed dependent data

$\mathbf{X}$: Observed independent data as a design matrix

$\mathbf{w}$: Weights / regression coefficients

θ: A placeholder for any kind of parameter

$$p(\mathbf{y}) = p(\mathbf{y}|X, \mathbf{w}, \theta)$$

Use maximum likelihood to find values for $\mathbf{w}$ and θ

e.g. OLS regression

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}|X\mathbf{w}, \theta)$$

A Bayesian regression model

Notation

$\mathbf{y}$: Observed dependent data

$\mathbf{X}$: Observed independent data as a design matrix

$\mathbf{w}$: Weights / regression coefficients

θ: A placeholder for any kind of parameter

$$p(\mathbf{y}) = p(\mathbf{y}|X, \mathbf{w}, \theta)p(\mathbf{w}|\theta)$$

Use maximum *marginal* likelihood to find values for θ

$$p(\mathbf{y}|\mathbf{X}, \theta) = \int p(\mathbf{y}|X, \mathbf{w}, \theta)p(\mathbf{w}|\theta)\, d\mathbf{w}$$

Remarks

Evaluating the marginal likelihood

$$p(\mathbf{y}|\mathbf{X}, \theta) = \int p(\mathbf{y}|X, \mathbf{w}, \theta) p(\mathbf{w}|\theta) \, d\mathbf{w}$$

The likelihood distribution is ok to evaluate

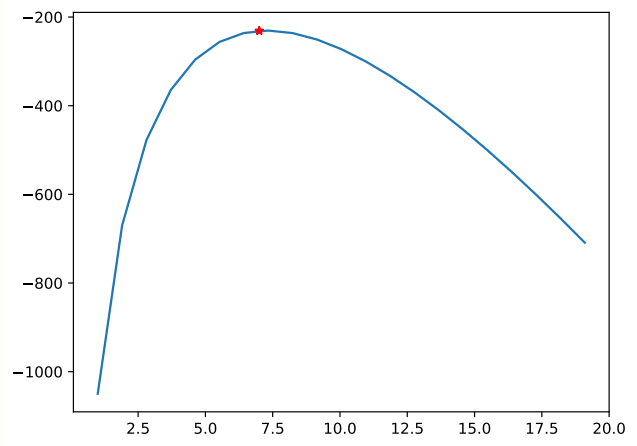
- Even if the dependency on $\mathbf{X}, \mathbf{w}$ is non-linear
- Provided we are given a value of $\mathbf{w}$ to use

The overall integral may well be intractable

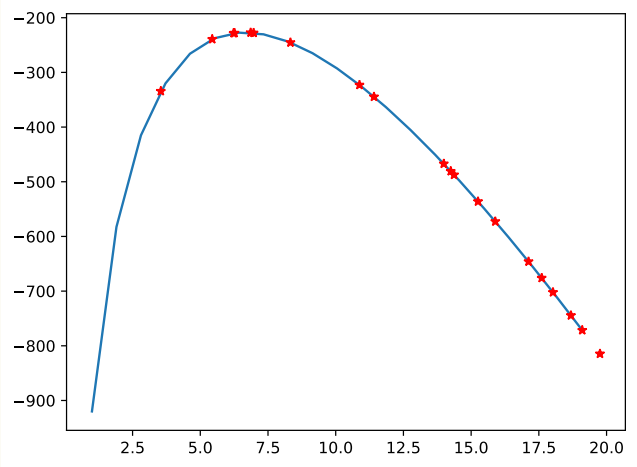
- We need a nice numerical method
- That can deliver sensible values of $\mathbf{w}$
- Based on the geometry of the prior distribution

Let’s look at something practical

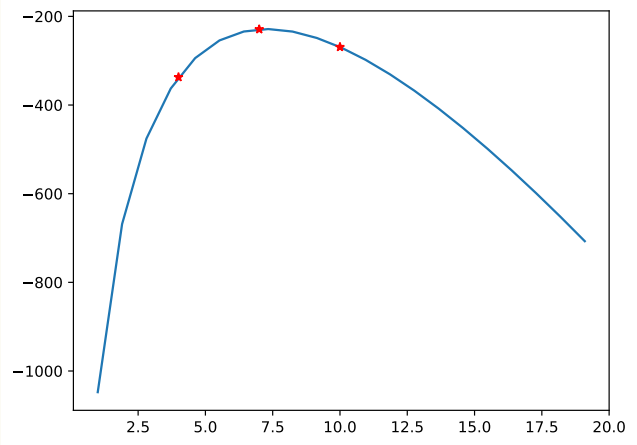
Choose just one parameter value?



