Bayesian Optimization for Accelerated Exploration of Chemical Space

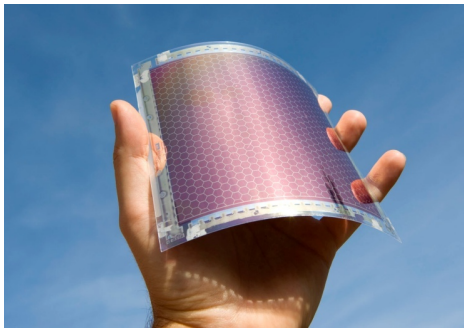
José Miguel Hernández–Lobato
Department of Engineering
University of Cambridge

http://jmhl.org, jmh233@cam.ac.uk

Drug and material design

Goal: find novel molecules that optimally fulfill various metrics.





About 10^8 compounds in databases, potential ones: $10^{20} - 10^{60}$.

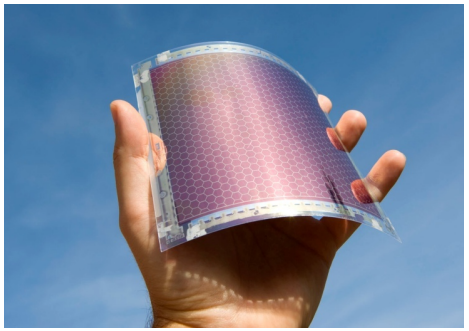
Challenges:

- Evaluating molecular properties is slow and expensive.
- Chemical space is huge.

Drug and material design

Goal: find novel molecules that optimally fulfill various metrics.





About 10^8 compounds in databases, potential ones: $10^{20} - 10^{60}$.

Challenges:

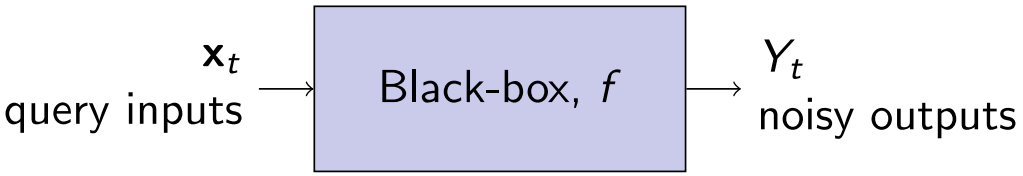
- Evaluating molecular properties is slow and expensive.
- Chemical space is huge.

Bayesian optimization can accelerate the search.

Bayesian optimization aims to efficiently optimize **black-box** functions:

$$\mathbf{x}^\star = \arg\max_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})$$

No gradients, observations may be **corrupted by noise**.

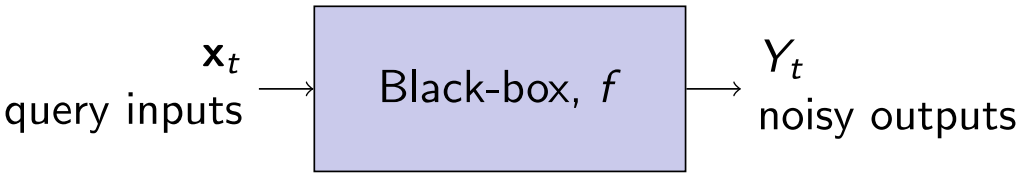


Black-box queries are **very expensive** (time, economic cost, etc...).

Bayesian optimization aims to efficiently optimize **black-box** functions:

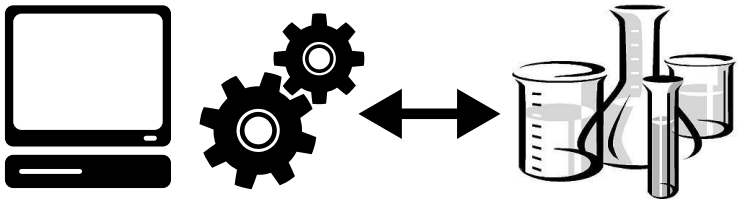
$$\mathbf{x}^\star = \arg\max_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$$

No gradients, observations may be **corrupted by noise**.



Black-box queries are **very expensive** (time, economic cost, etc...).

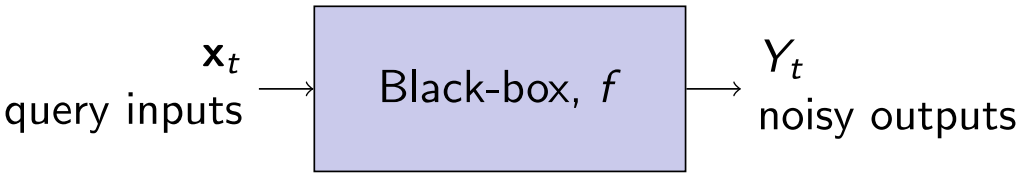
Main idea: replace expensive black-box quieries with **cheaper computations** that will save additional queries in the long run.



Bayesian optimization aims to efficiently optimize **black-box** functions:

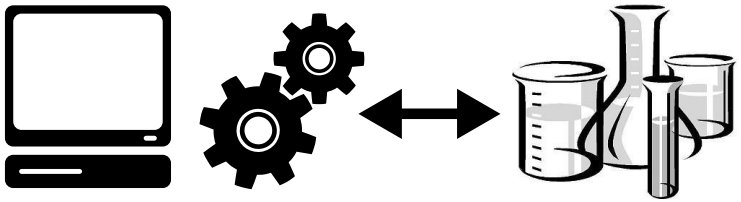
$$\mathbf{x}^\star = \arg\max_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$$

No gradients, observations may be **corrupted by noise**.



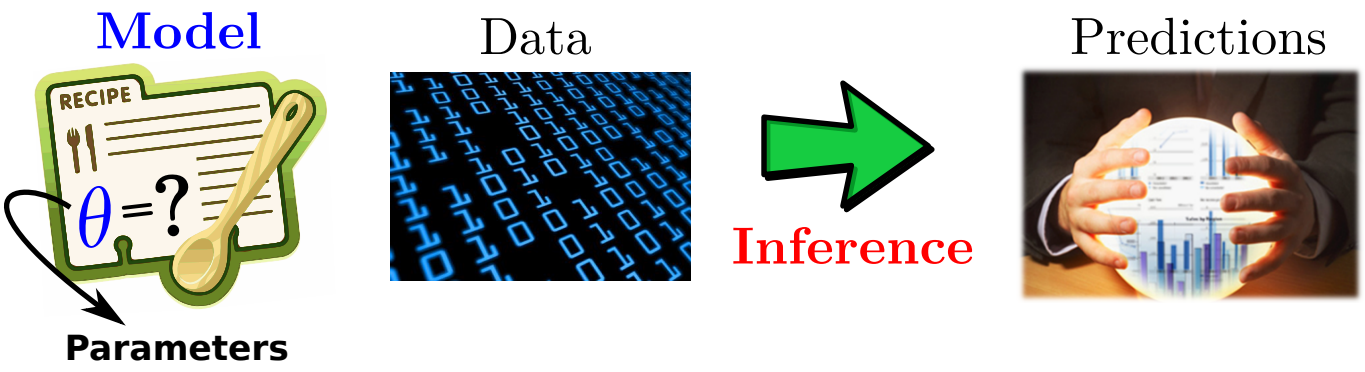
Black-box queries are **very expensive** (time, economic cost, etc...).

Main idea: replace expensive black-box quieries with **cheaper computations** that will save additional queries in the long run.



Done in practice by using **Bayesian machine learning**.

The Bayesian approach to machine learning: **probabilistic models** and **inference algorithms**.



The Bayesian approach to machine learning: **probabilistic models** and **inference algorithms**.

Model

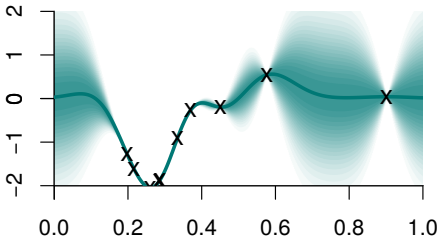
Parameters

Data

Inference

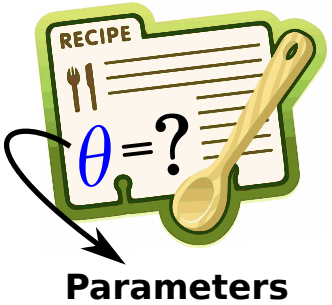
Predictions

Uncertainty



The Bayesian approach to machine learning: **probabilistic models** and **inference algorithms**.

Model

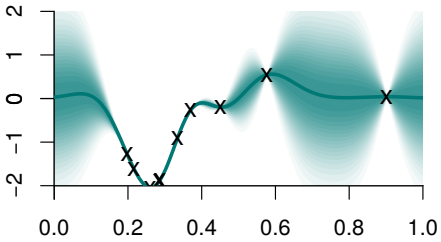


Data

Inference

Predictions

Uncertainty



Implemented using **probability theory** and **Bayes' rule**.

The Bayesian approach to machine learning: **probabilistic models** and **inference algorithms**.

Model

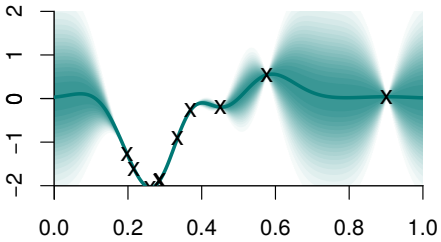
Parameters

Data

Inference

Predictions

Uncertainty



Implemented using **probability theory** and **Bayes' rule**.

$$p(\theta\,|\text{Data}) = \frac{p(\text{Data}|\theta)p(\theta)}{p(\text{Data})}, \qquad \textbf{Model}$$

The Bayesian approach to machine learning: **probabilistic models** and **inference algorithms**.

Model

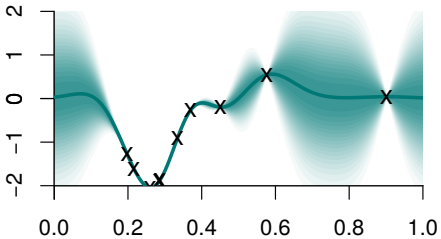
Parameters

Data

Inference

Predictions

Uncertainty



Implemented using **probability theory** and **Bayes' rule**.

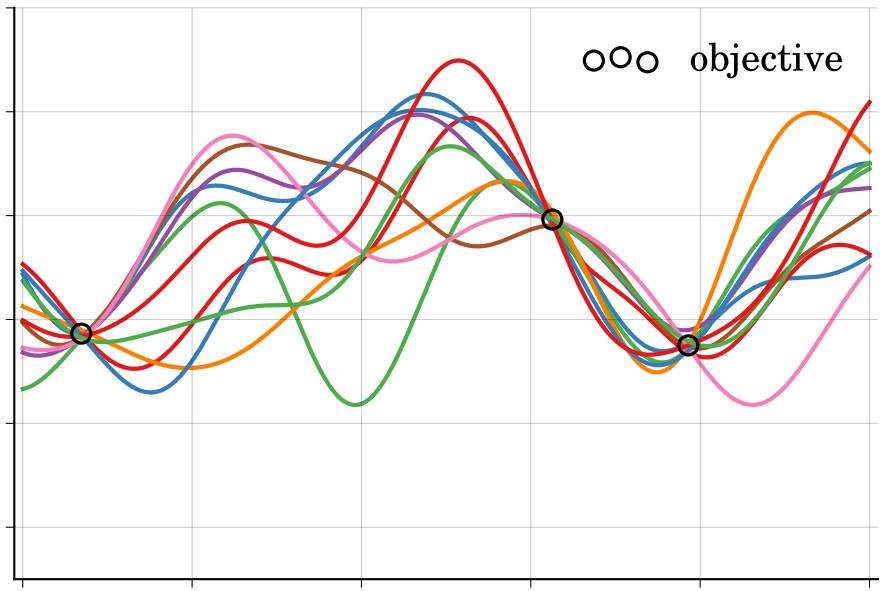
$$p(\theta\,|\text{Data}) = \frac{p(\text{Data}|\theta)p(\theta)}{p(\text{Data})}, \qquad \textbf{Model}$$

$$p(y|\text{Data}) = \int p(y|\theta)p(\theta\,|\text{Data})\,d\theta. \qquad \textbf{Inference}$$

Gaussian processes are the default **model** in Bayesian optimization.

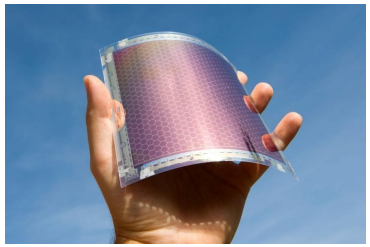
Advantages:

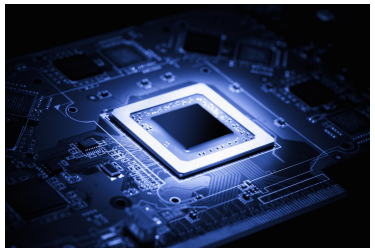
- They are non-parametric.
- **Inference** is exact: closed-form **Gaussian predictive distribution**.



A **covariance function** $C(\mathbf{x}_1, \mathbf{x}_2)$ is used to measure data point similarity.

Drugs **Materials** **Robots** **Hardware**

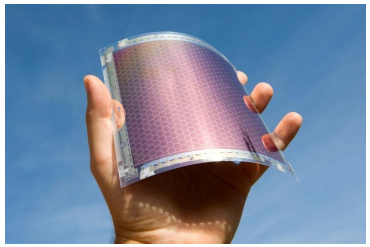


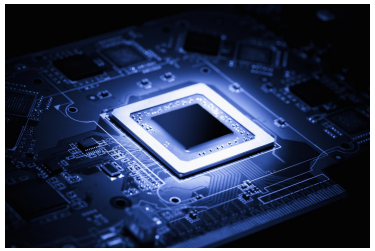
Optimal design in engineering

problems ↓ ↑ solutions

Bayesian optimization

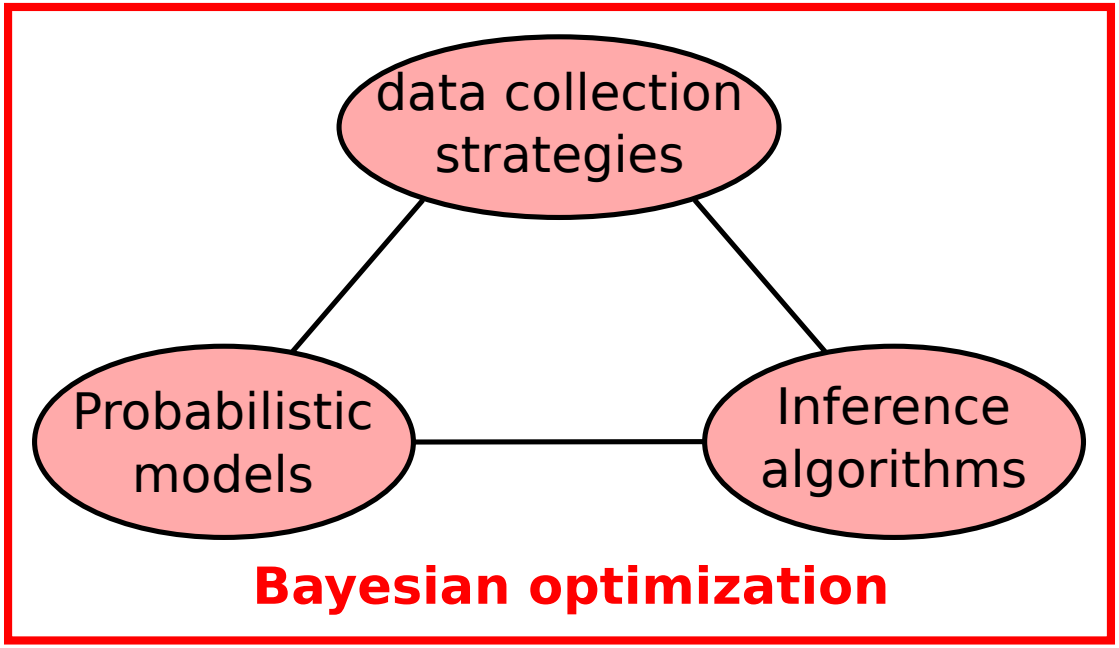
Drugs **Materials** **Robots** **Hardware**



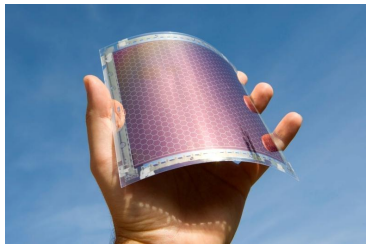
Optimal design in engineering

problems ↓ ↑ solutions



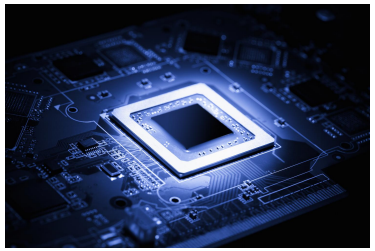
Drugs

Materials

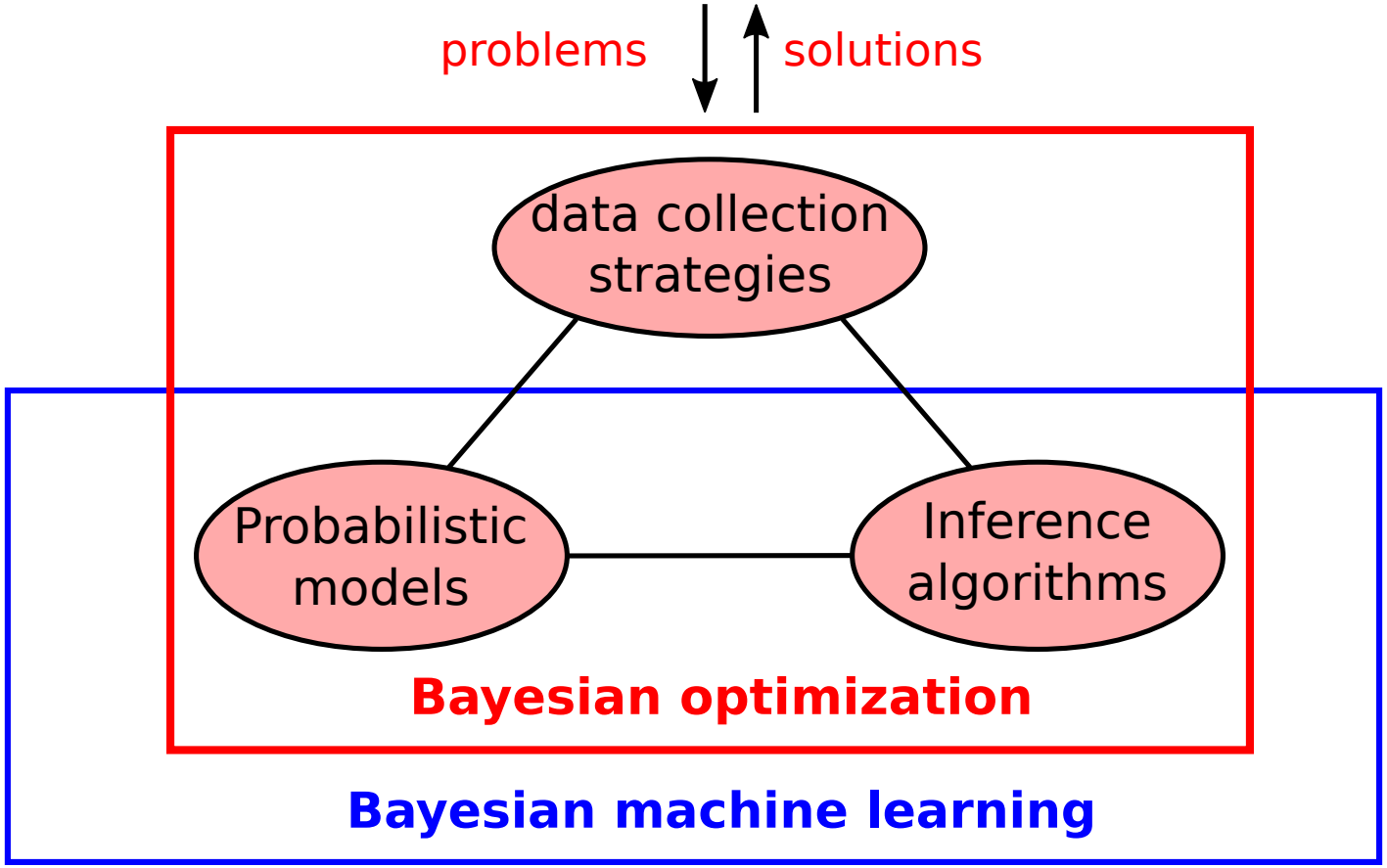


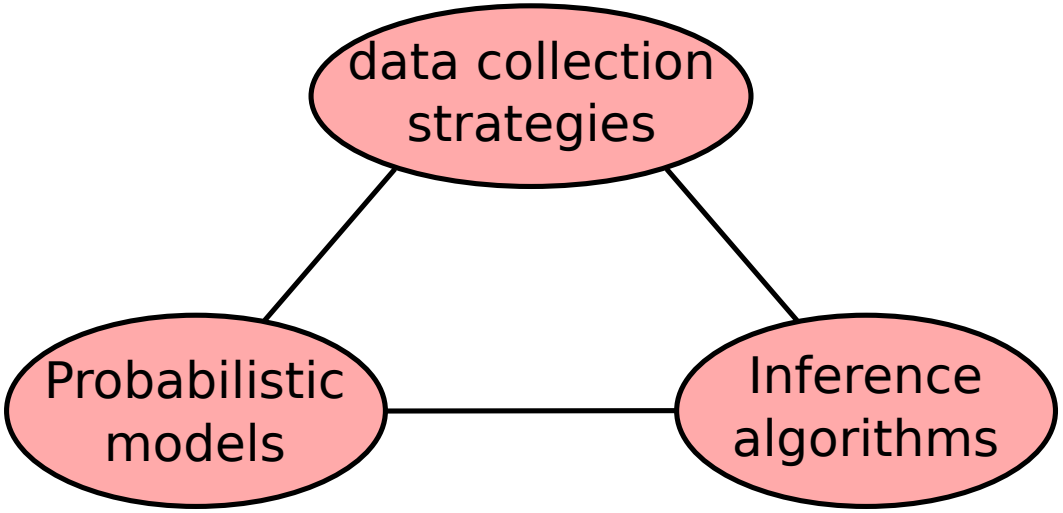
Robots

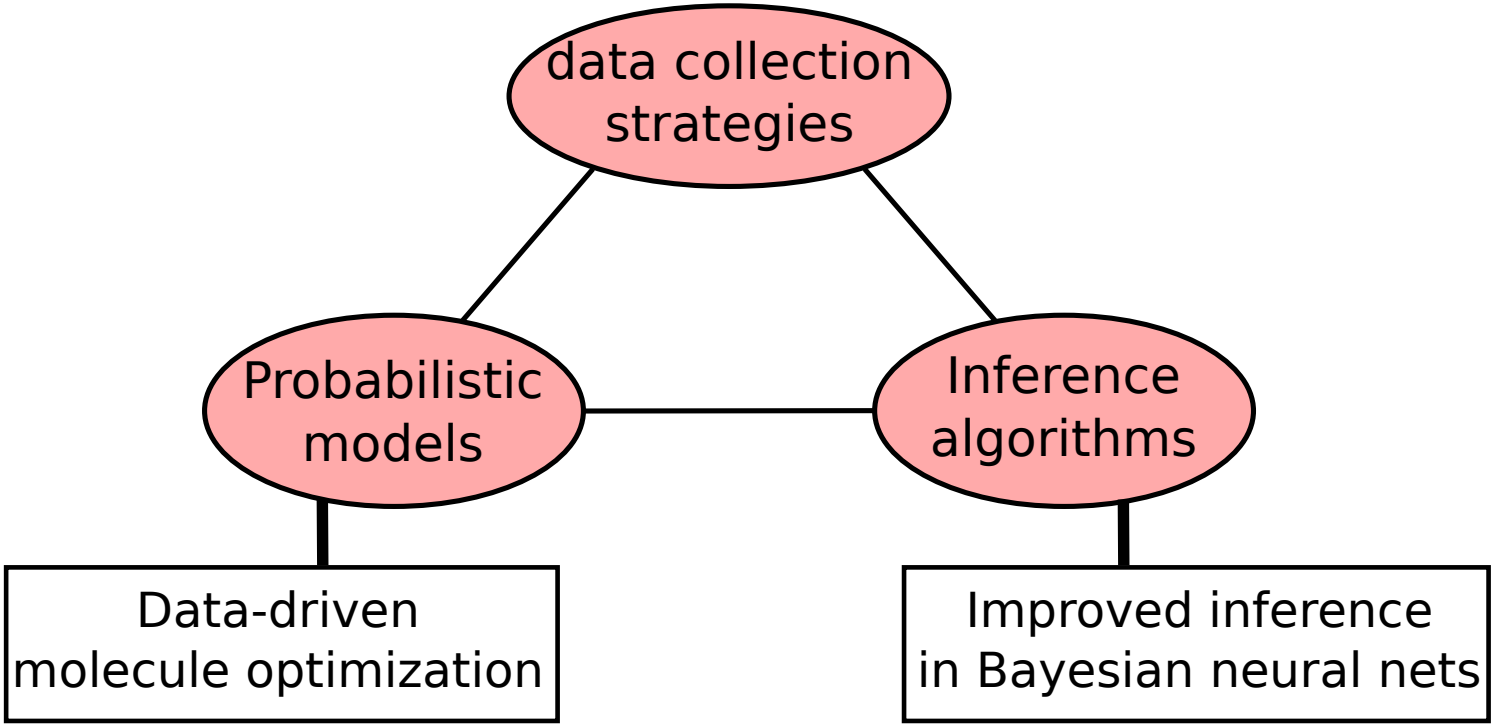
Hardware



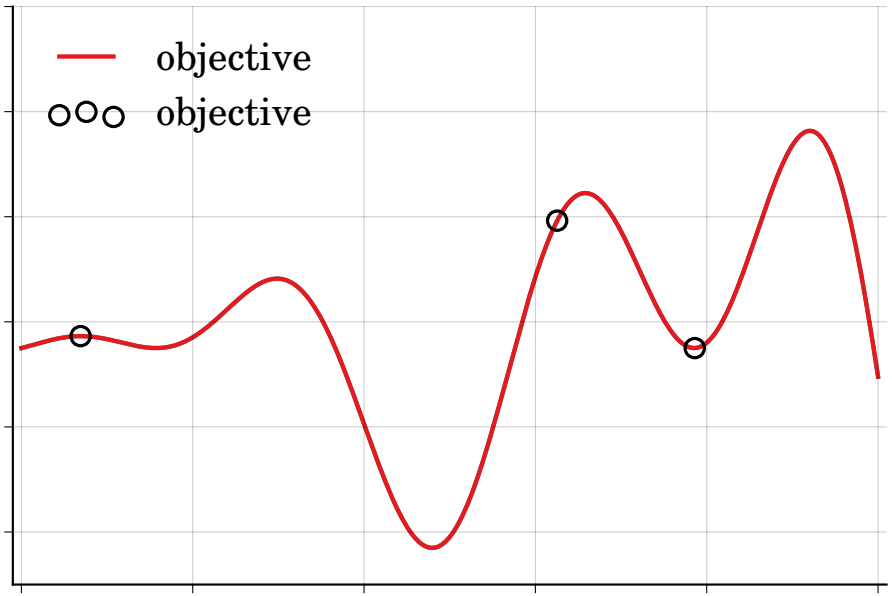
Optimal design in engineering



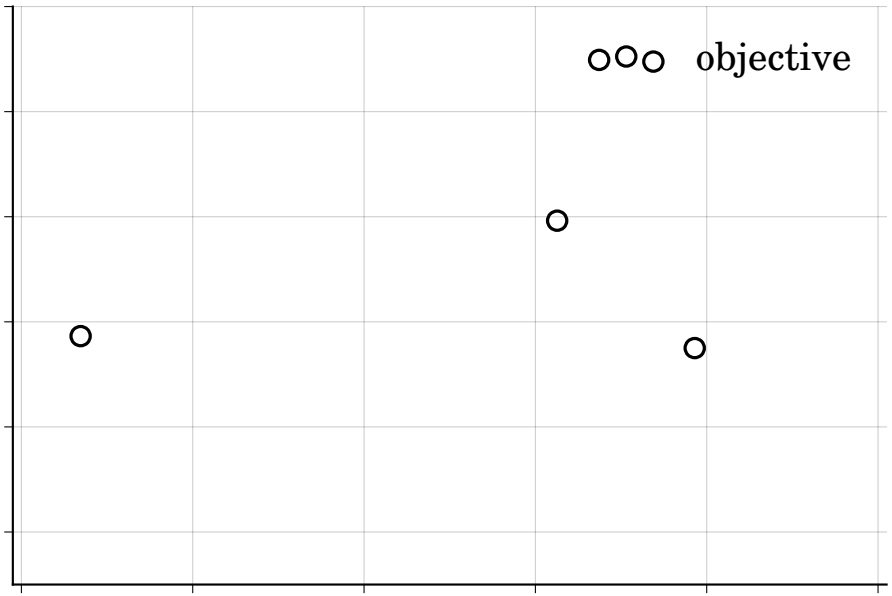

data collection strategies
Probabilistic models
Inference algorithms


data collection strategies
Probabilistic models
Inference algorithms
Data-driven molecule optimization
Improved inference in Bayesian neural nets

1. Get initial sample.

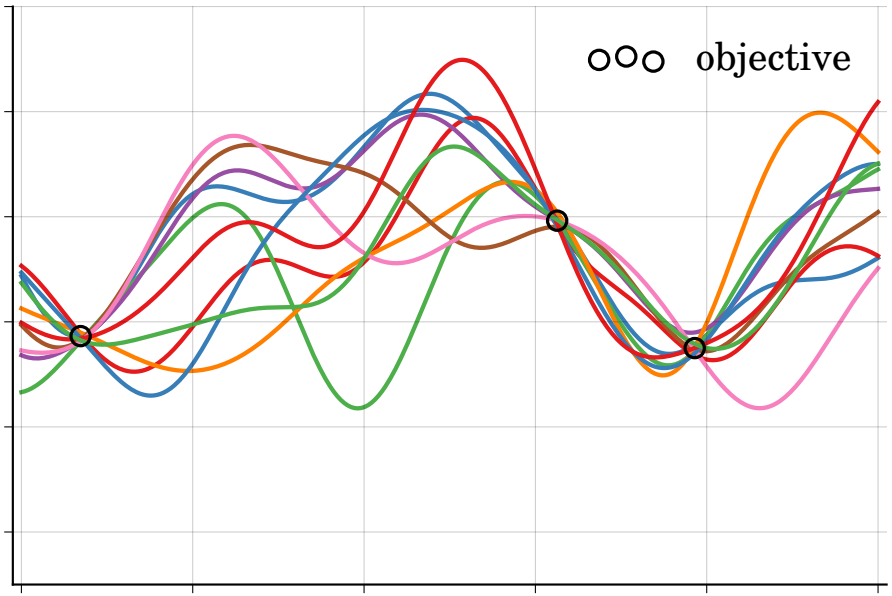


❶ **Get initial sample.**



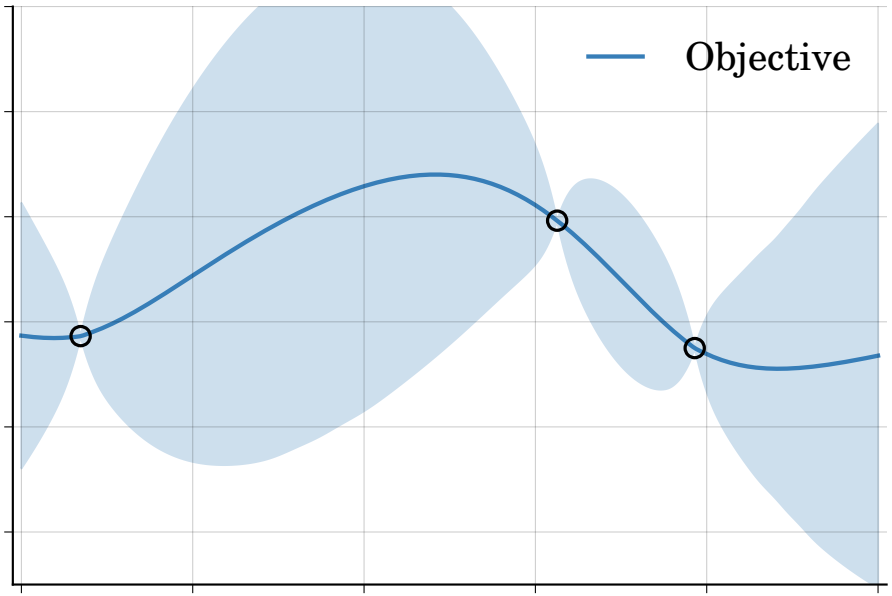
1. **Get initial sample.**
2. Fit a model to the data:

$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$



1. Get initial sample.
2. **Fit a model to the data:**

$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$

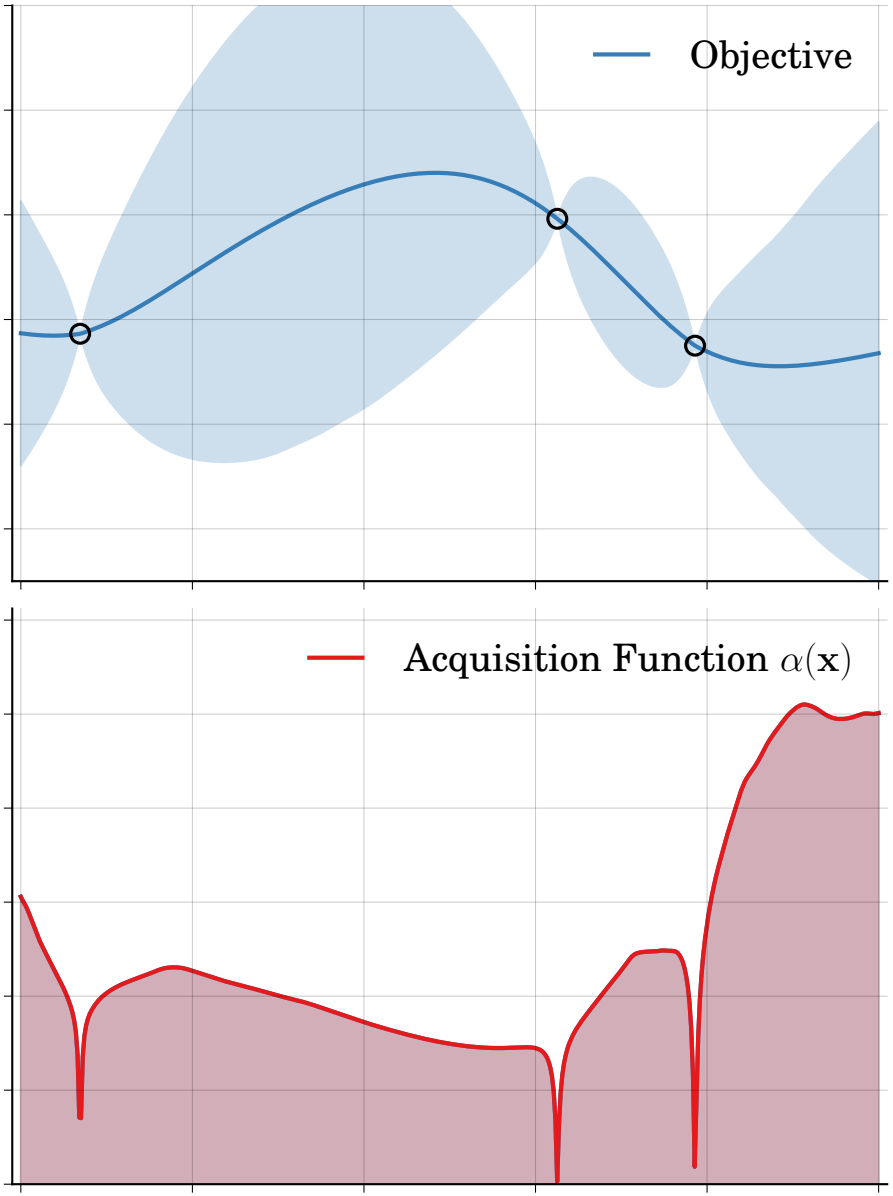


1. Get initial sample.
2. **Fit a model to the data:**

$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$

3. Select data collection strategy:

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[U(y|\mathbf{x}, \mathcal{D}_n)]\,.$$



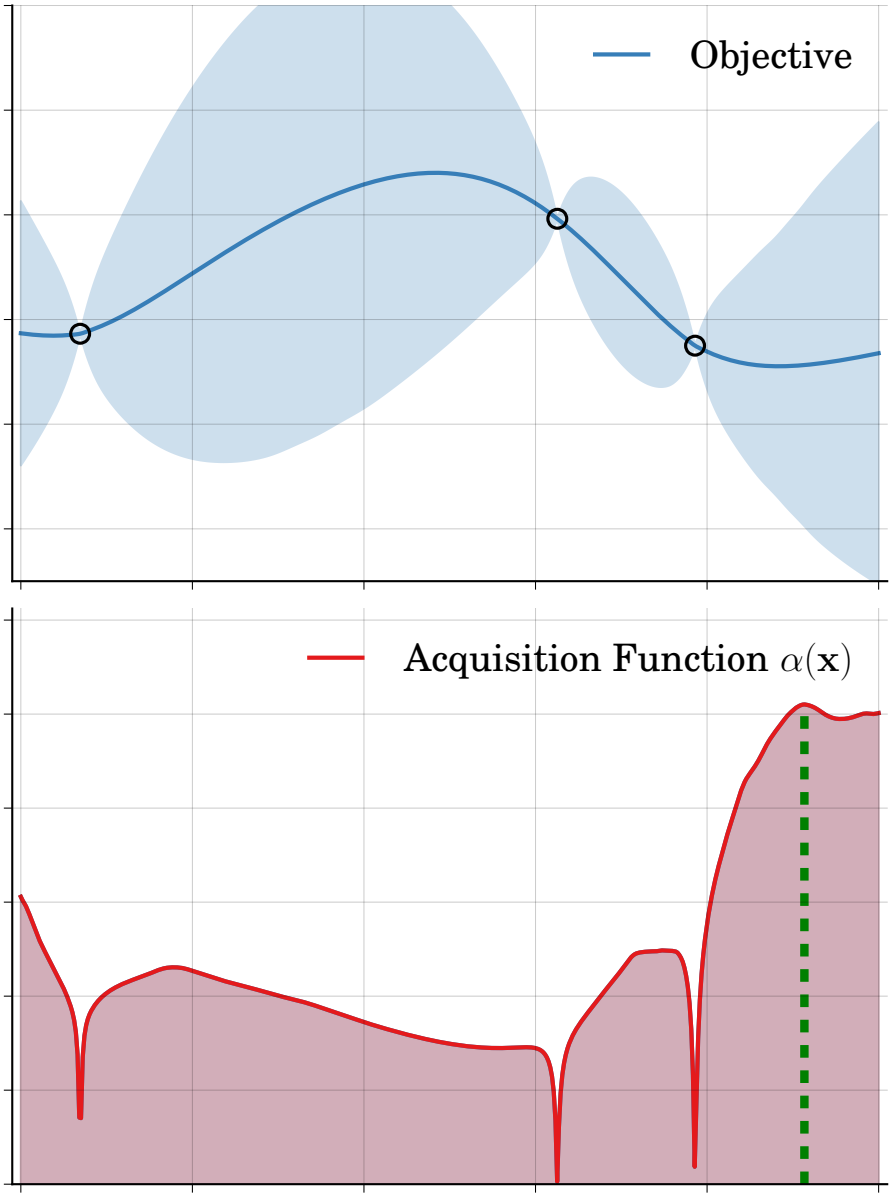
1. Get initial sample.
2. Fit a model to the data:

$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$

3. **Select data collection strategy:**

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[U(y|\mathbf{x}, \mathcal{D}_n)]\,.$$

4. Optimize acquisition function $\alpha(\mathbf{x})$.



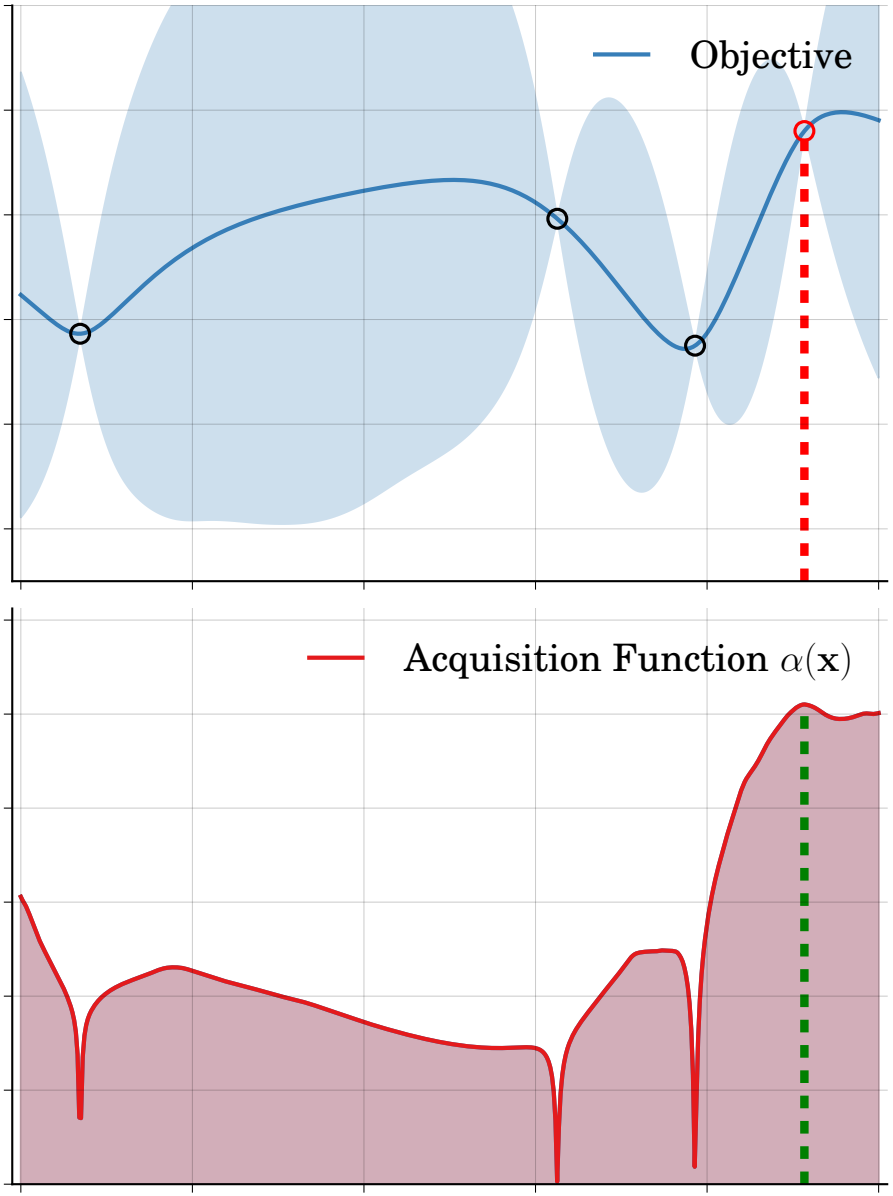
1. Get initial sample.
2. Fit a model to the data:

$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$

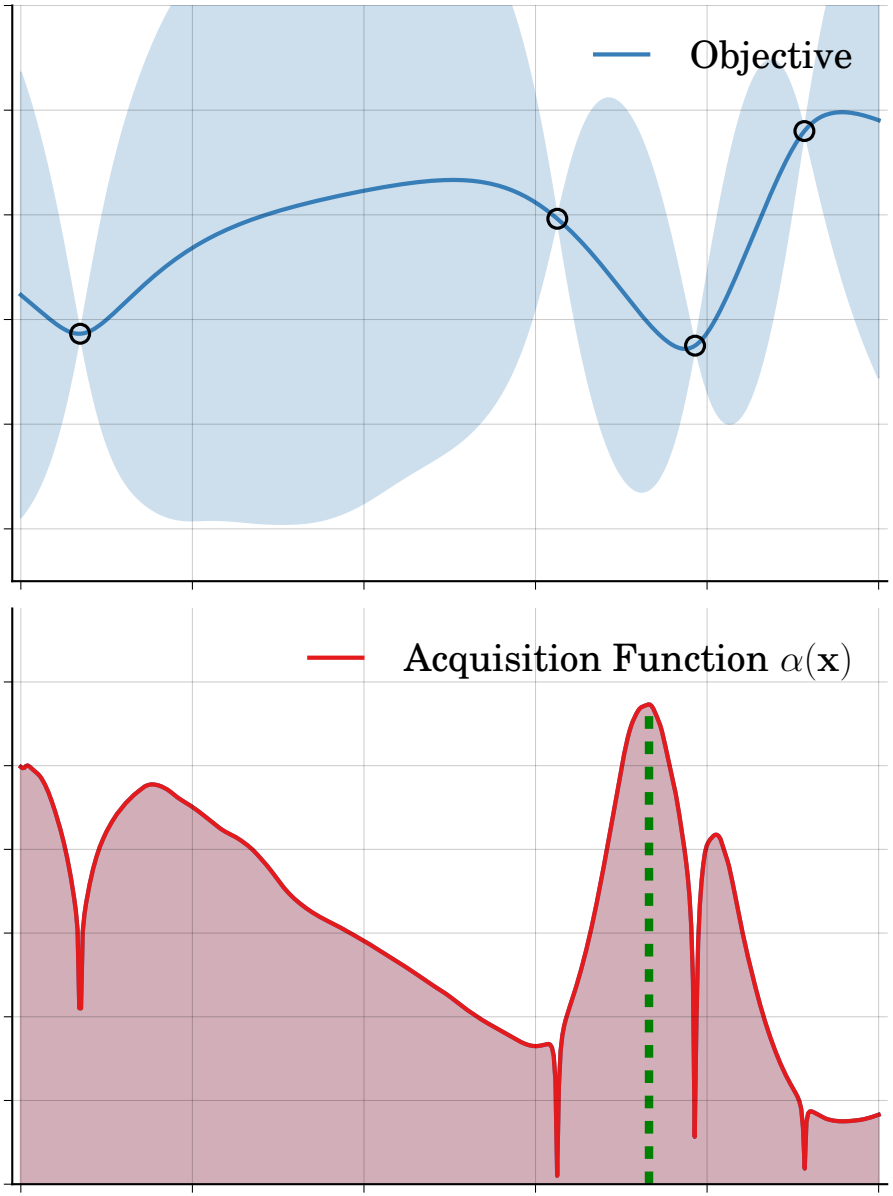
3. Select data collection strategy:

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[U(y|\mathbf{x}, \mathcal{D}_n)]\,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. Collect data and update model.



1. Get initial sample.
2. Fit a model to the data:
$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$
3. Select data collection strategy:
$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[U(y|\mathbf{x}, \mathcal{D}_n)]\,.$$
4. Optimize acquisition function $\alpha(\mathbf{x})$.
5. **Collect data and update model.**
6. Repeat!



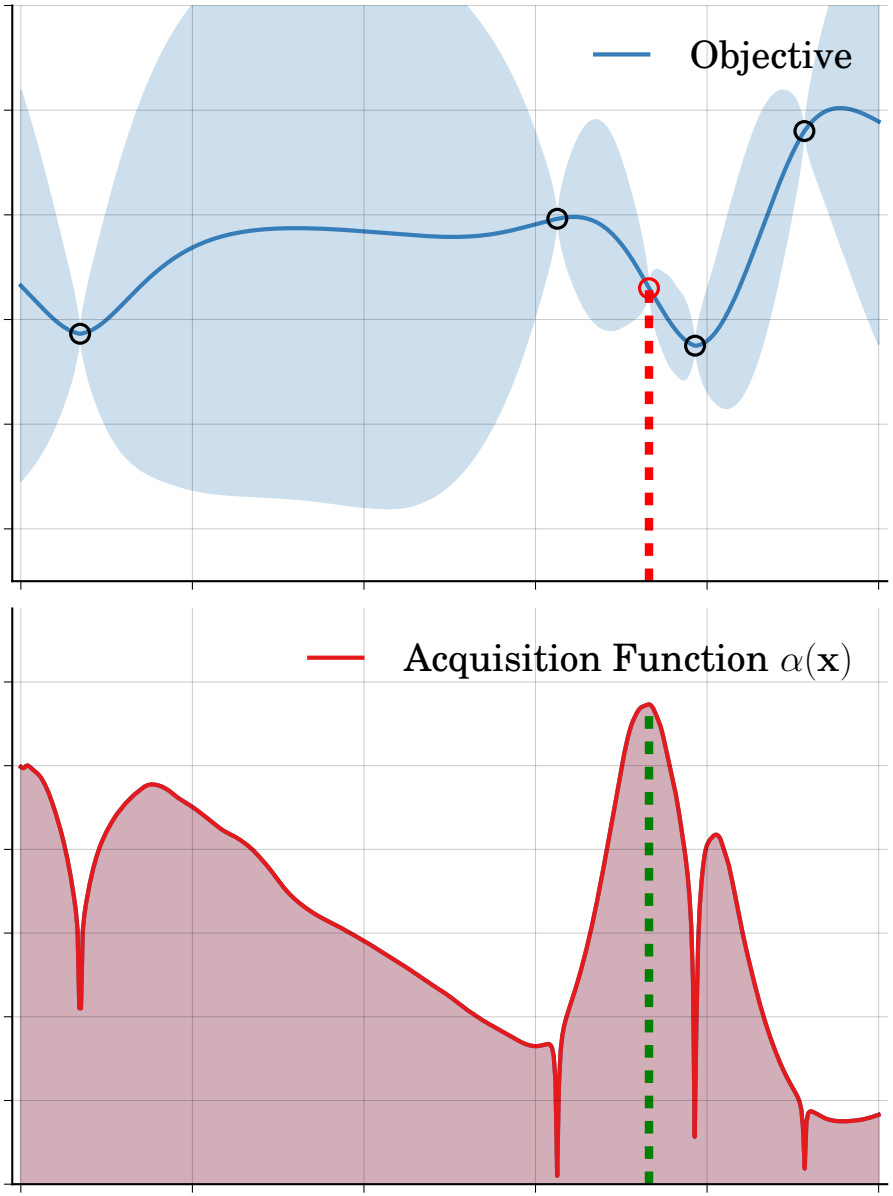
1. Get initial sample.
2. Fit a model to the data:

$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$

3. **Select data collection strategy:**

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[U(y|\mathbf{x}, \mathcal{D}_n)]\,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**