Use *many* parameter values?

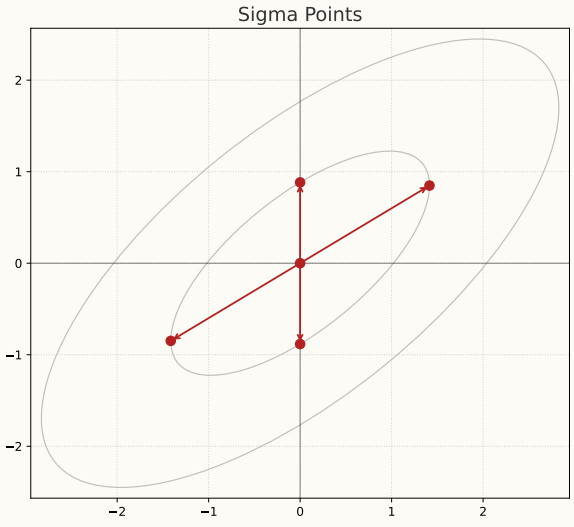


. . . or a few carefully selected parameter values?



Sigma Points

Sigma Points for a Gaussian prior



Using Sigma Points

Evaluate the maximum *marginal* likelihood:

$$p(\mathbf{y}|\mathbf{X},\theta) = \int p(\mathbf{y}|X,\mathbf{w},\theta)\mathcal{N}(\mathbf{w}|\theta)\,d\mathbf{w}$$

```
For each sigma point:
    Evaluate prior prob of sigma point
    Evaluate likelihood of data given sigma point value
    Obtain joint: prior x likelihood
Obtain marginal: sum of joint probabilities
```

Easy, scalable and accurate!
2 sigma-points in each dimension plus one at the centre: $2M + 1$ in total

Where do we place the sigma points?

Sigma-point locations

- De-mean the prior
- Place a central sigma-point at $\mathbf{0}$
- Place "outer" sigma-points at $\pm\mathbf{v_m}$ for each dimension $m \in 1, \ldots, M$

Cholesky factor

- $\mathbf{v_m}$ is the m^{th} column of $\mathbf{C}$
- $\mathbf{C}$ is the Cholesky factor of $\mathbf{\Sigma}$
- $\mathbf{\Sigma}$ is the covariance of the prior over weights $\mathbf{w}$

Why Cholesky?

Cholesky factor

- Cholesky factorisation algorithm is extremely stable
- Cholesky factor is a square root of a square matrix
- Cholesky factor of a covariance is multivariate analog to standard deviation

- Given uncorrelated samples $\boldsymbol{\epsilon}$ from a multivariate Gaussian
- There is a matrix $\mathbf{C}$ that will give *correlated* samples $\boldsymbol{\phi} = \mathbf{C}\boldsymbol{\epsilon}$
- $\boldsymbol{\Sigma} = E[\boldsymbol{\phi}\boldsymbol{\phi}^T] = E[\mathbf{C}\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T\mathbf{C}^T] = \mathbf{C}\mathbf{C}^T$
- So we must have $\boldsymbol{\Sigma} = \mathbf{C}\mathbf{C}^T$ i.e. $\mathbf{C}$ is the Cholesky factor of $\boldsymbol{\Sigma}$

The Unscented Transform

The Unscented Transform

Motivation: a non-linear function somewhere in our model

$$p(\mathbf{y}) = p(\mathbf{y}|f(X, \mathbf{w}), \theta)p(\mathbf{w}|\theta)$$

Examples

- Aerospace engineering: radar sightings, polar coordinates, *sin*, *cos*, etc
- Probabilistic modelling: non-Gaussian likelihood densities
  - GLM link functions
  - Neural network activation functions

The Unscented Transform

Approximate f using a Taylor expansion

$$\begin{aligned}\mathbf{y} &= f(\mathbf{x}) \\ &\approx f(\bar{\mathbf{x}}) + \nabla f(\bar{\mathbf{x}})\delta x + \frac{1}{2}\nabla^2 f(\bar{\mathbf{x}})\delta x^2 + \dots\end{aligned}$$