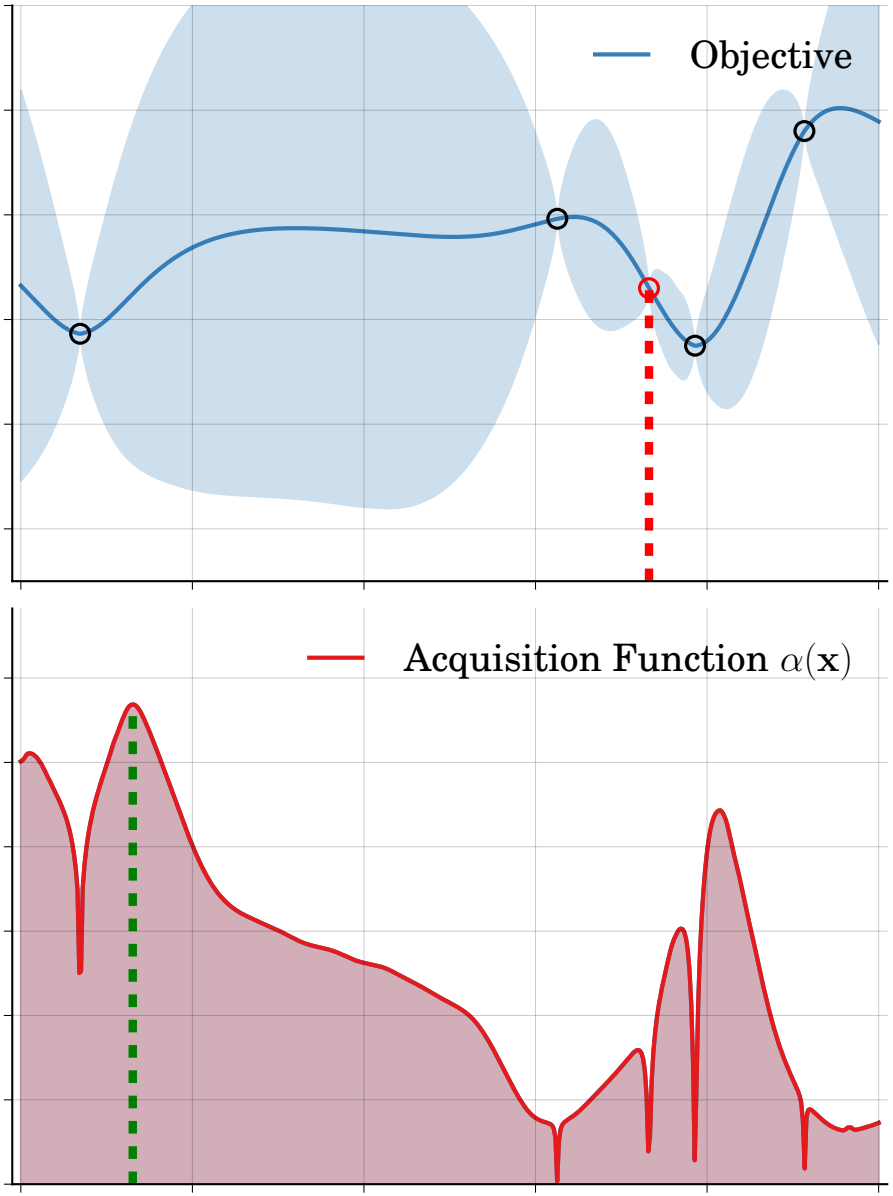
1. Get initial sample.
2. Fit a model to the data:

$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$

3. **Select data collection strategy:**

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[U(y|\mathbf{x}, \mathcal{D}_n)]\,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**



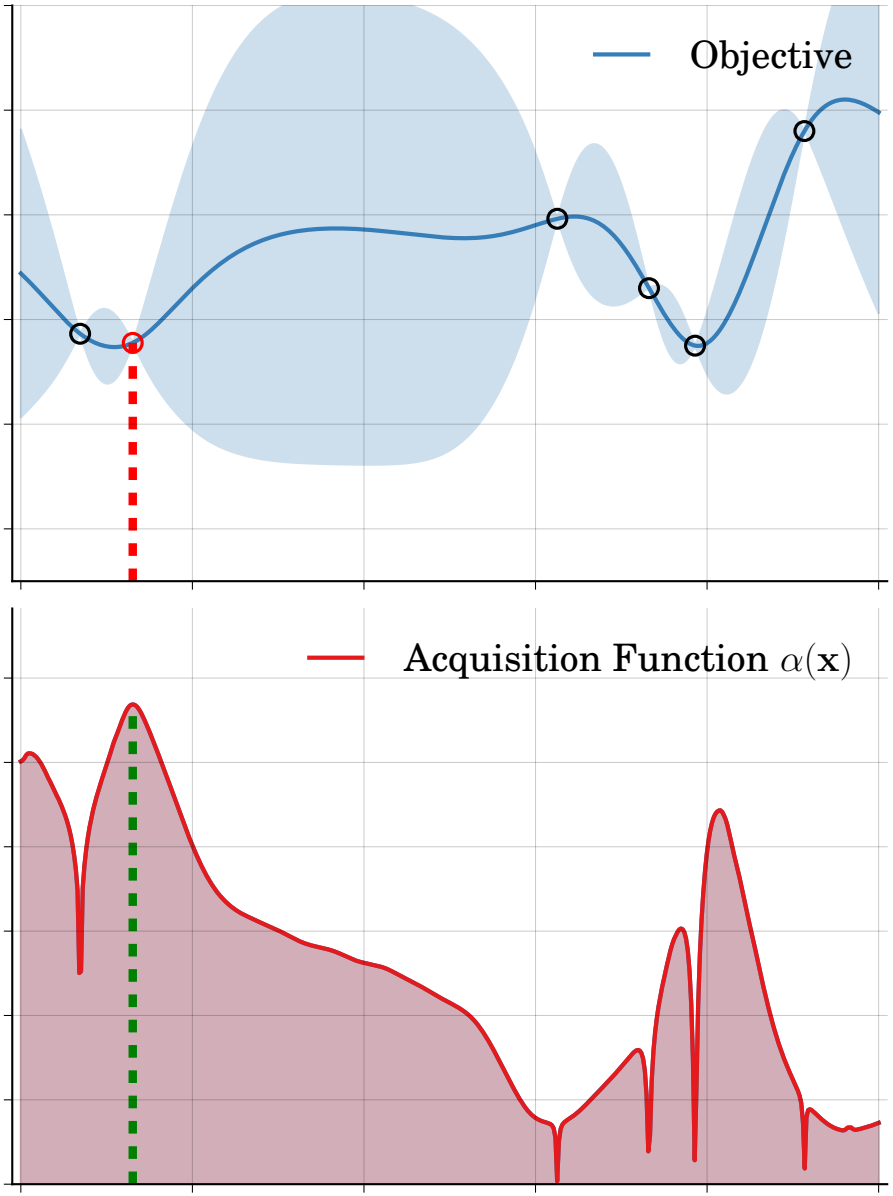
1. Get initial sample.
2. Fit a model to the data:

$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$

3. **Select data collection strategy:**

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[U(y|\mathbf{x}, \mathcal{D}_n)]\,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**



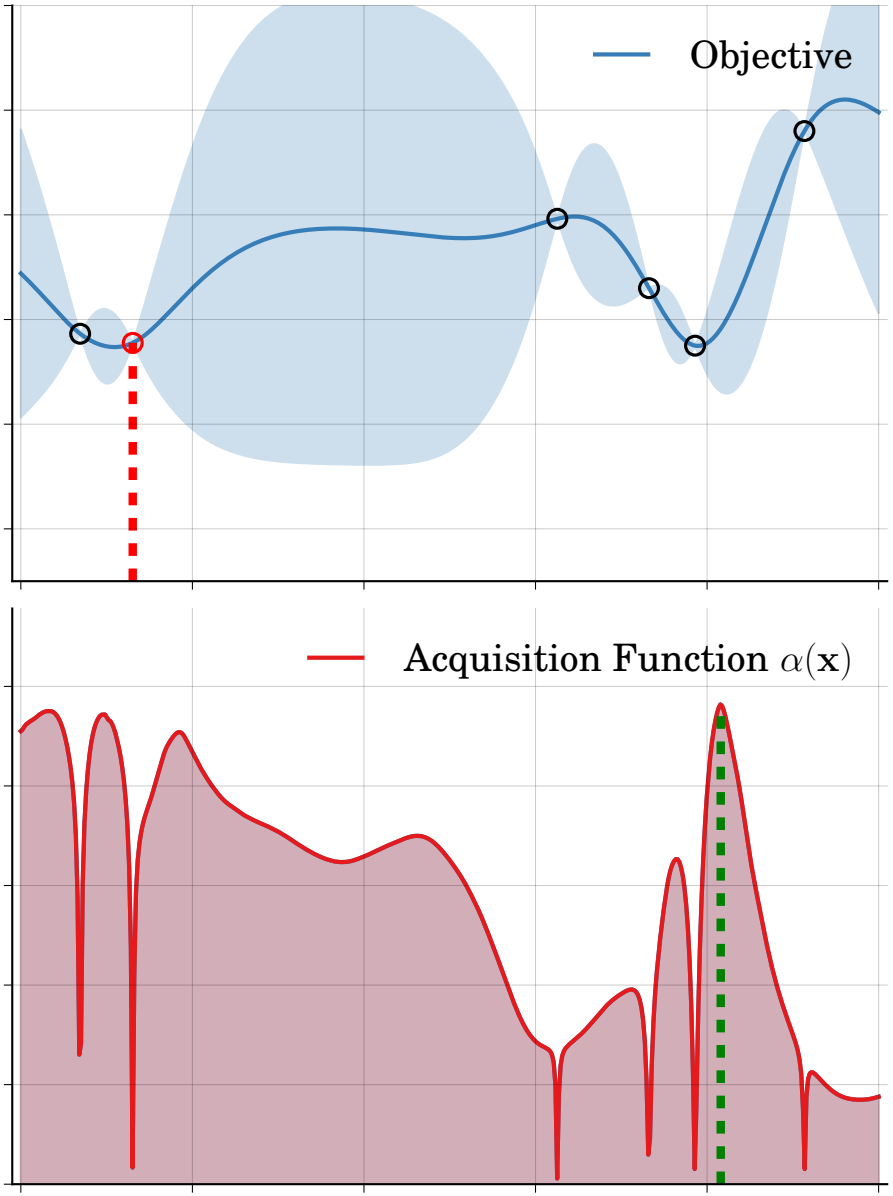
1. Get initial sample.
2. Fit a model to the data:

$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$

3. **Select data collection strategy:**

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[U(y|\mathbf{x}, \mathcal{D}_n)]\,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**



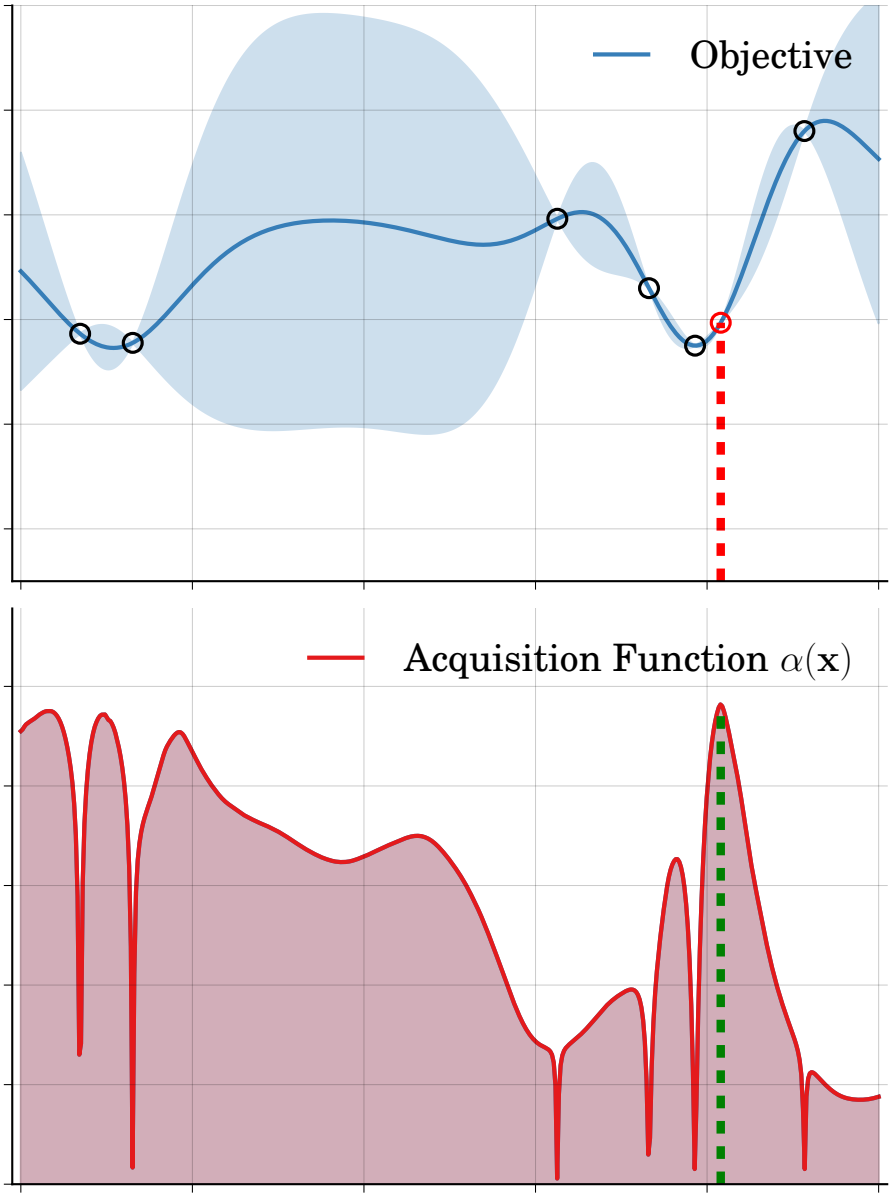
1. Get initial sample.
2. Fit a model to the data:

$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$

3. **Select data collection strategy:**

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[U(y|\mathbf{x}, \mathcal{D}_n)]\,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**



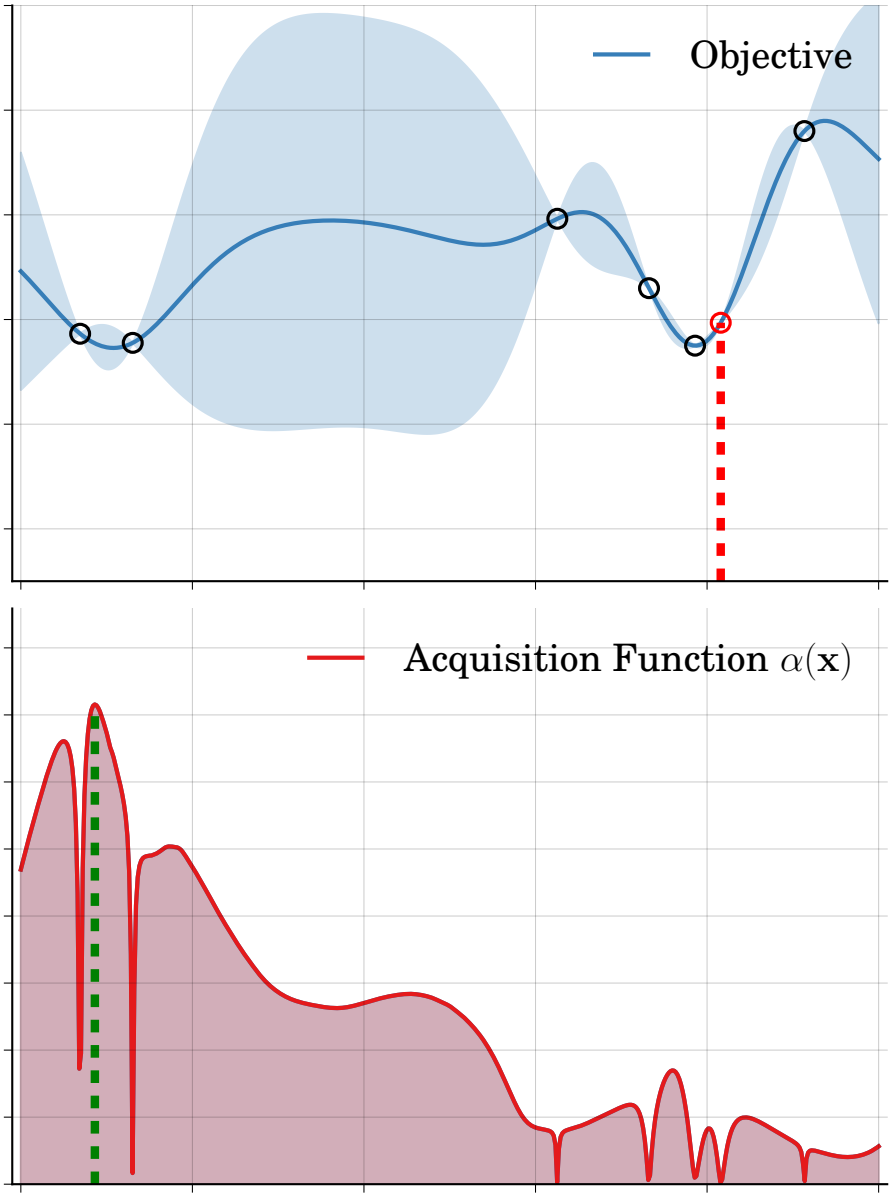
1. Get initial sample.
2. Fit a model to the data:

$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$

3. **Select data collection strategy:**

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[U(y|\mathbf{x}, \mathcal{D}_n)]\,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**



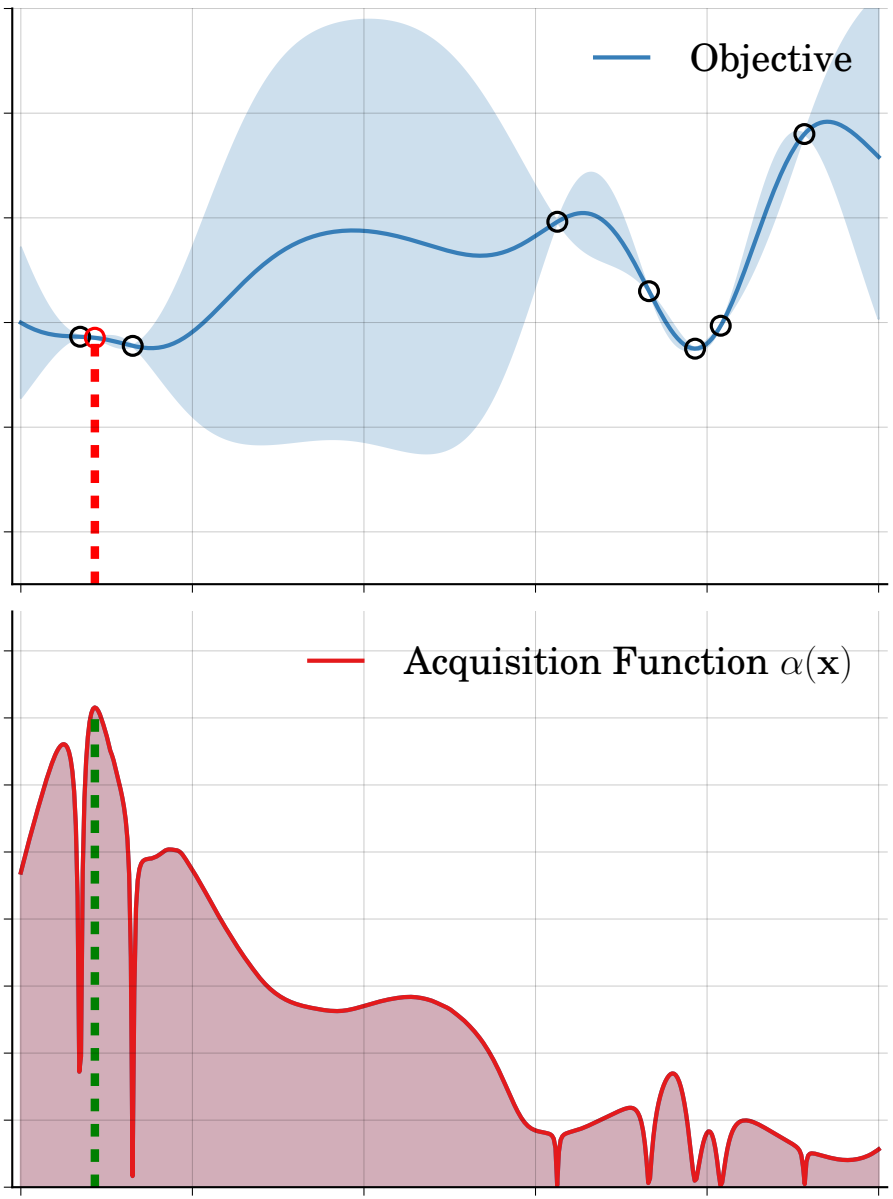
1. Get initial sample.
2. Fit a model to the data:

$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$

3. **Select data collection strategy:**

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[U(y|\mathbf{x}, \mathcal{D}_n)]\,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**



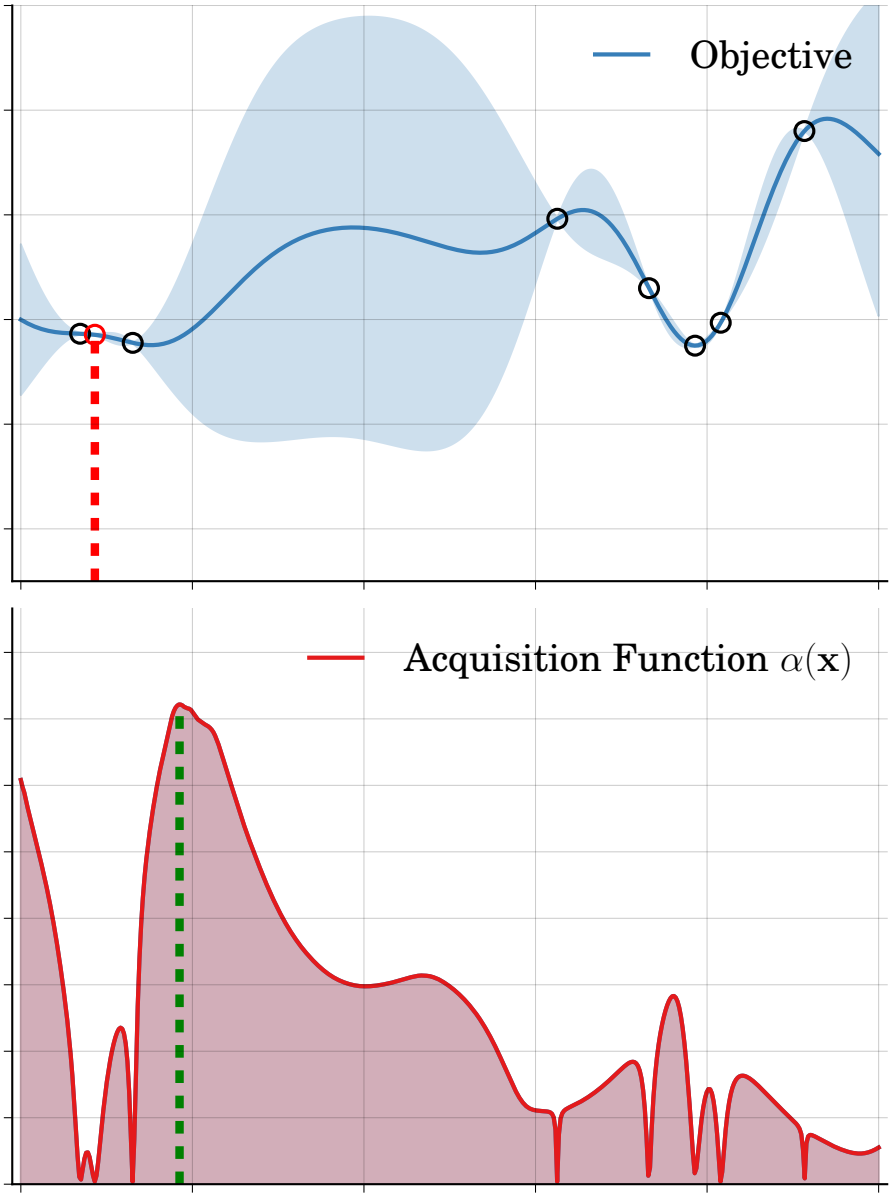
1. Get initial sample.
2. Fit a model to the data:

$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$

3. **Select data collection strategy:**

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[U(y|\mathbf{x}, \mathcal{D}_n)]\,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**



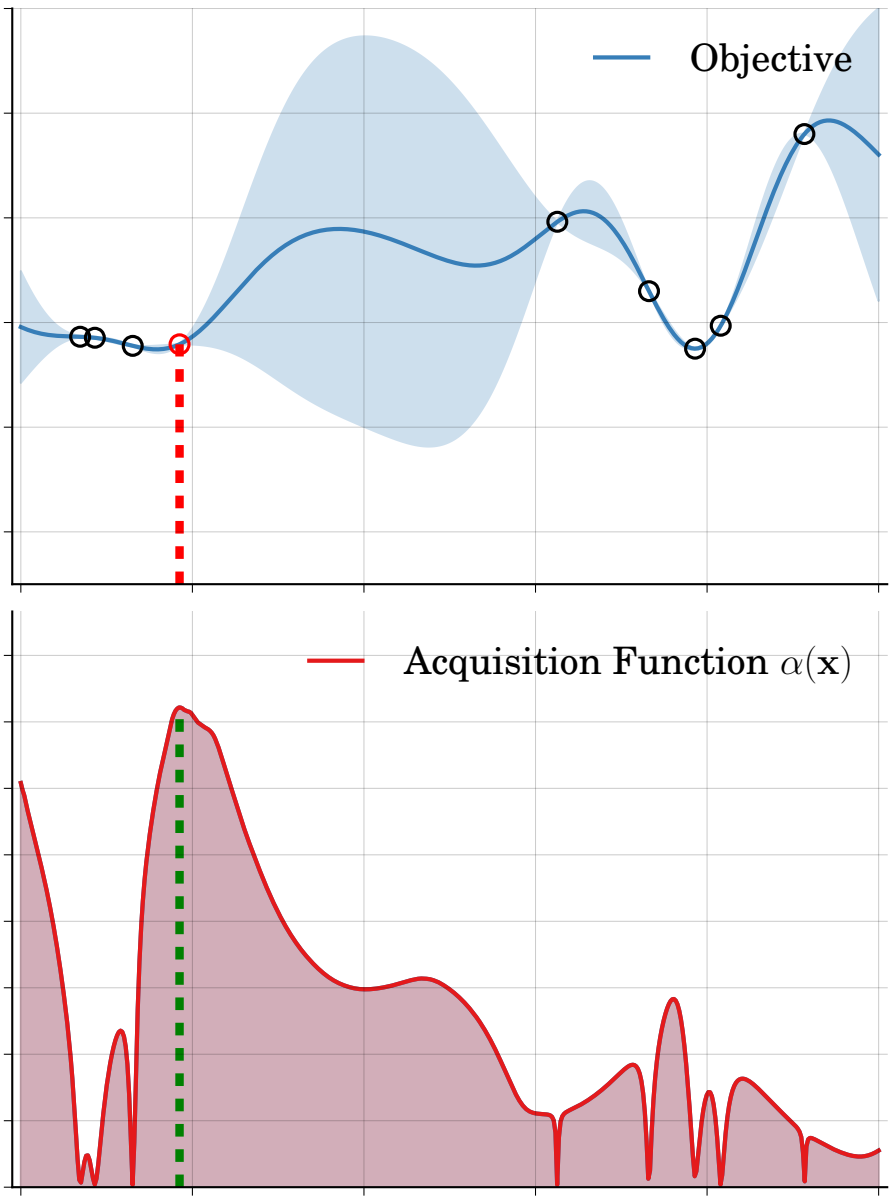
1. Get initial sample.
2. Fit a model to the data:

$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$

3. **Select data collection strategy:**

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[U(y|\mathbf{x}, \mathcal{D}_n)]\,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**



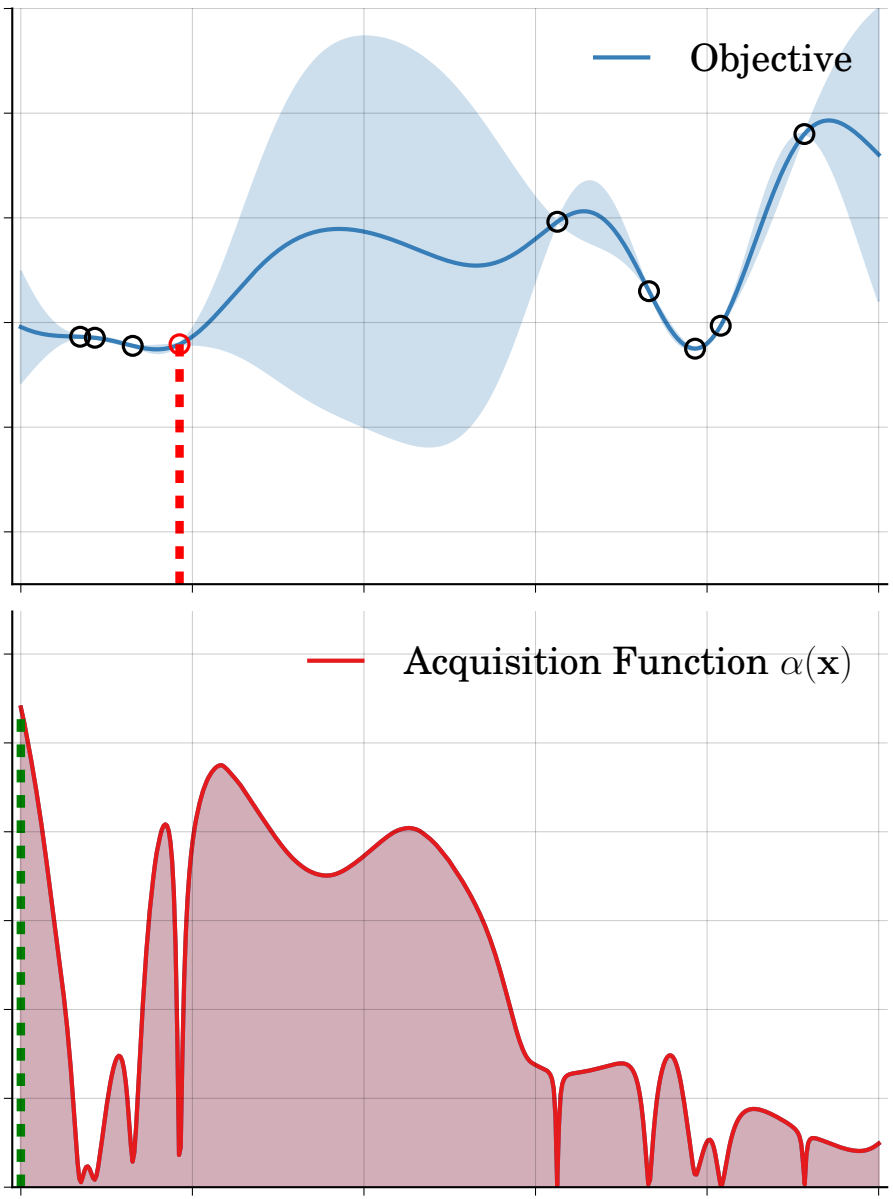
1. Get initial sample.
2. Fit a model to the data:

$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$

3. **Select data collection strategy:**

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[U(y|\mathbf{x}, \mathcal{D}_n)]\,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**



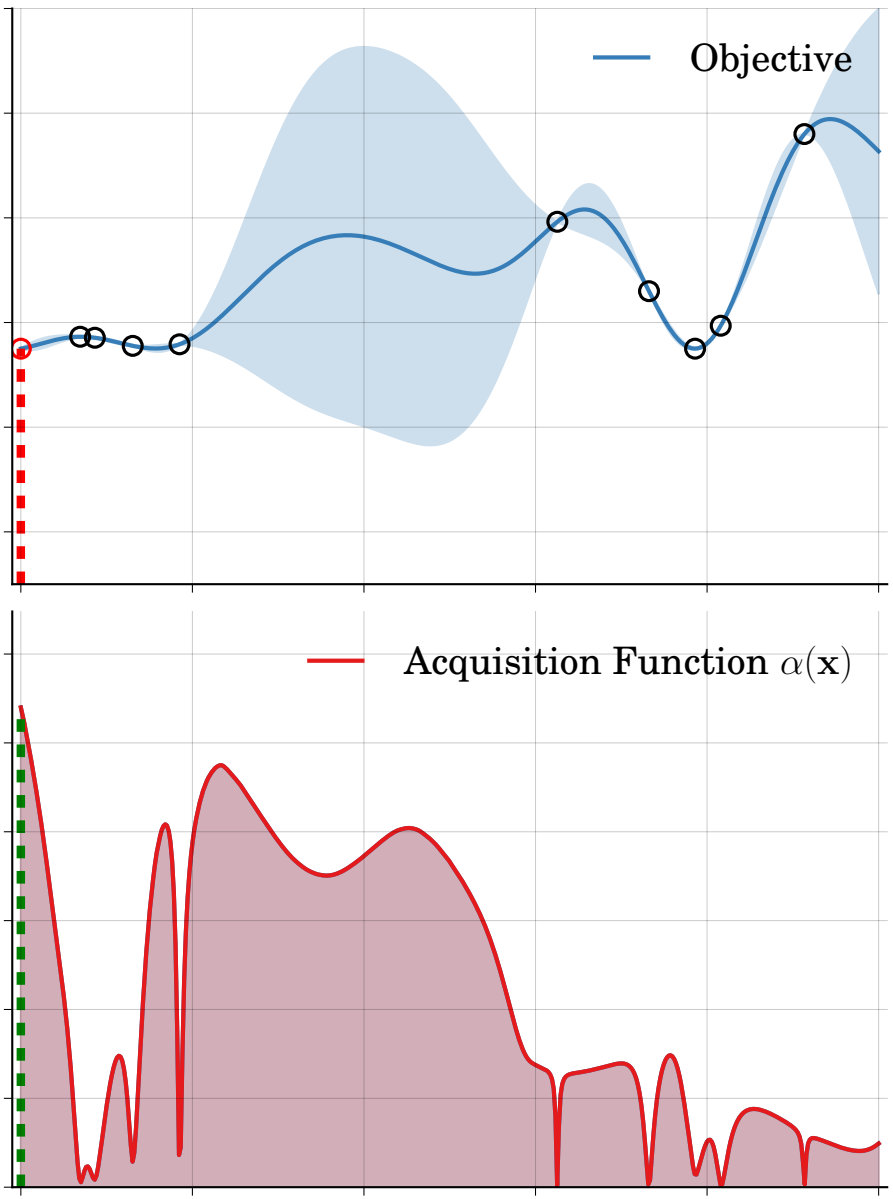
1. Get initial sample.
2. Fit a model to the data:

$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$

3. **Select data collection strategy:**

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[U(y|\mathbf{x}, \mathcal{D}_n)]\,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**



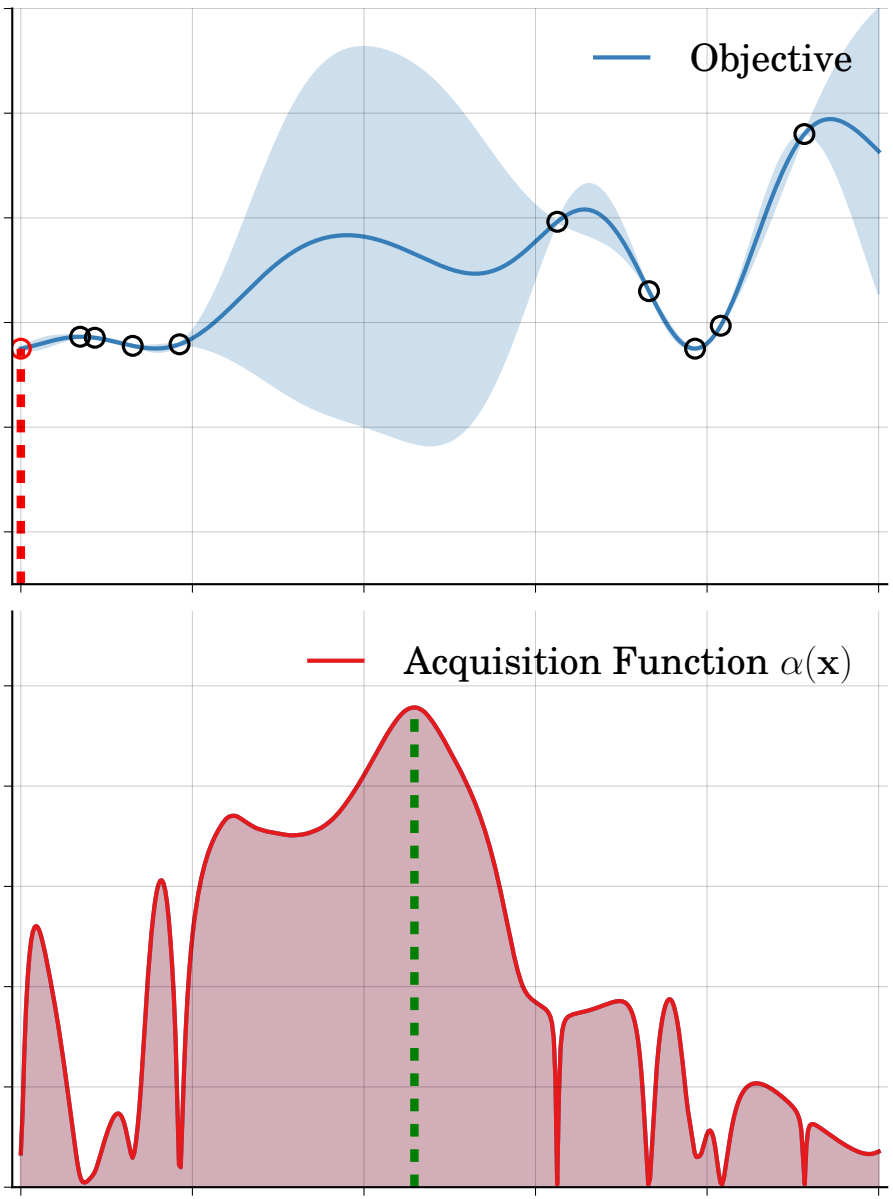
1. Get initial sample.
2. Fit a model to the data:

$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$

3. **Select data collection strategy:**

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[U(y|\mathbf{x}, \mathcal{D}_n)]\,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**



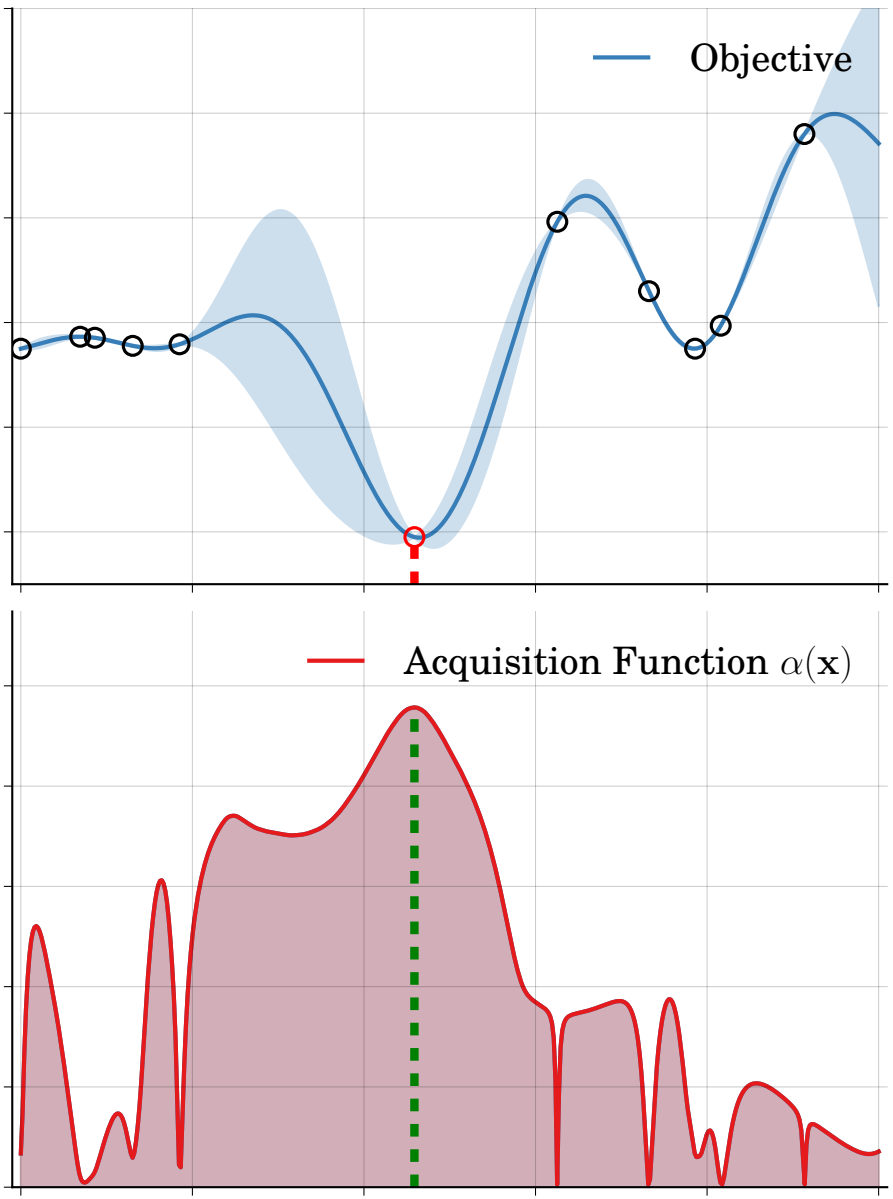
1. Get initial sample.
2. Fit a model to the data:

$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$

3. **Select data collection strategy:**

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[U(y|\mathbf{x}, \mathcal{D}_n)]\,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**



1. Get initial sample.
2. Fit a model to the data:

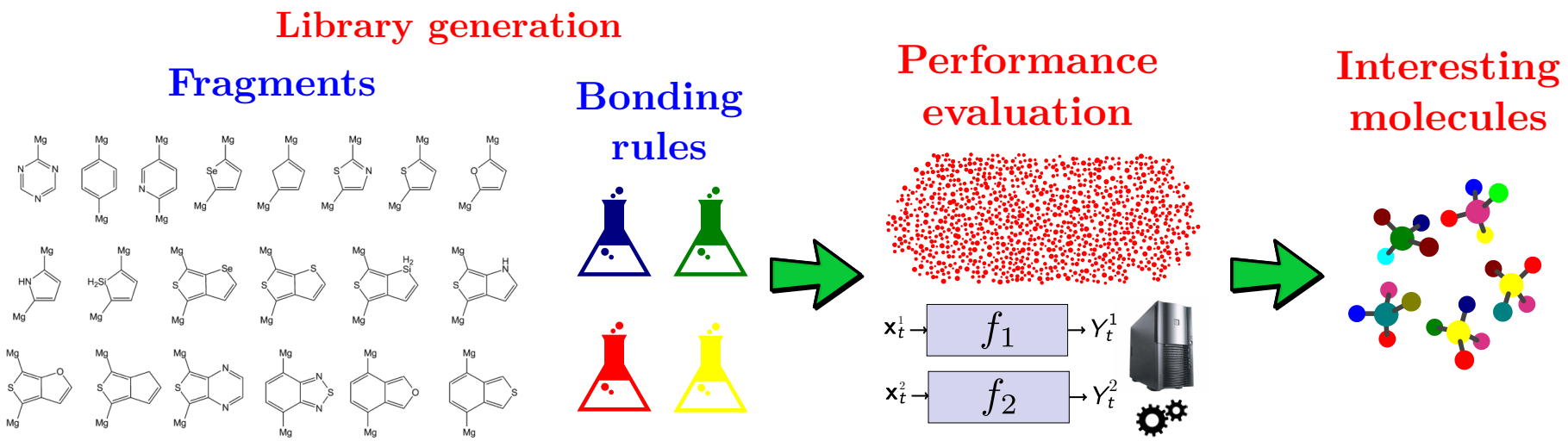
$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$

3. **Select data collection strategy:**

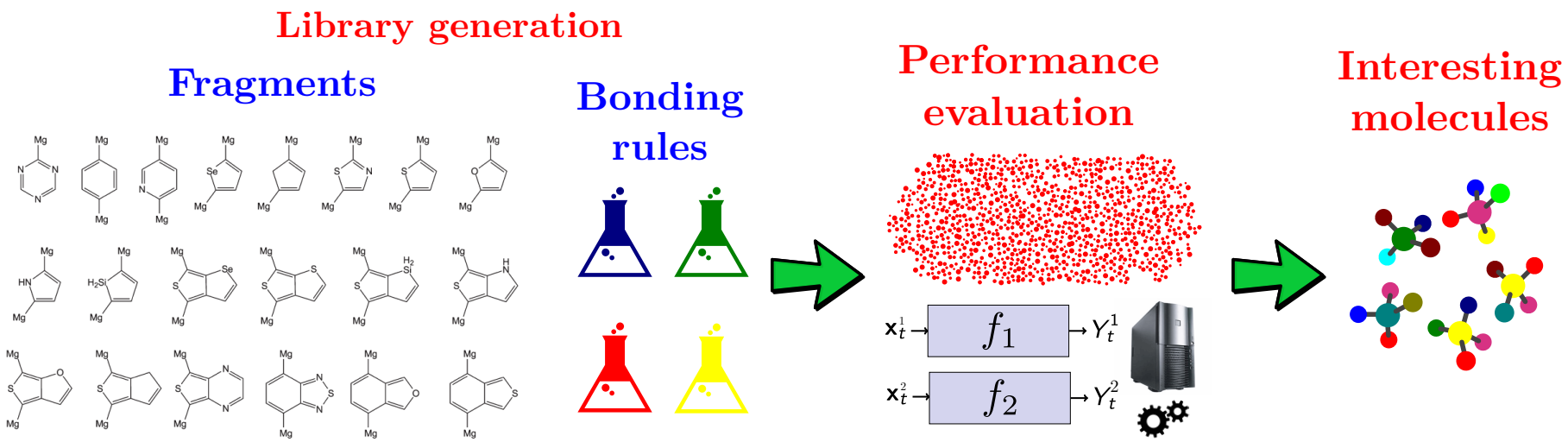
$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[U(y|\mathbf{x}, \mathcal{D}_n)]\,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**

Discovering new optimal molecules

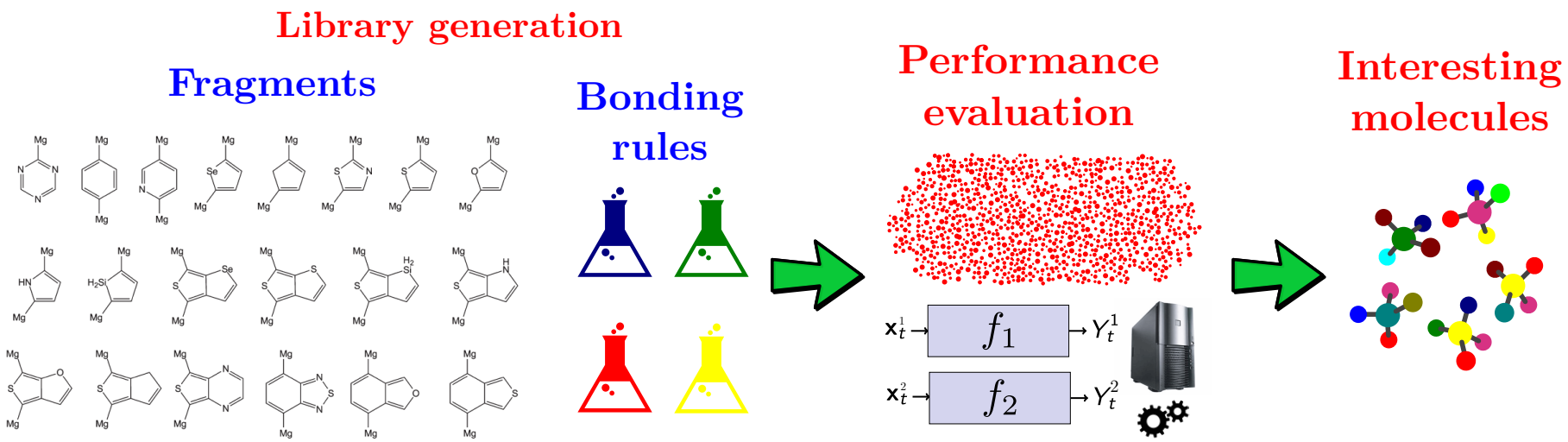


Discovering new optimal molecules



Bayesian optimization can **accelerate** the search!

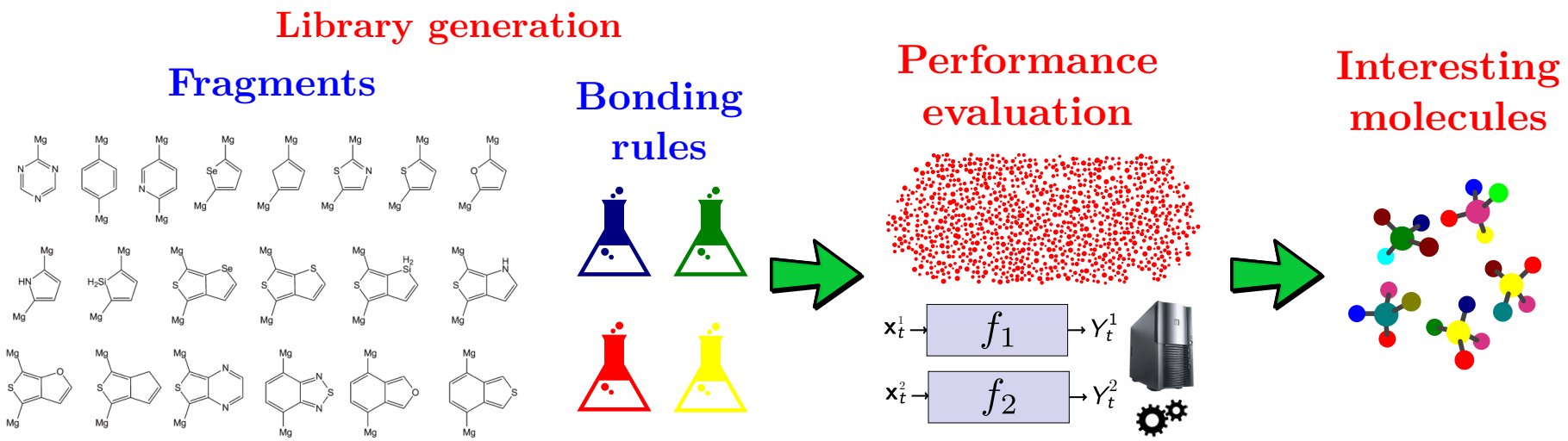
Discovering new optimal molecules



Bayesian optimization can **accelerate** the search!

Need **model** to map molecules to properties:

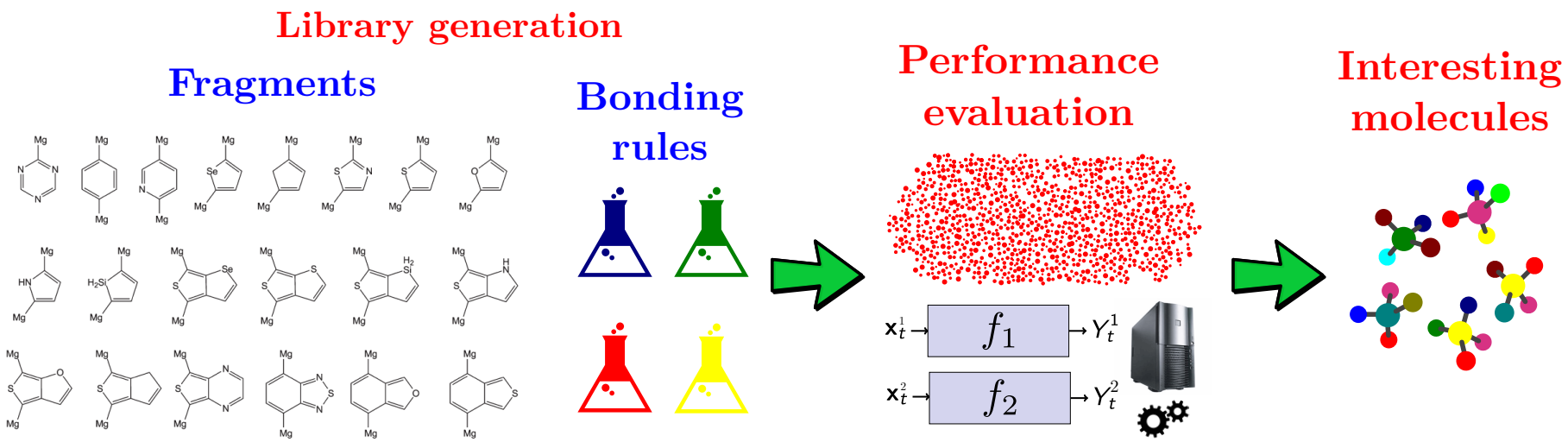
Discovering new optimal molecules



Bayesian optimization can **accelerate** the search!