Take expectations

$$\begin{aligned}\bar{\mathbf{y}} &= f(\bar{\mathbf{x}}) + \frac{1}{2}\nabla^2 f(\bar{\mathbf{x}})\mathbb{E}[\delta x^2] + \dots \\ &= f(\bar{\mathbf{x}}) + \frac{1}{2}\nabla^2 f(\bar{\mathbf{x}})\boldsymbol{\Sigma} + \dots\end{aligned}$$

The Unscented Transform

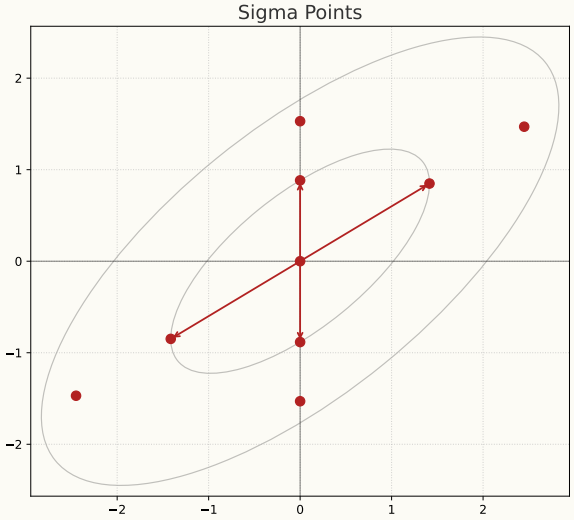
UT as a second-order Taylor approximation

$$\bar{\mathbf{y}} = f(\bar{\mathbf{x}}) + \frac{1}{2}\nabla^2 f(\bar{\mathbf{x}})\boldsymbol{\Sigma} + \dots$$

The Unscented Transform was a reaction to the Extended Kalman Filter

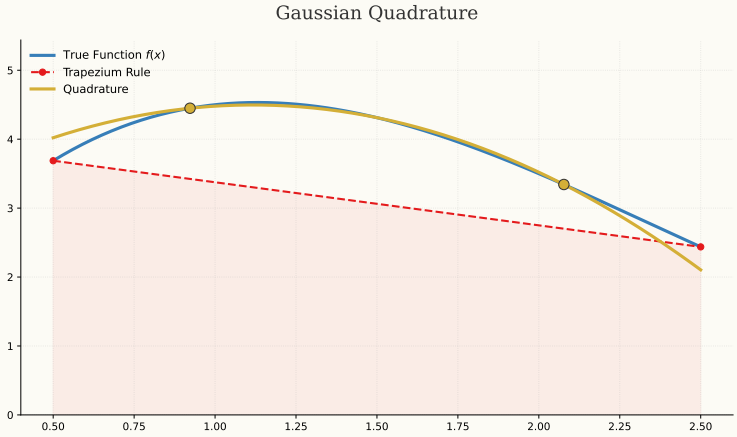
- EKF: only propagates first-order term $f(\bar{\mathbf{x}})$
- UT / UKF: propagates first and second order terms
- Propagate higher order terms by adding further rings of sigma points

Rings of Saturn



Gaussian Quadrature

Gaussian quadrature



Sigma points as Gaussian quadrature

With a Gaussian prior

- Gauss-Hermite quadrature
- Uses Hermite (probabilists') polynomials $He_i(x)$
- With one ring of sigma points, we use the roots of $He_3(x) = x^3 - 3x$
- And place the sigma points at distances of 0 and $\pm\sqrt{3}$ from the mean
- ...measured in units of the lengths of each column of the Cholesky factor

With other priors we can* use other orthogonal polynomial sequences

- Gamma: Laguerre
- Beta: Jacobi
- Uniform: Legendre

* In principle. Work in progress!

Connections

Finite elements

©The Guardian

Deep learning with ReLU

Orthogonal polynomial sequences

- Can be used to define a basis over function space
- Which is (implicitly) how we have used them

But!

- They are "active" everywhere . . .
- . . . which makes them global and not local

Contrast with ReLU

- Defines a half-plane
- which can be used to define a basis in function space
- . . . piecewise linear approximations (like Finite Elements Analysis)
- But ReLU is (half-) local . . . (because half-planes)

Suggests the idea of a small ReLU-based network as the latent variable of Bayesian models.

A sort of conclusion

Connections between

- Sigma points and the unscented transform
- Bases defined by orthogonal polynomial sequences (cf Fourier bases)
- Finite elements analysis
- ReLU-based deep learning