Need **model** to map molecules to properties: **deep neural networks**.

Discovering new optimal molecules

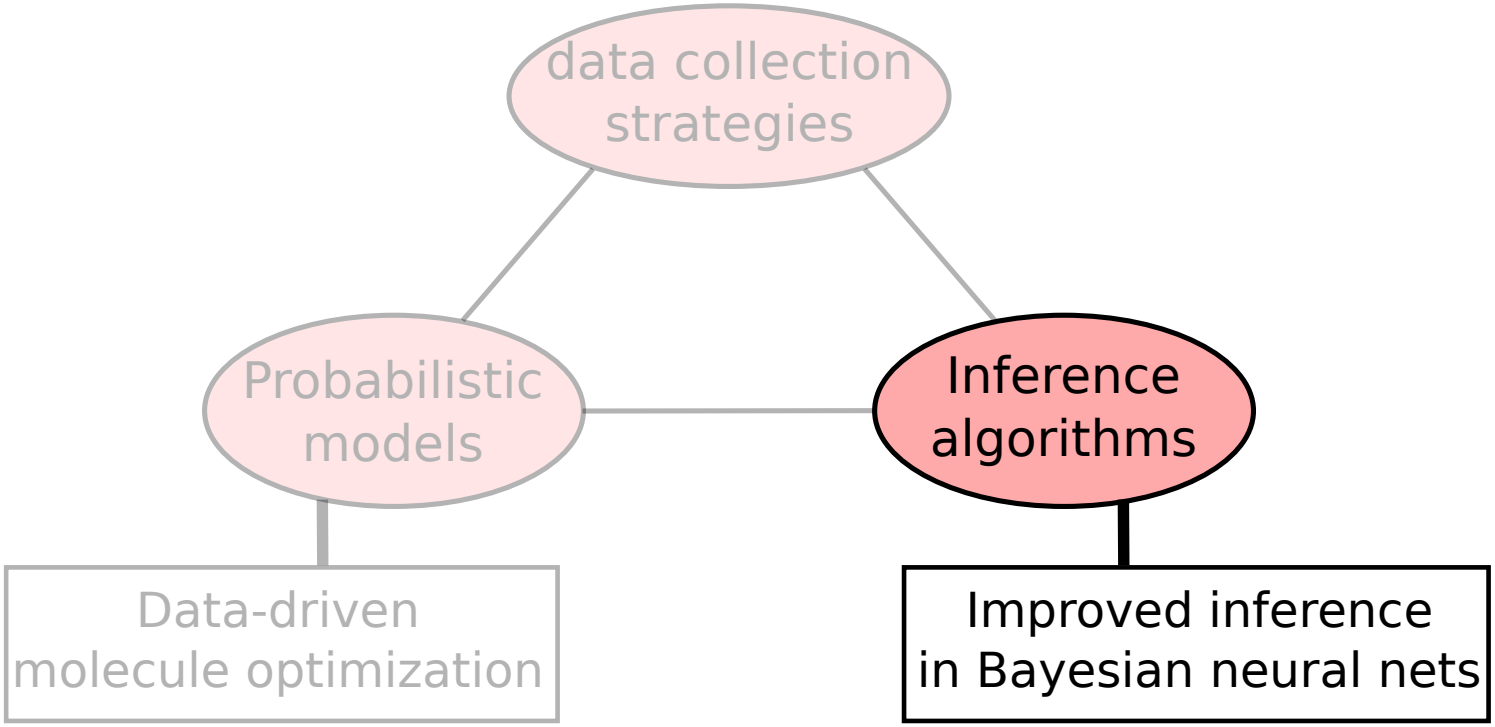


Bayesian optimization can **accelerate** the search!

Need **model** to map molecules to properties: **deep neural networks**.

Challenges:

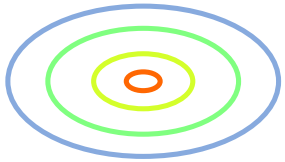
- **Probabilistic inference** is very difficult!
- Very complicated models, very large number of parameters.


data collection strategies
Probabilistic models
Inference algorithms
Data-driven molecule optimization
Improved inference in Bayesian neural nets

$$p(y|\text{Data}) = \int p(y|\theta)p(\theta|\text{Data})d\theta$$

Inference algorithm

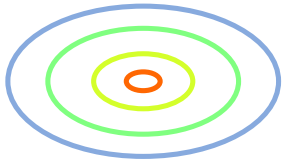
Approximation

$q(\theta) =$ 

$$p(y|\text{Data}) = \int p(y|\theta)p(\theta|\text{Data})d\theta$$

Inference algorithm

Approximation

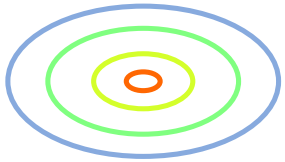
$q(\theta) =$ 

Minimize a divergence between the posterior p and q: $D_\alpha(p||q)$.

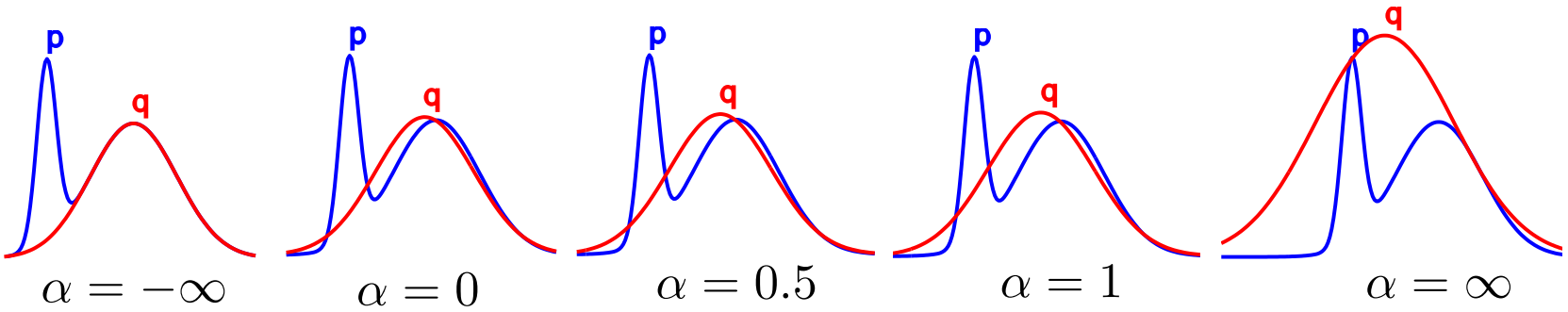
$$p(y|\text{Data}) = \int p(y|\theta)p(\theta|\text{Data})d\theta$$

Inference algorithm

Approximation

$q(\theta) =$ 

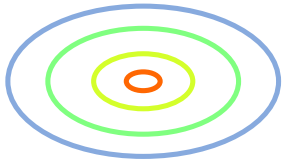
Minimize a divergence between the posterior p and q: $D_\alpha(p||q)$.



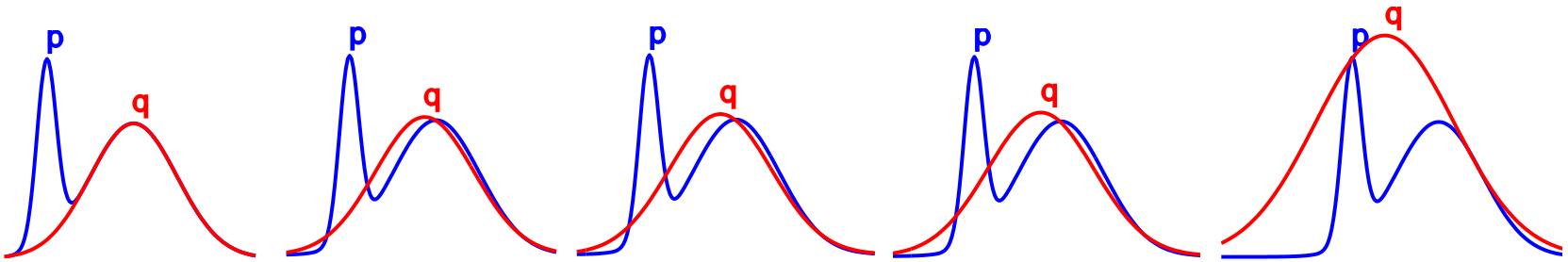
$$p(y|\text{Data}) = \int p(y|\theta)p(\theta|\text{Data})d\theta$$

Inference algorithm

Approximation

$q(\theta) =$ 

Minimize a divergence between the posterior p and q: $D_\alpha(p||q)$.



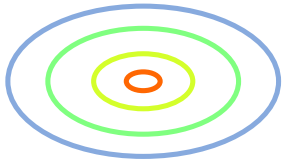
Variational Bayes

Expectation propagation

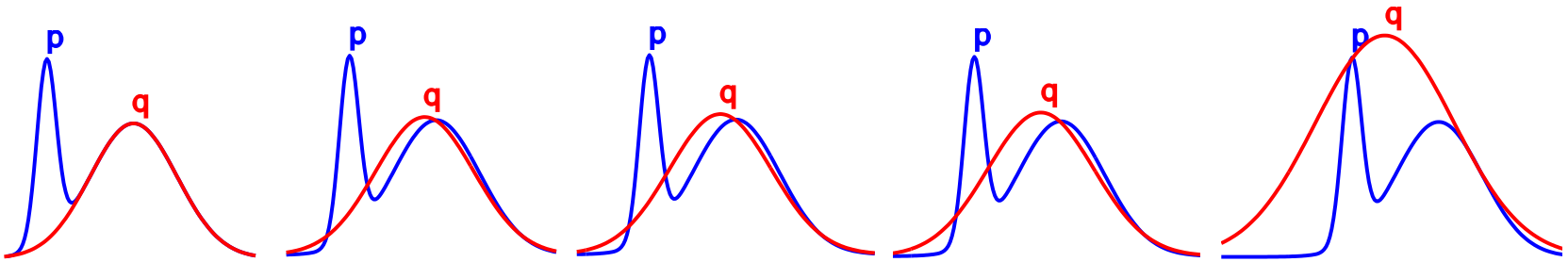
$$p(y|\text{Data}) = \int p(y|\theta)p(\theta|\text{Data})d\theta$$

Inference algorithm

Approximation

$q(\theta) =$ 

Minimize a divergence between the posterior p and q: $D_\alpha(p||q)$.

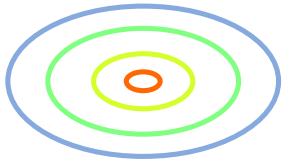


Variational Bayes

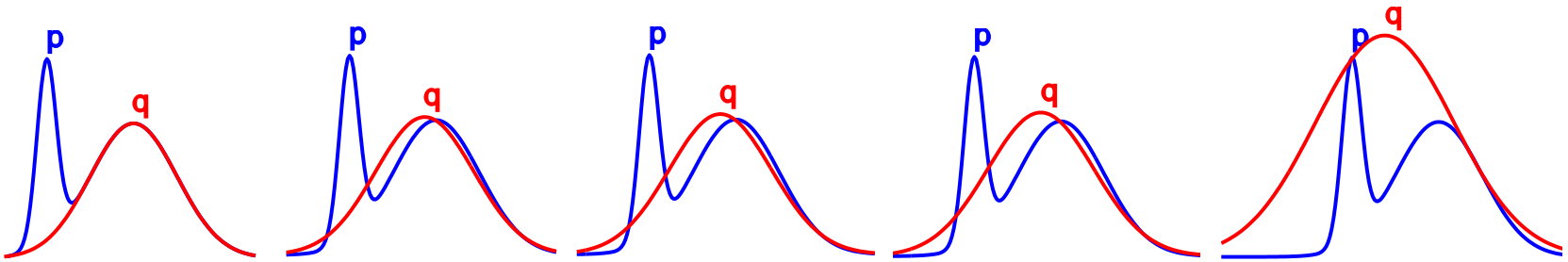
Expectation propagation

$$p(y|\text{Data}) = \int p(y|\theta)p(\theta|\text{Data})d\theta$$ **Inference algorithm**

Approximation

$q(\theta) =$ 

Minimize a divergence between the posterior p and q: $D_\alpha(p||q)$.



Variational Bayes

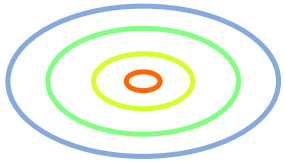
Expectation propagation

$$\min_q \; -\log Z_q - \sum_{n=1}^{N} \frac{1}{\alpha} \log \mathbb{E}_q \left[\left(\frac{f_n(\theta)}{\tilde{f}(\theta)} \right)^\alpha \right]$$

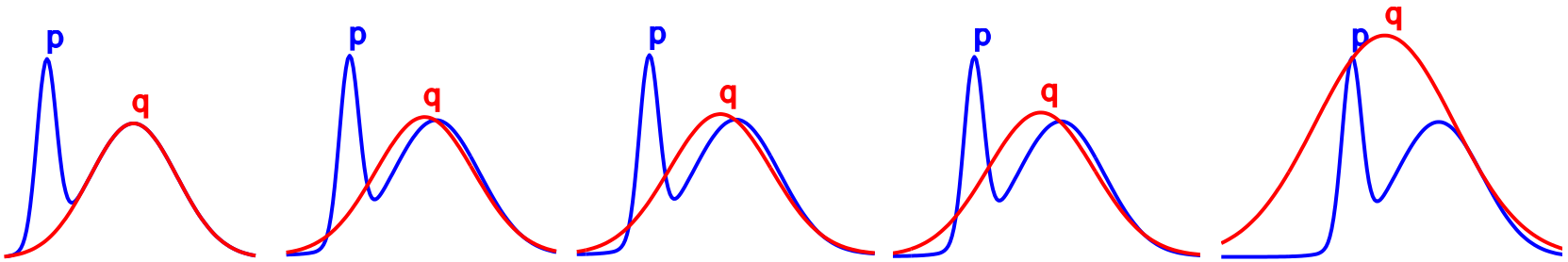
$$p(y|\text{Data}) = \int p(y|\theta)p(\theta|\text{Data})d\theta$$

Inference algorithm

Approximation

$q(\theta) =$ 

Minimize a divergence between the posterior p and q: $D_\alpha(p||q)$.



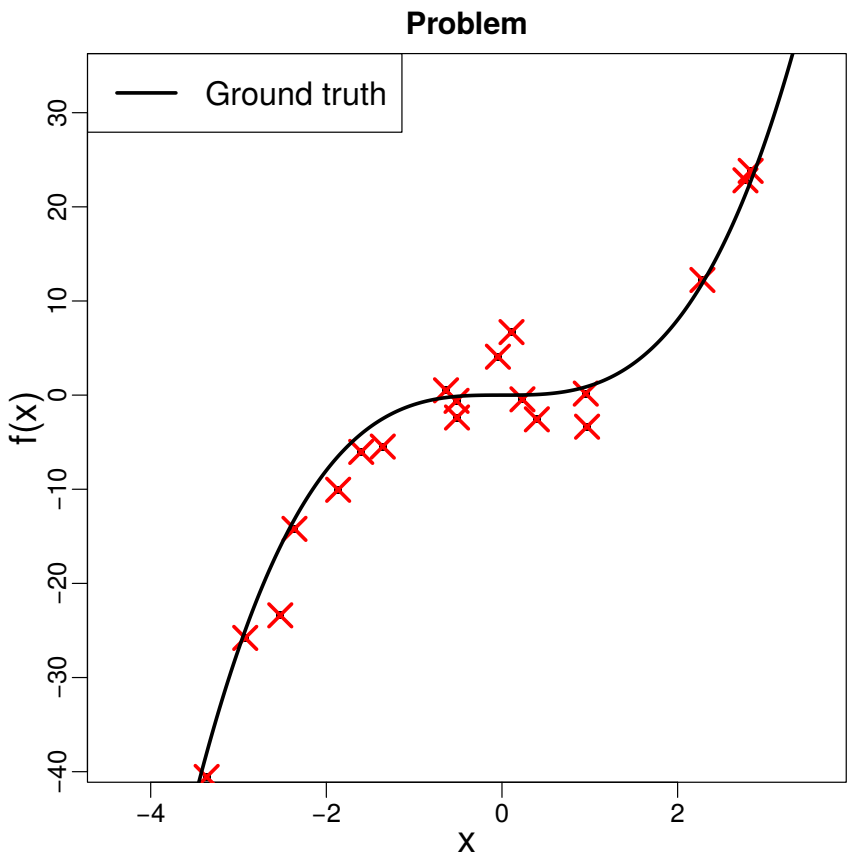
Variational Bayes

Expectation propagation

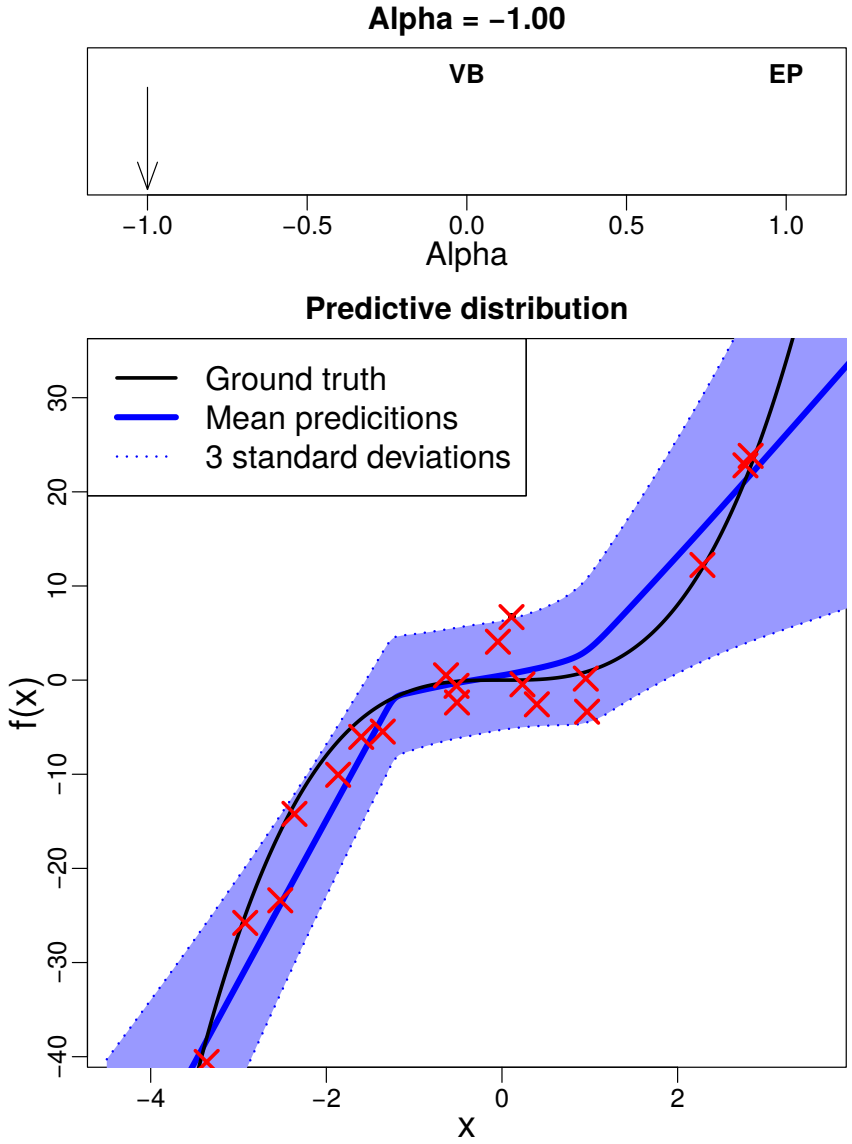
$$\min_q \; -\log Z_q - \sum_{n=1}^{N} \frac{1}{\alpha} \log \mathbb{E}_q \left[\left(\frac{f_n(\theta)}{\tilde{f}(\theta)} \right)^\alpha \right]$$

Easy to apply to complex models using **stochastic optimization** and **autodiff**.

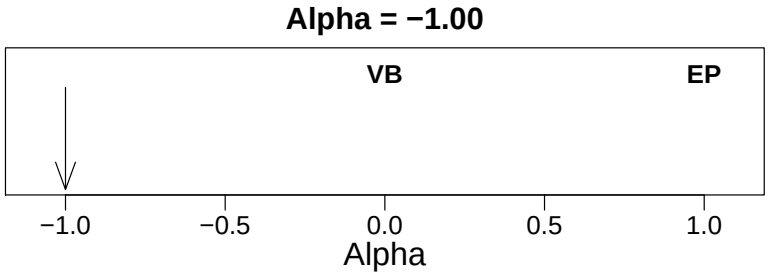
Example

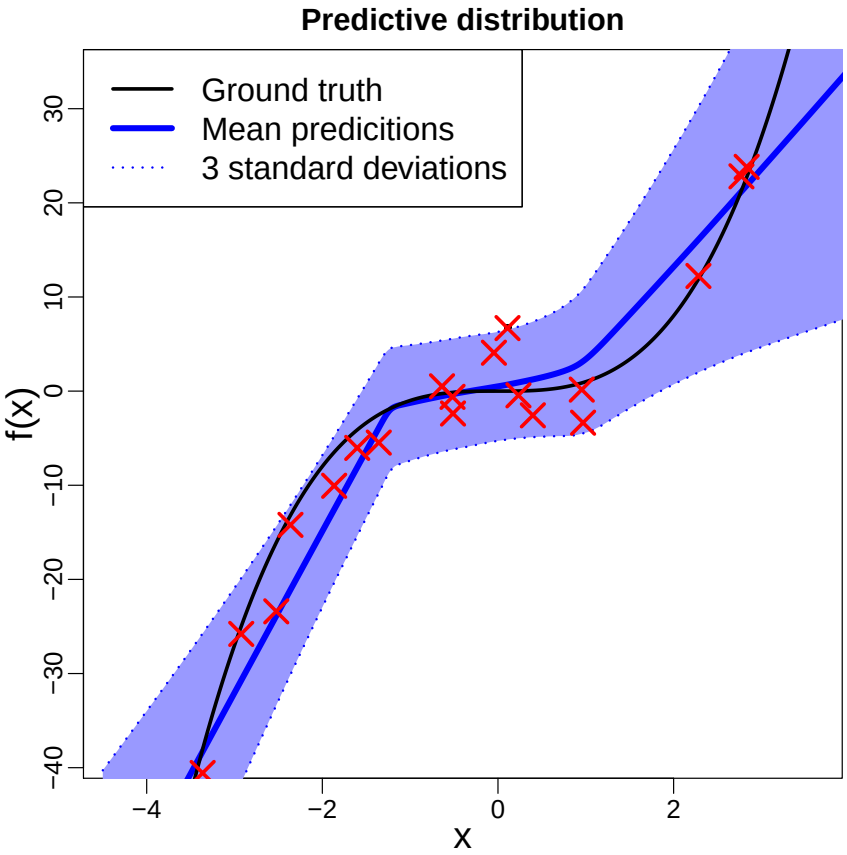


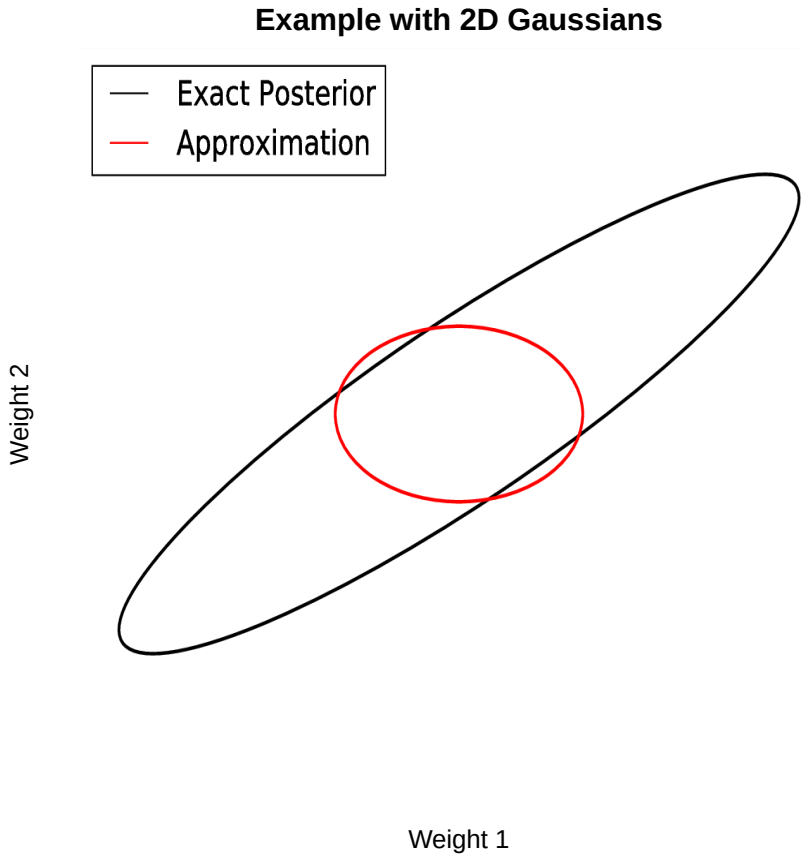
Example


Alpha = −1.00
VB
EP
−1.0
−0.5
0.0
0.5
1.0
Alpha
Predictive distribution
Ground truth
Mean predicitions
3 standard deviations
f(x)
30
20
10
0
−10
−20
−30
−40
−4
−2
0
2
x

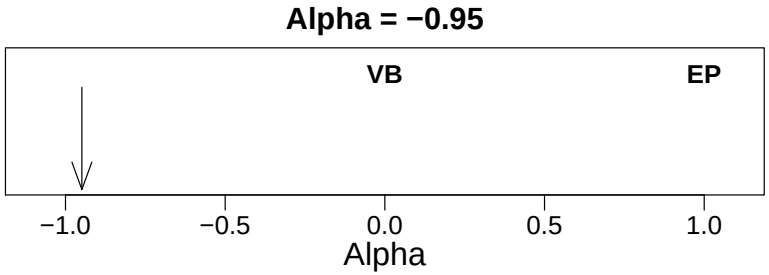
Example

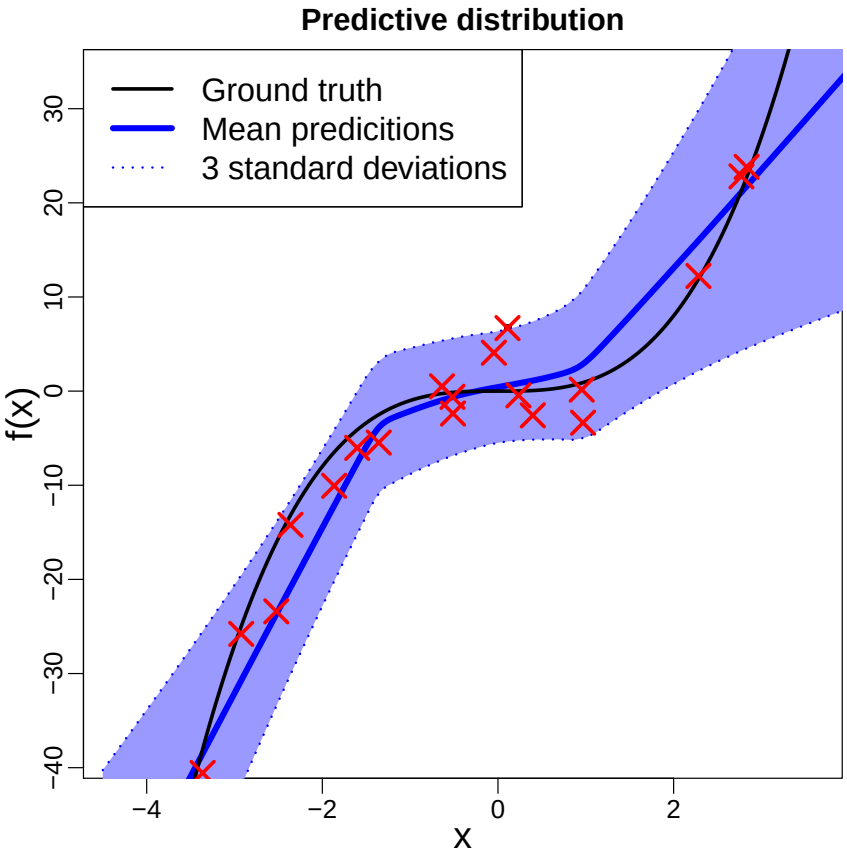

Alpha = −1.00
VB
EP
−1.0
−0.5
0.0
0.5
1.0
Alpha

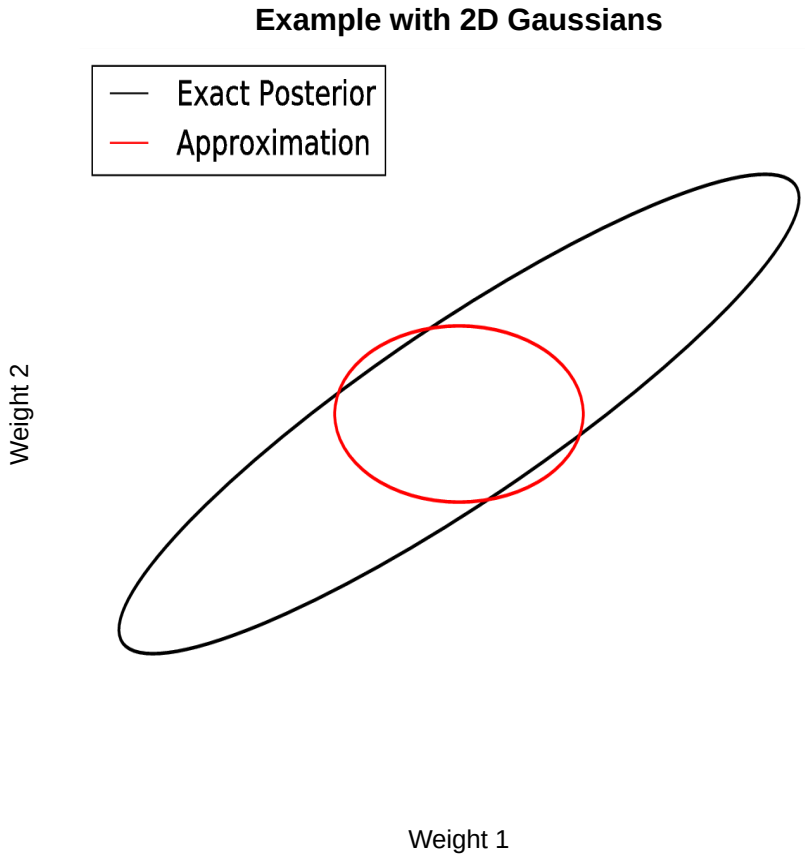

Predictive distribution
Ground truth
Mean predicitions
3 standard deviations
f(x)
30
20
10
0
−10
−20
−30
−40
−4
−2
0
2
X


Example with 2D Gaussians
Exact Posterior
Approximation
Weight 2
Weight 1

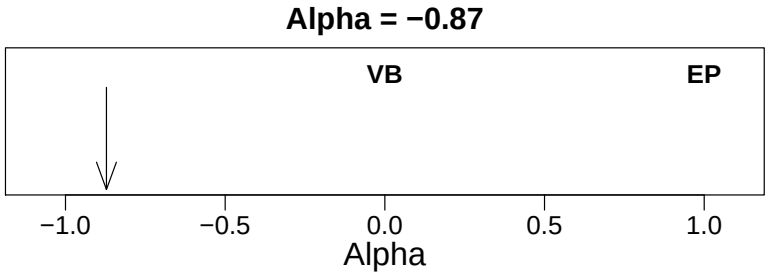
Example

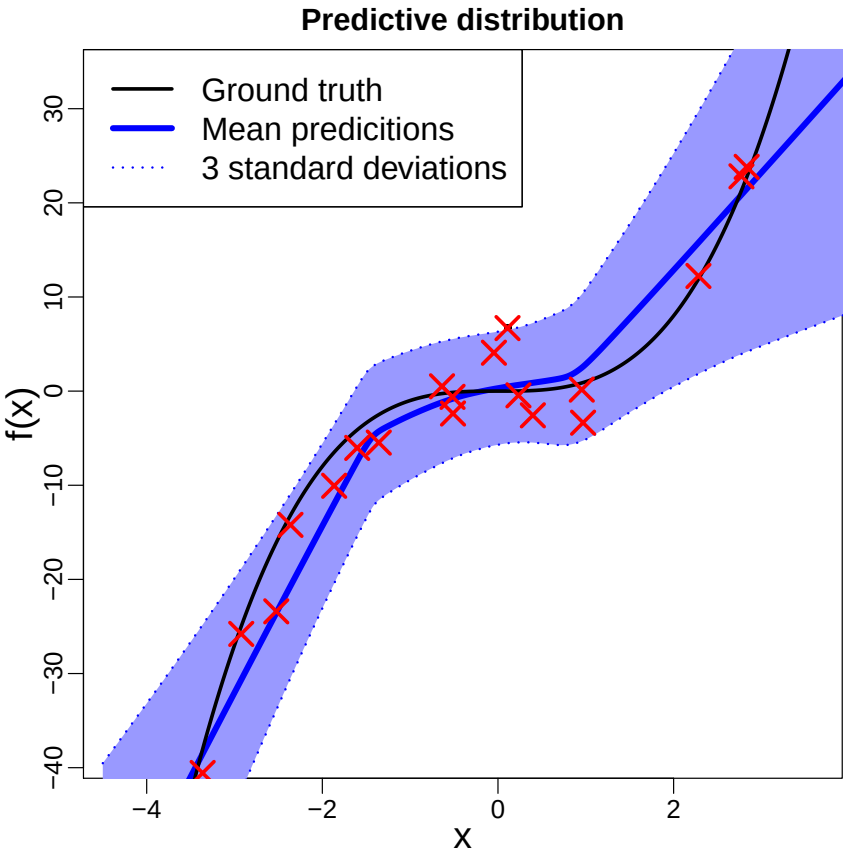

Alpha = −0.95
VB
EP
−1.0
−0.5
0.0
0.5
1.0
Alpha

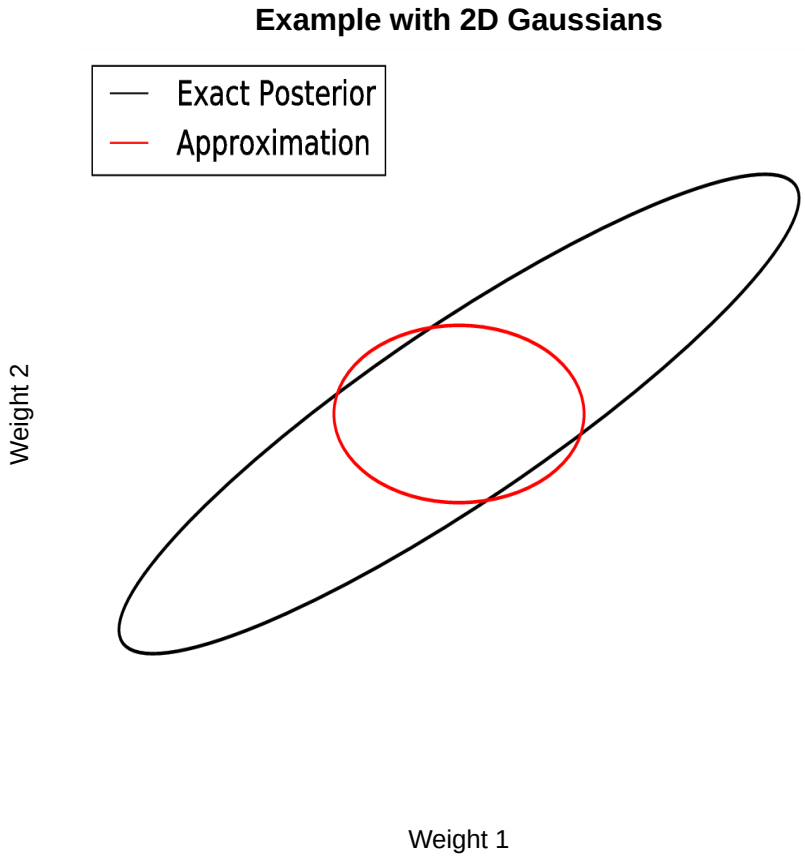

Predictive distribution
Ground truth
Mean predicitions
3 standard deviations
f(x)
30
20
10
0
−10
−20
−30
−40
−4
−2
0
2
X


Example with 2D Gaussians
Exact Posterior
Approximation
Weight 2
Weight 1

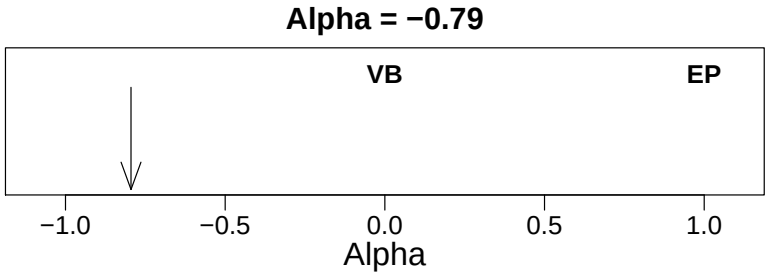
Example

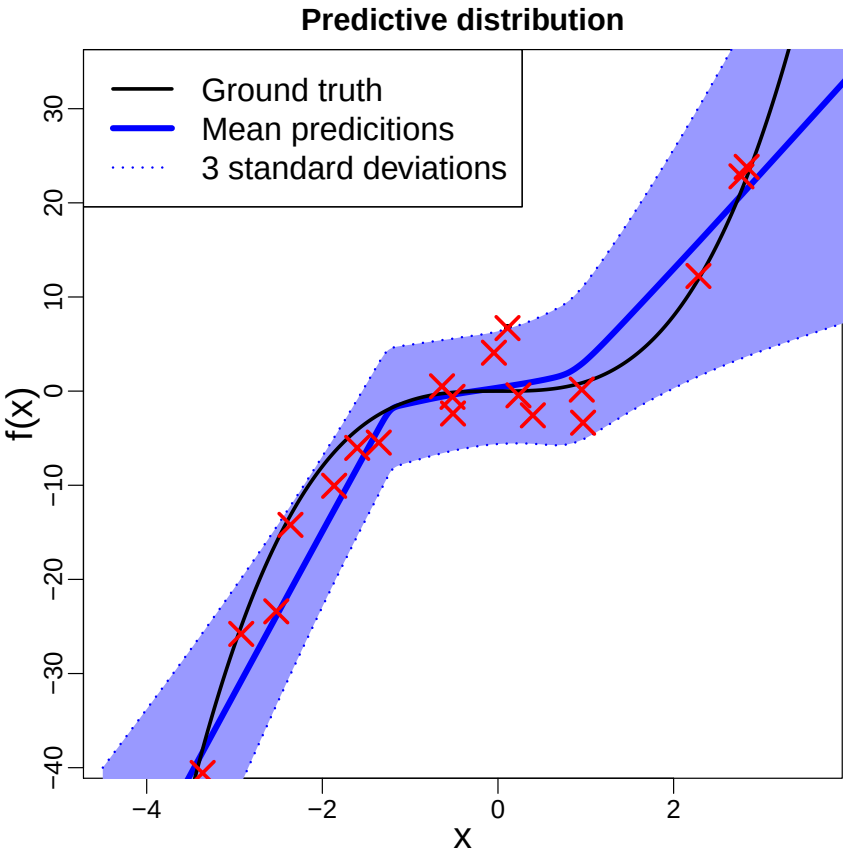

Alpha = −0.87
VB
EP
−1.0
−0.5
0.0
0.5
1.0
Alpha

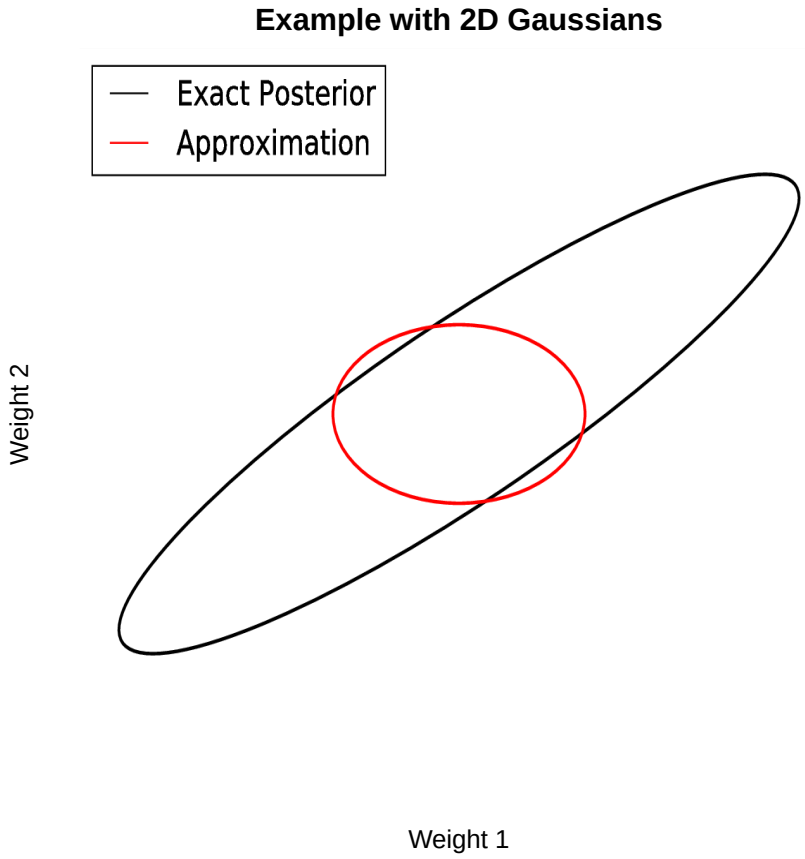

Predictive distribution
Ground truth
Mean predicitions
3 standard deviations
f(x)
30
20
10
0
−10
−20
−30
−40
−4
−2
0
2
X


Example with 2D Gaussians
Exact Posterior
Approximation
Weight 2
Weight 1

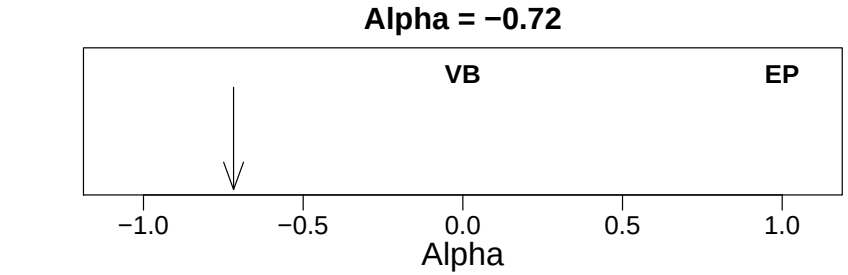
Example

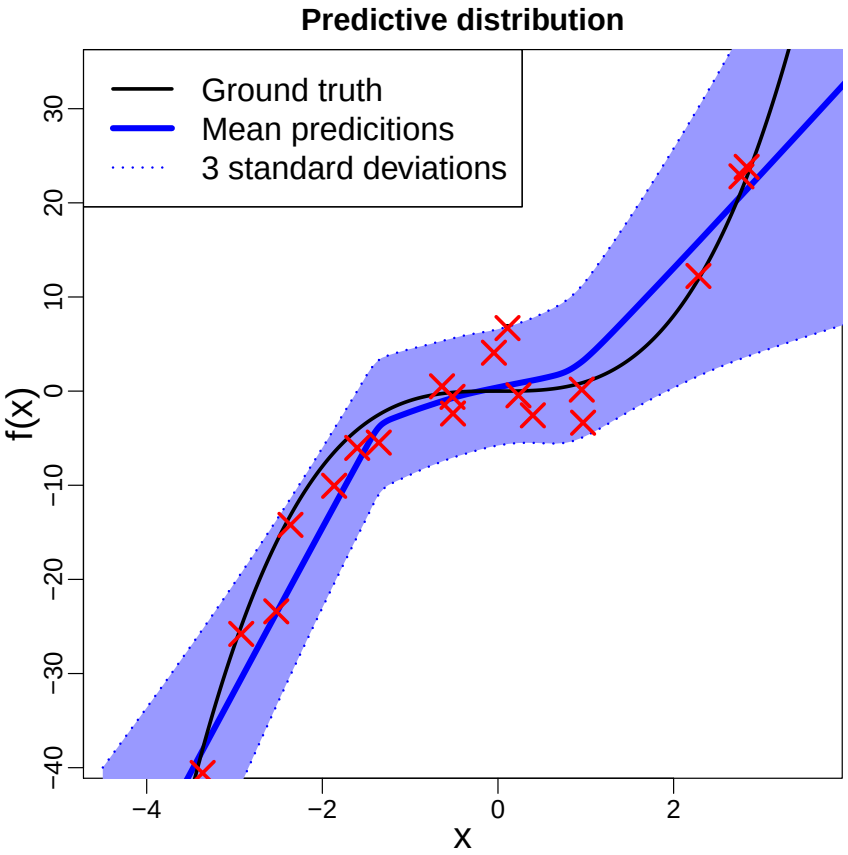

Alpha = −0.79
VB
EP
−1.0
−0.5
0.0
0.5
1.0
Alpha

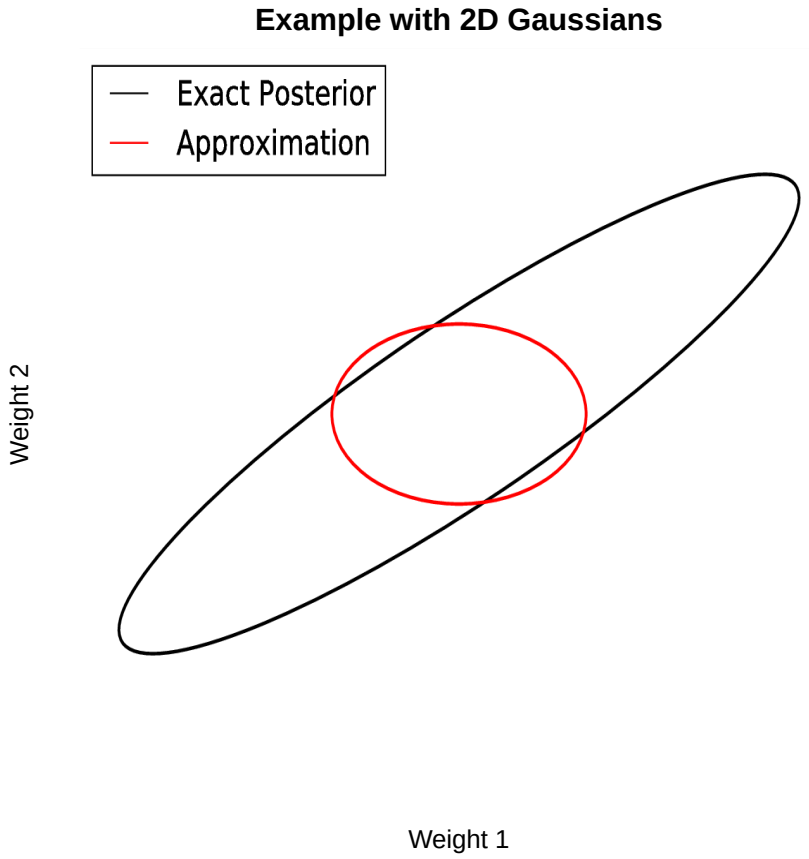

Predictive distribution
Ground truth
Mean predicitions
3 standard deviations
f(x)
30
20
10
0
−10
−20
−30
−40
−4
−2
0
2
X


Example with 2D Gaussians
Exact Posterior
Approximation
Weight 2
Weight 1

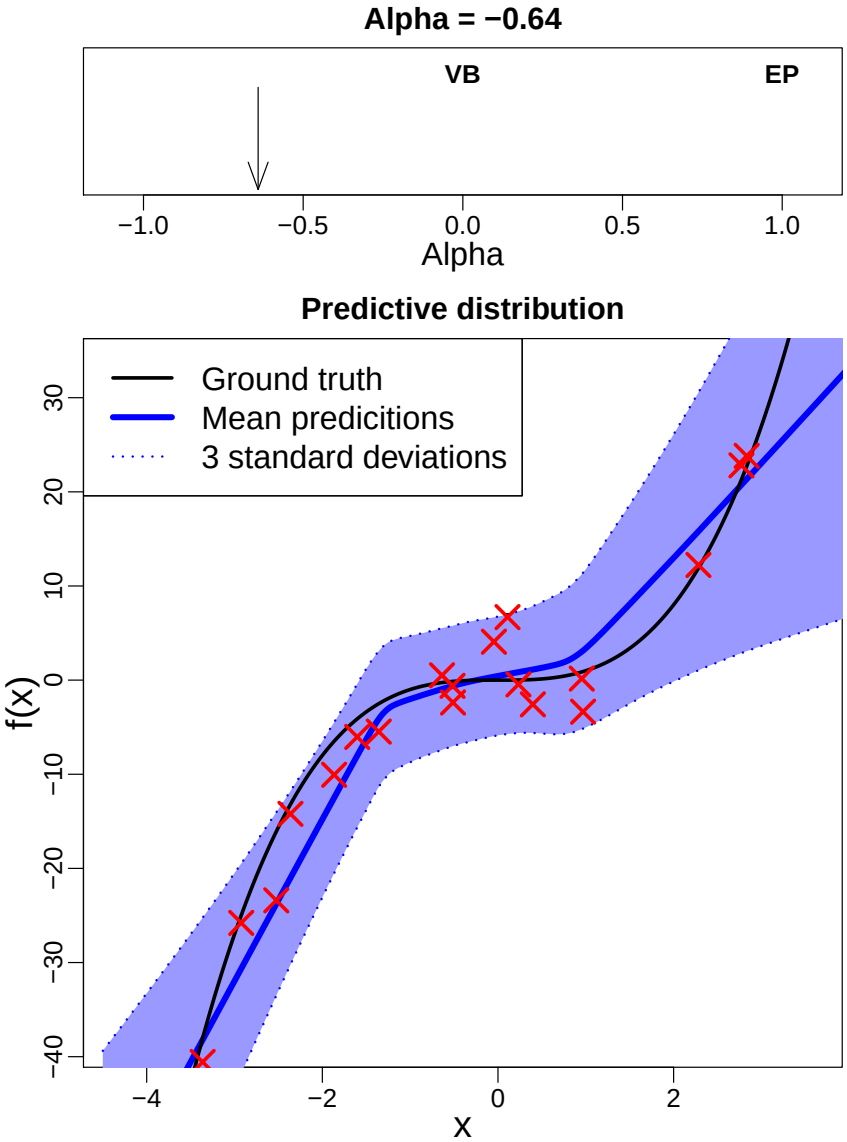
Example

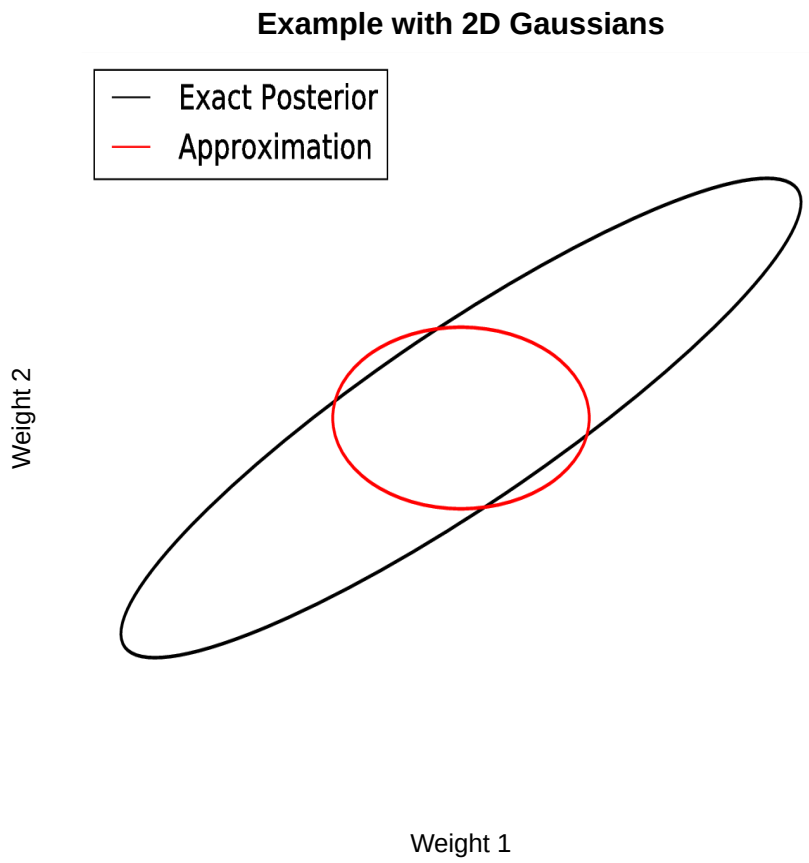

Alpha = −0.72
VB
EP
−1.0
−0.5
0.0
0.5
1.0
Alpha


Predictive distribution
Ground truth
Mean predicitions
3 standard deviations
f(x)
30
20
10
0
−10
−20
−30
−40
−4
−2
0
2
X

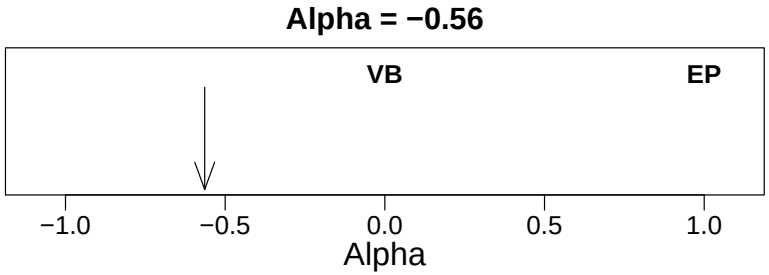

Example with 2D Gaussians
Exact Posterior
Approximation
Weight 2
Weight 1

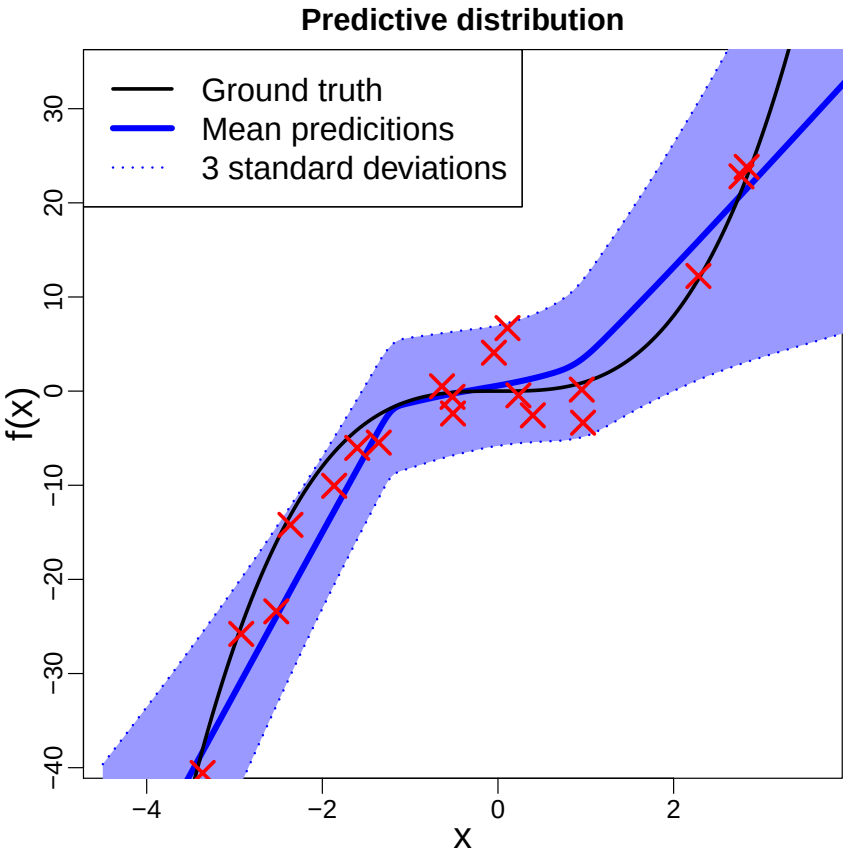
Example

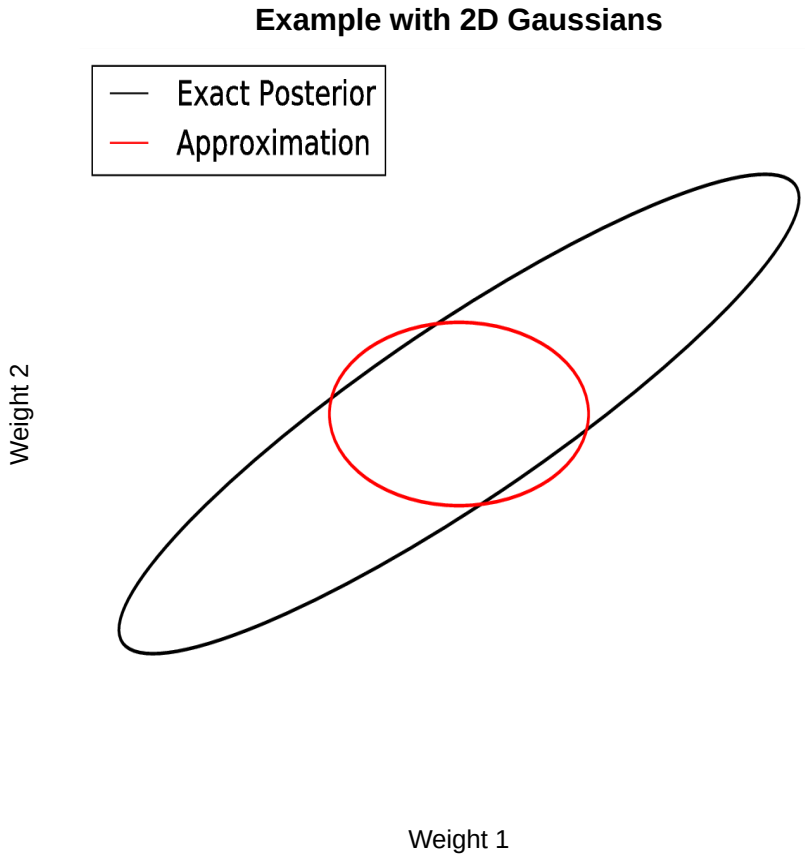

Alpha = −0.64
VB
EP
−1.0
−0.5
0.0
0.5
1.0
Alpha
Predictive distribution
Ground truth
Mean predicitions
3 standard deviations
f(x)
30
20
10
0
−10
−20
−30
−40
−4
−2
0
2
X


Example with 2D Gaussians
Exact Posterior
Approximation
Weight 2
Weight 1

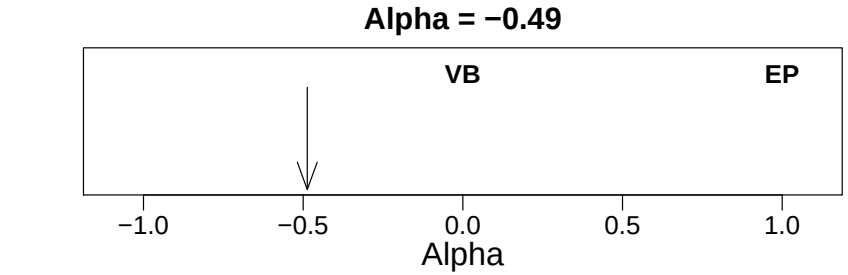
Example

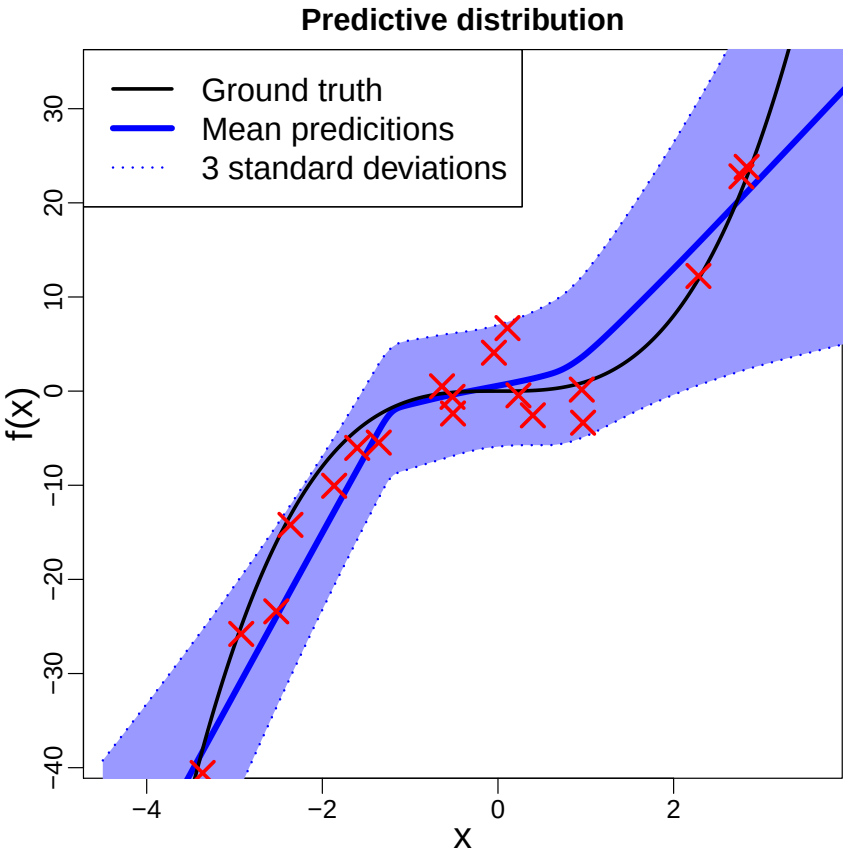

Alpha = −0.56
VB
EP
−1.0
−0.5
0.0
0.5
1.0
Alpha

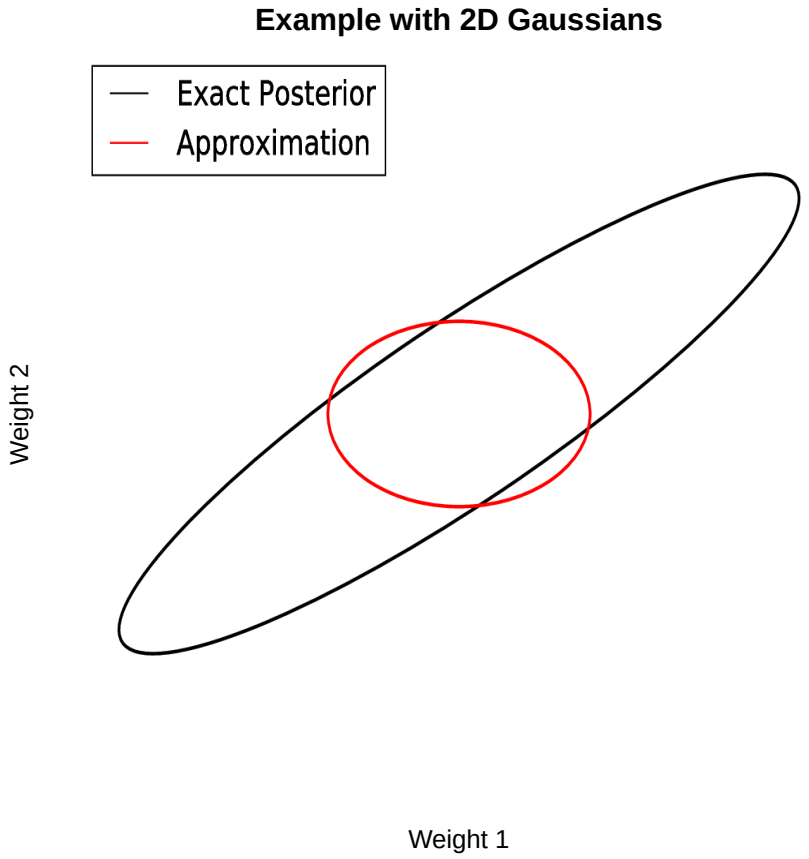

Predictive distribution
Ground truth
Mean predicitions
3 standard deviations
f(x)
30
20
10
0
−10
−20
−30
−40
−4
−2
0
2
X


Example with 2D Gaussians
Exact Posterior
Approximation
Weight 2
Weight 1

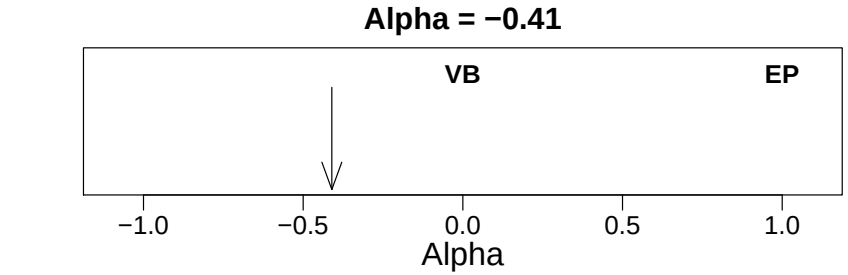
Example

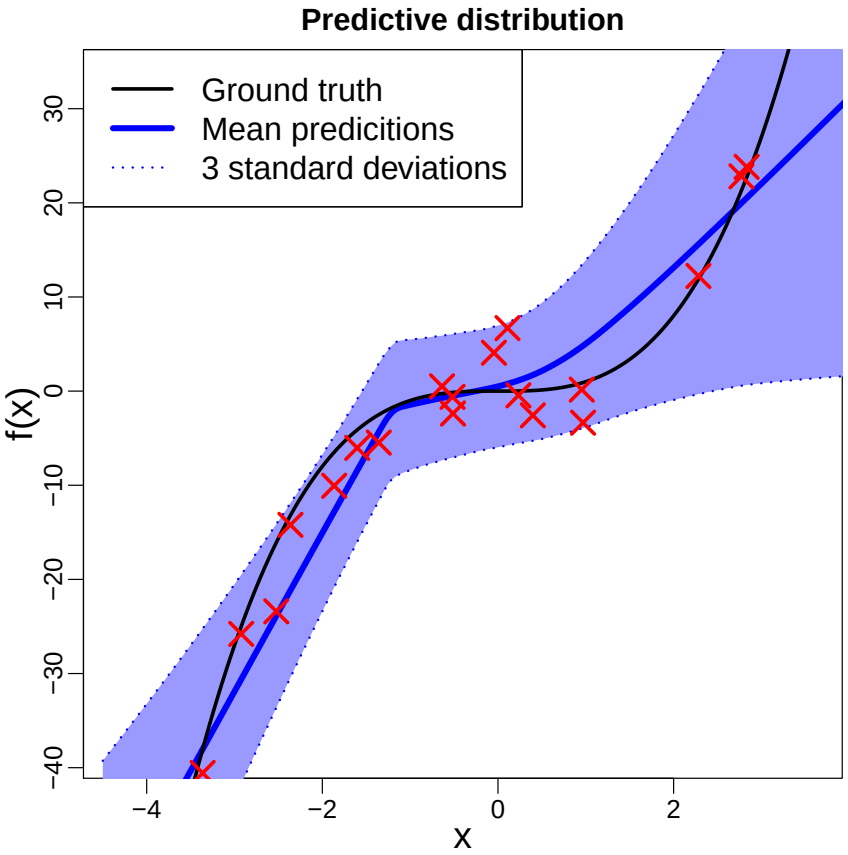

Alpha = −0.49
VB
EP
−1.0
−0.5
0.0
0.5
1.0
Alpha

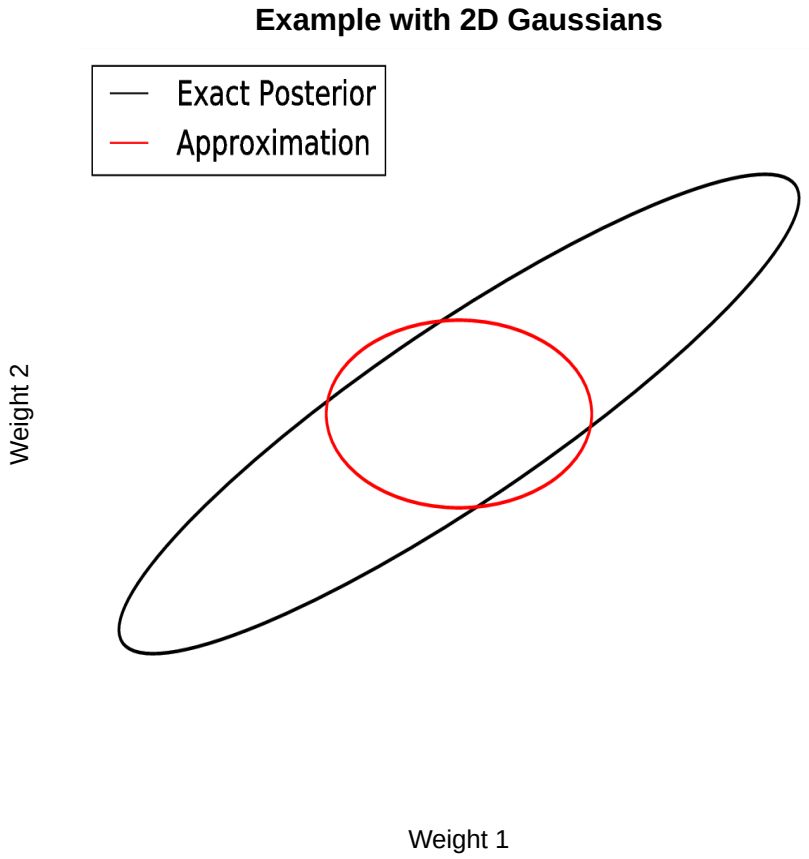

Predictive distribution
Ground truth
Mean predicitions
3 standard deviations
f(x)
30
20
10
0
−10
−20
−30
−40
−4
−2
0
2
X


Example with 2D Gaussians
Exact Posterior
Approximation
Weight 2
Weight 1

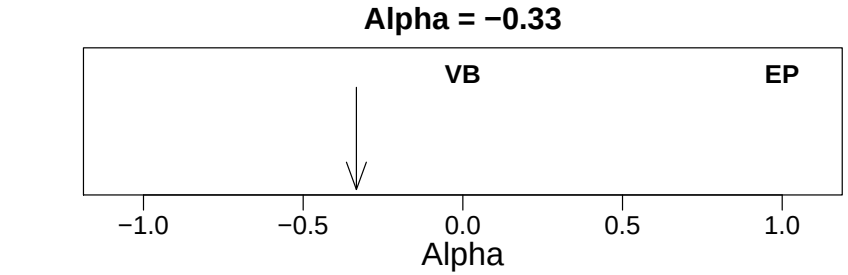
Example

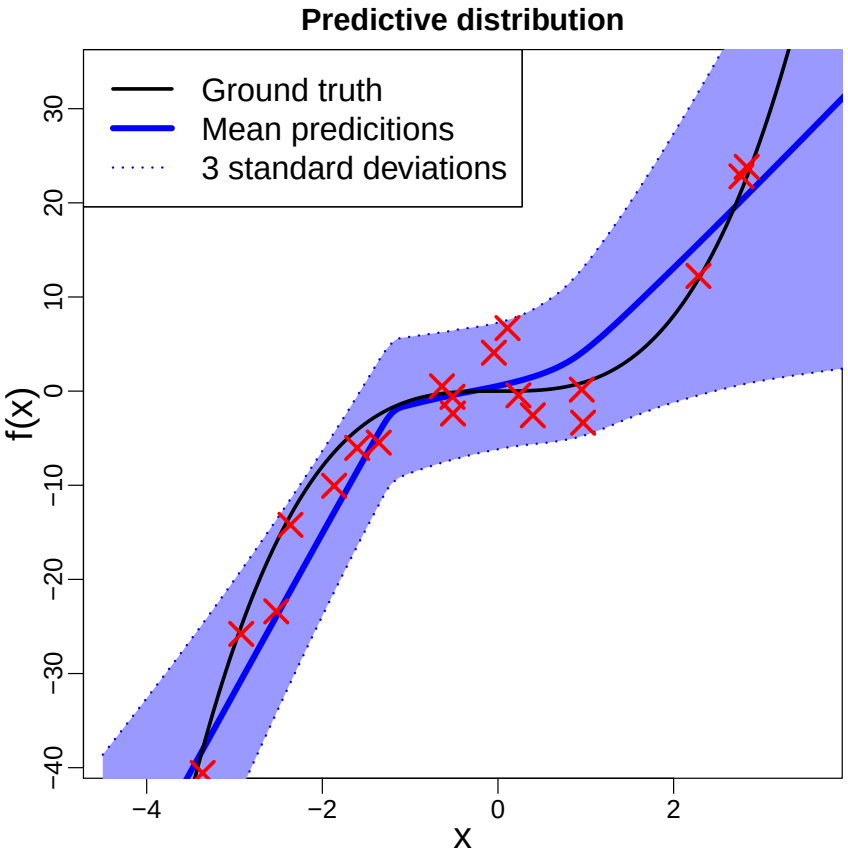

Alpha = −0.41
VB
EP
−1.0
−0.5
0.0
0.5
1.0
Alpha

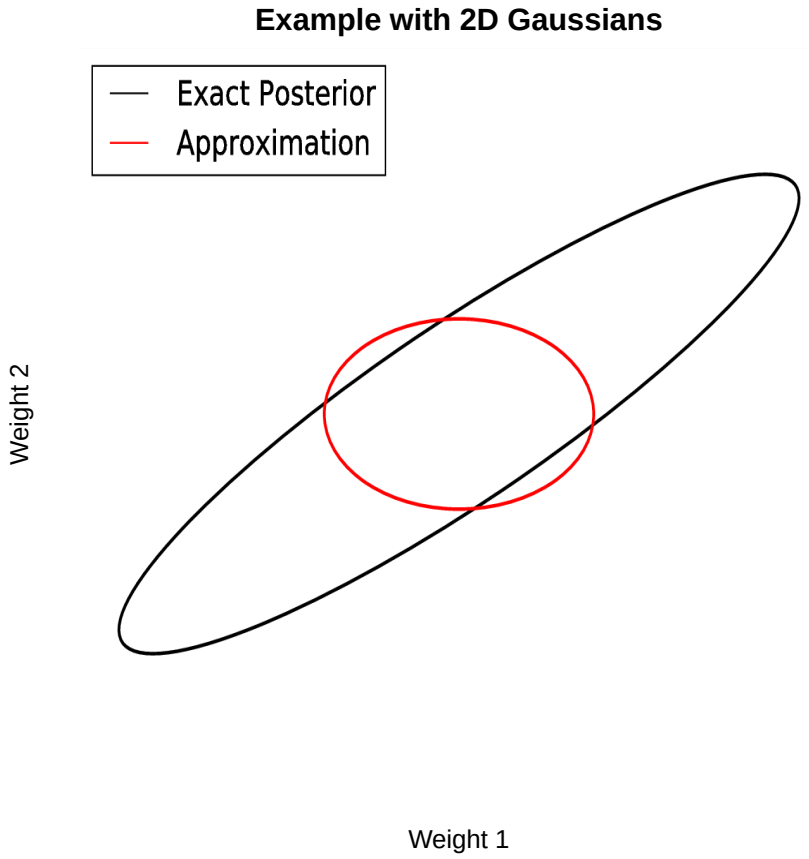

Predictive distribution
Ground truth
Mean predicitions
3 standard deviations
f(x)
30
20
10
0
−10
−20
−30
−40
−4
−2
0
2
X


Example with 2D Gaussians
Exact Posterior
Approximation
Weight 2
Weight 1

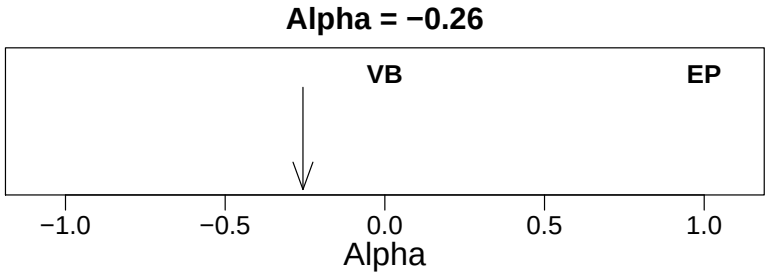
Example

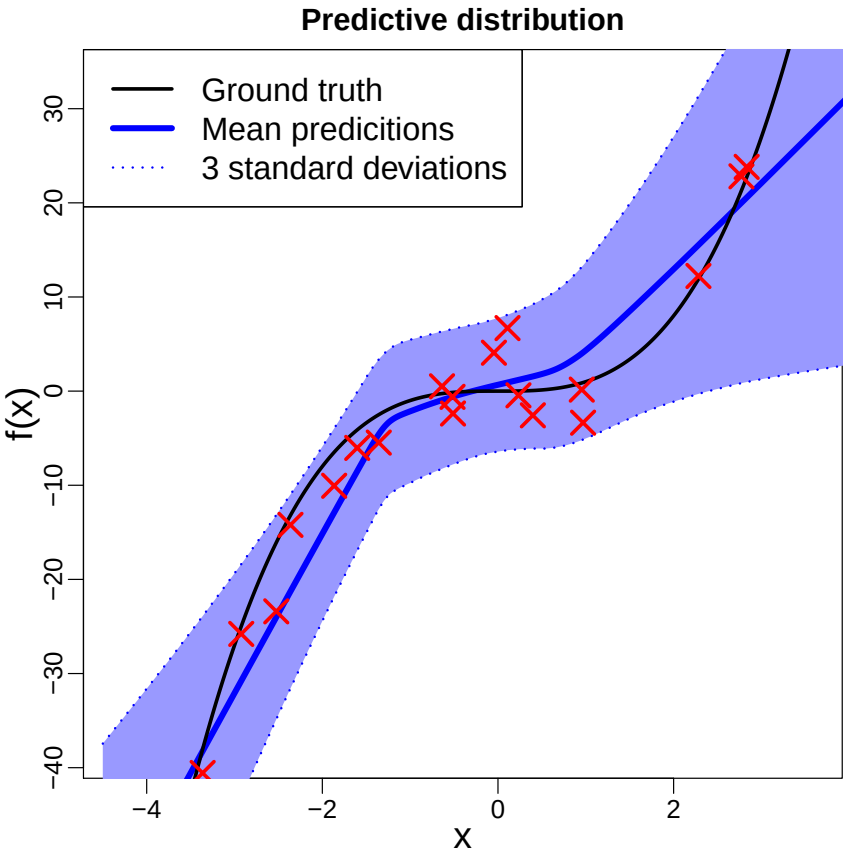

Alpha = −0.33
VB
EP
−1.0
−0.5
0.0
0.5
1.0
Alpha

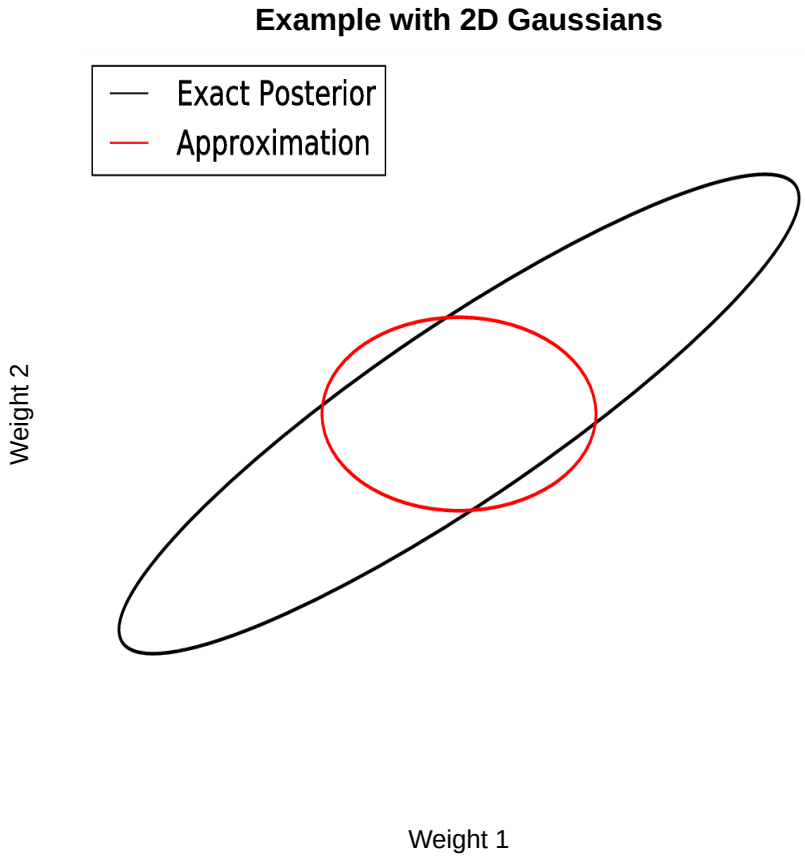

Predictive distribution
Ground truth
Mean predicitions
3 standard deviations
f(x)
30
20
10
0
−10
−20
−30
−40
−4
−2
0
2
X


Example with 2D Gaussians
Exact Posterior
Approximation
Weight 2
Weight 1

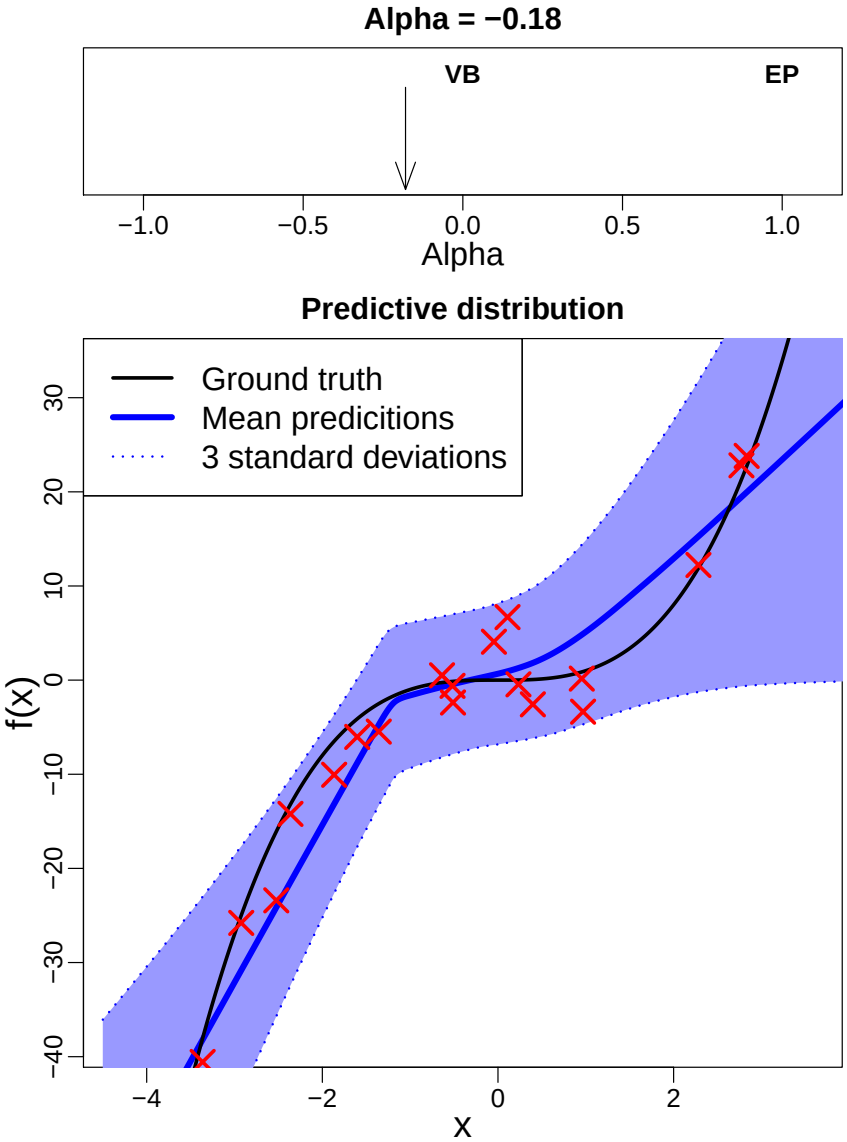
Example

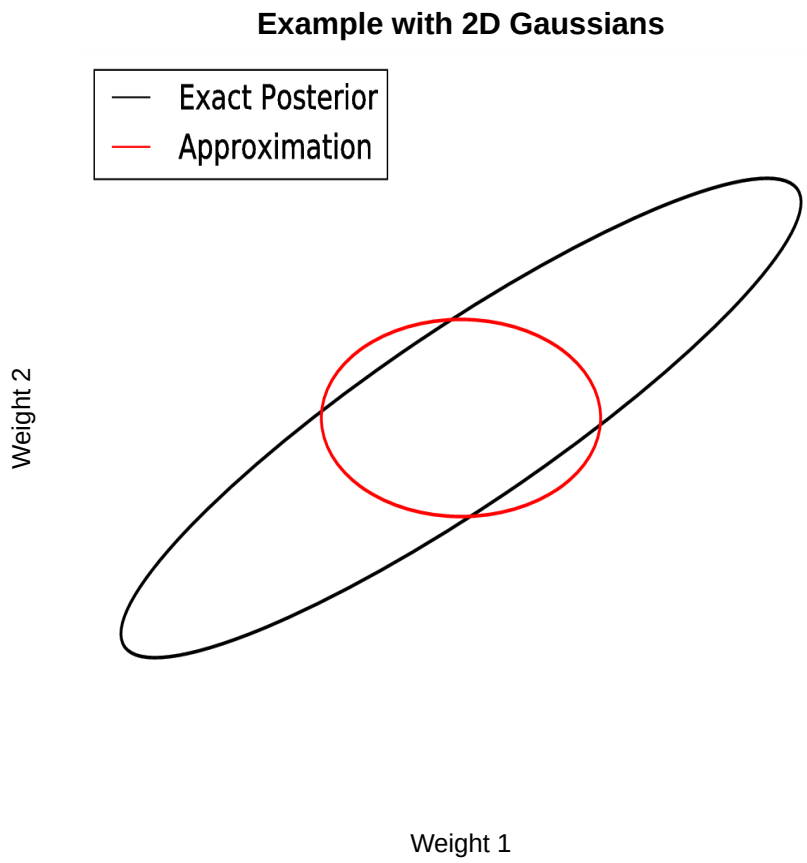

Alpha = −0.26
VB
EP
−1.0
−0.5
0.0
0.5
1.0
Alpha


Predictive distribution
Ground truth
Mean predicitions
3 standard deviations
f(x)
30
20
10
0
−10
−20
−30
−40
−4
−2
0
2
X

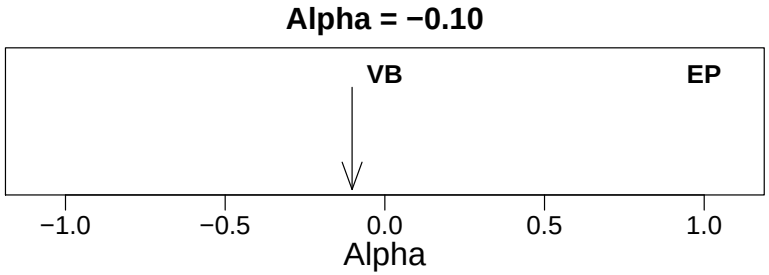

Example with 2D Gaussians
Exact Posterior
Approximation
Weight 2
Weight 1

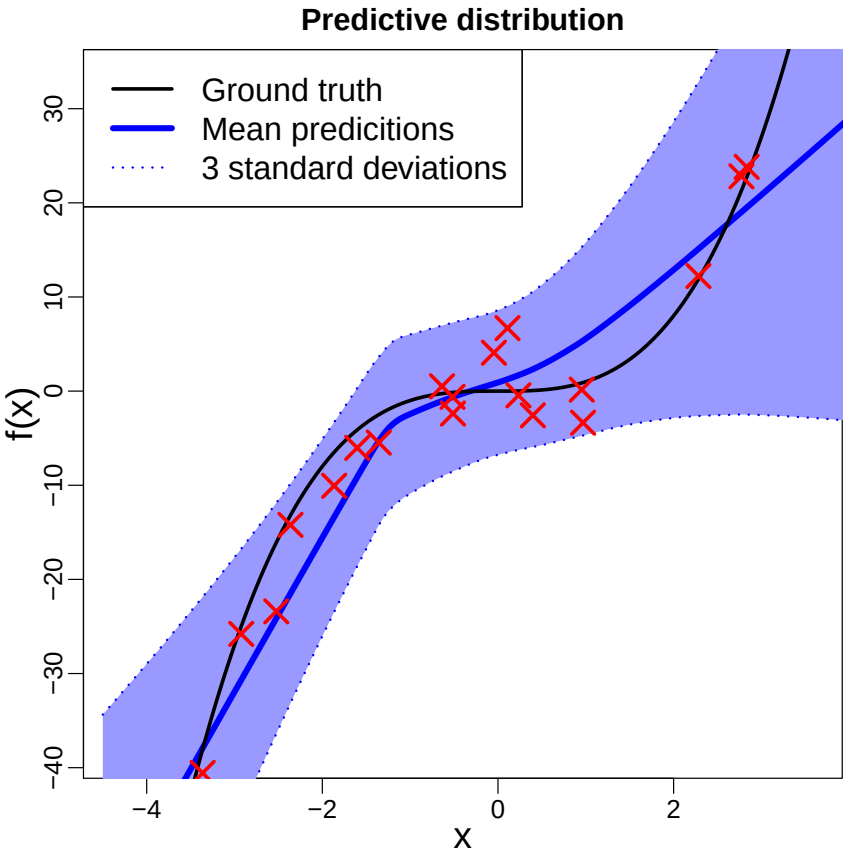
Example

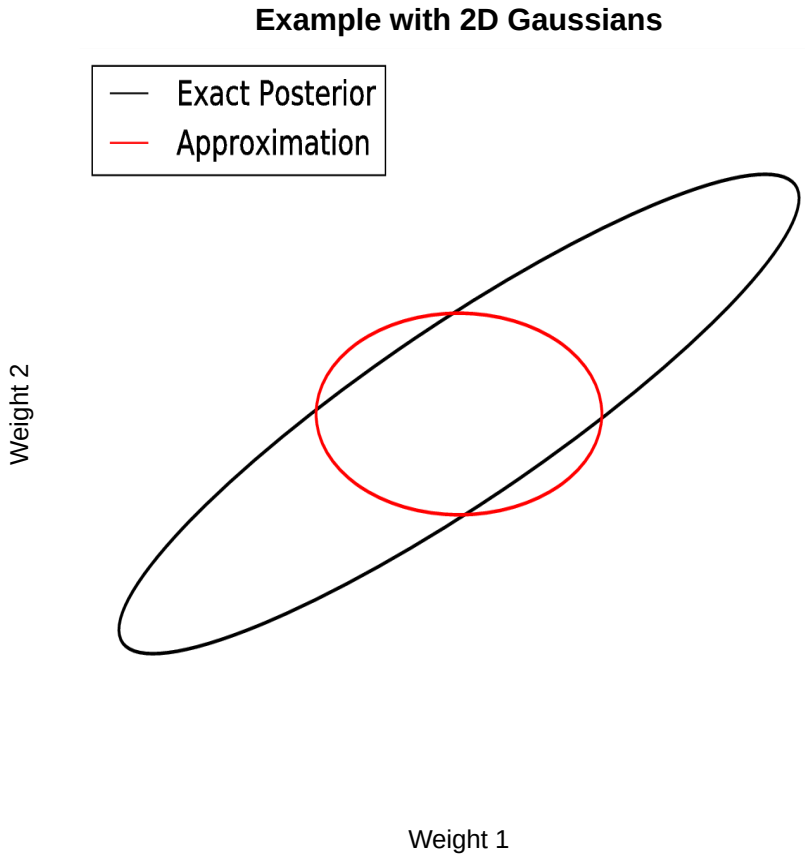

Alpha = −0.18
VB
EP
−1.0
−0.5
0.0
0.5
1.0
Alpha
Predictive distribution
Ground truth
Mean predicitions
3 standard deviations
f(x)
30
20
10
0
−10
−20
−30
−40
−4
−2
0
2
X


Example with 2D Gaussians
Exact Posterior
Approximation
Weight 2
Weight 1

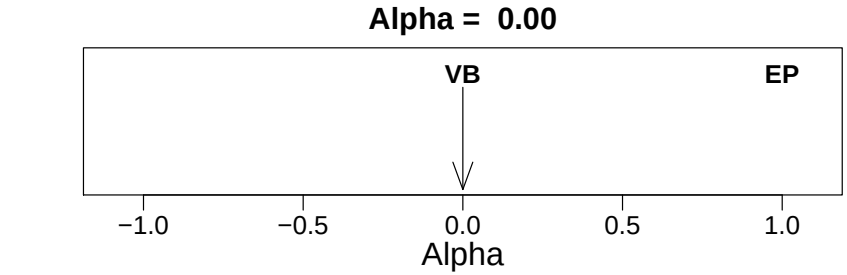
Example

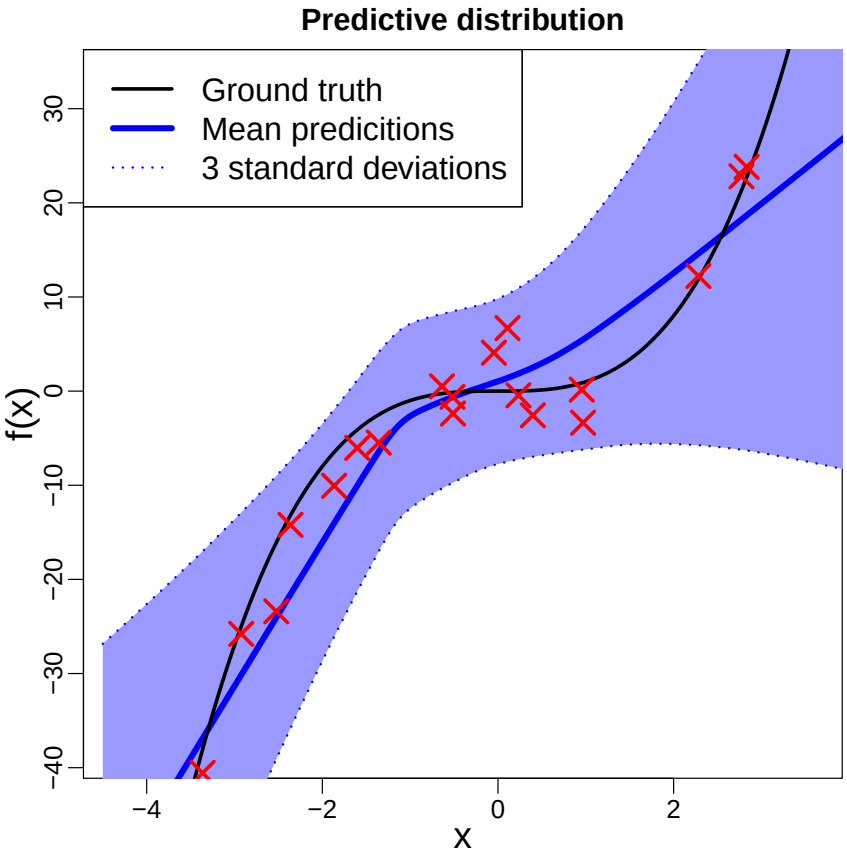

Alpha = −0.10
VB
EP
−1.0
−0.5
0.0
0.5
1.0
Alpha

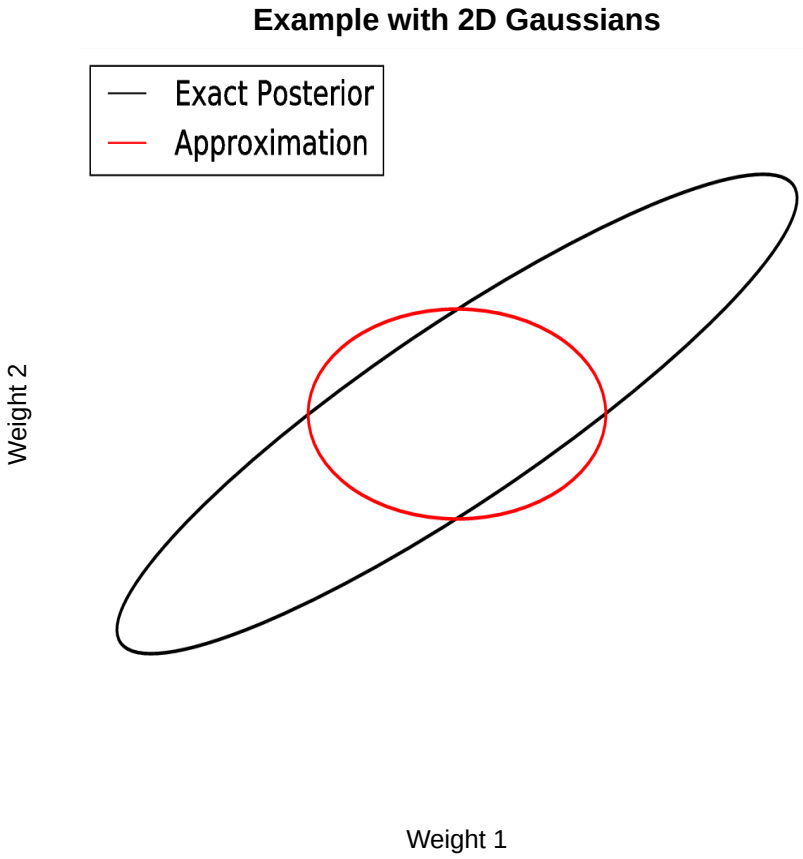

Predictive distribution
Ground truth
Mean predicitions
3 standard deviations
f(x)
30
20
10
0
−10
−20
−30
−40
−4
−2
0
2
X


Example with 2D Gaussians
Exact Posterior
Approximation
Weight 2
Weight 1

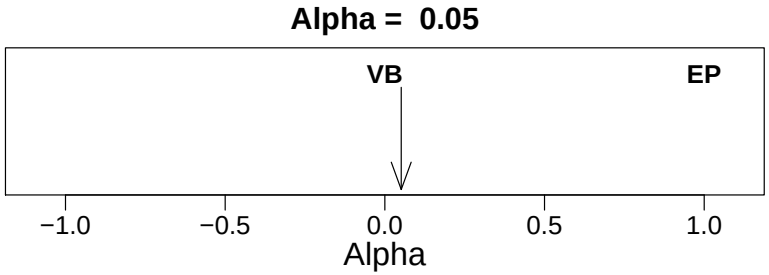
Example

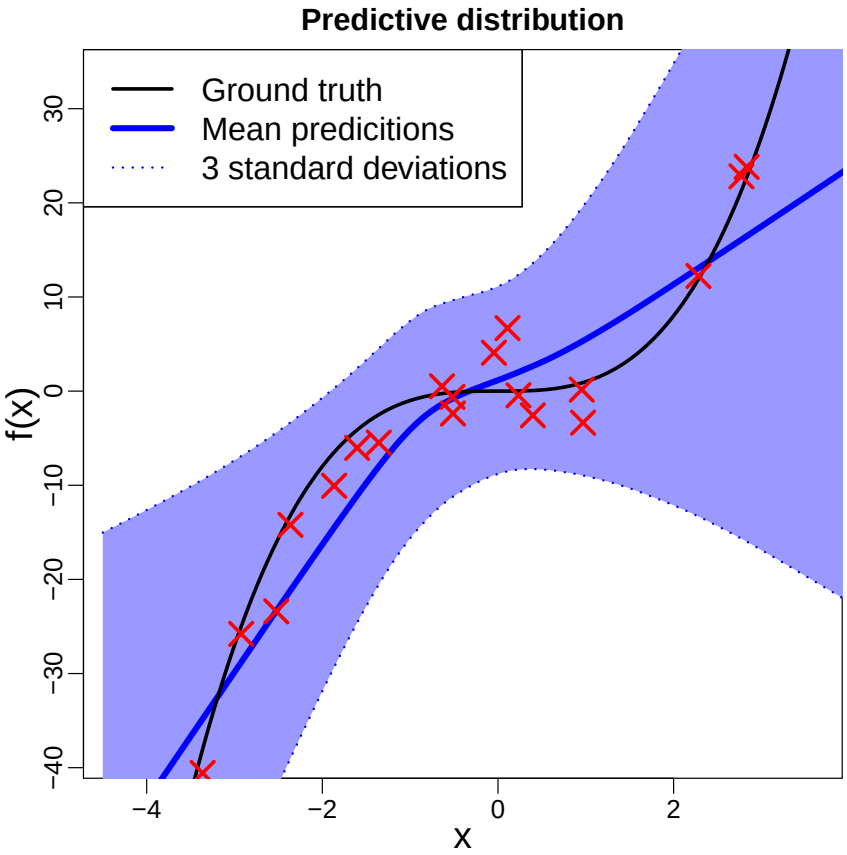

Alpha = 0.00
VB
EP
−1.0
−0.5
0.0
0.5
1.0
Alpha

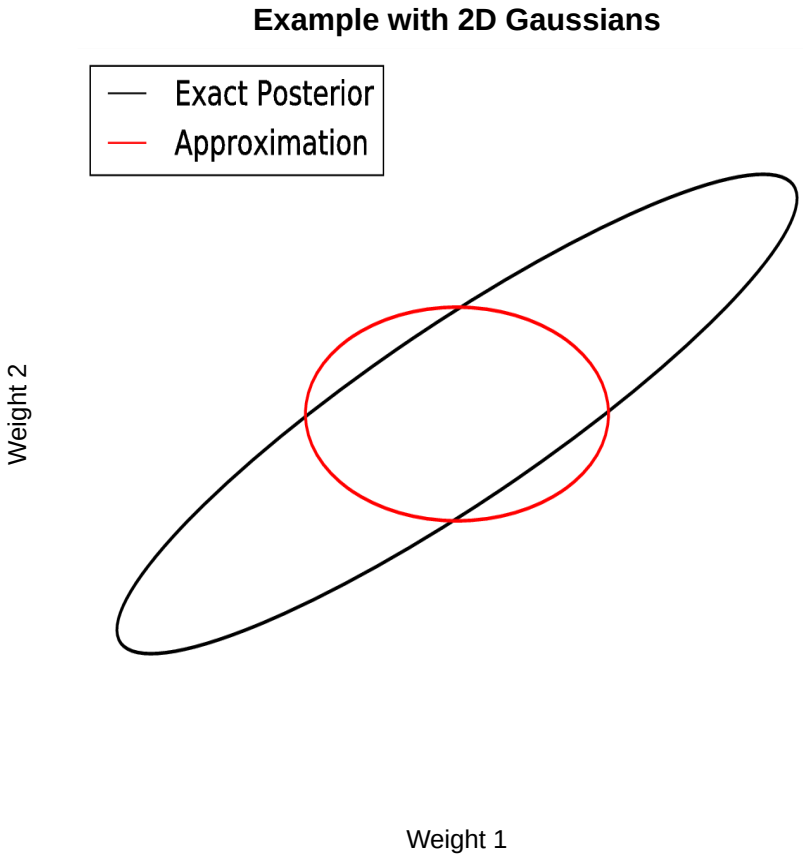

Predictive distribution
Ground truth
Mean predicitions
3 standard deviations
f(x)
30
20
10
0
−10
−20
−30
−40
−4
−2
0
2
X


Example with 2D Gaussians
Exact Posterior
Approximation
Weight 2
Weight 1

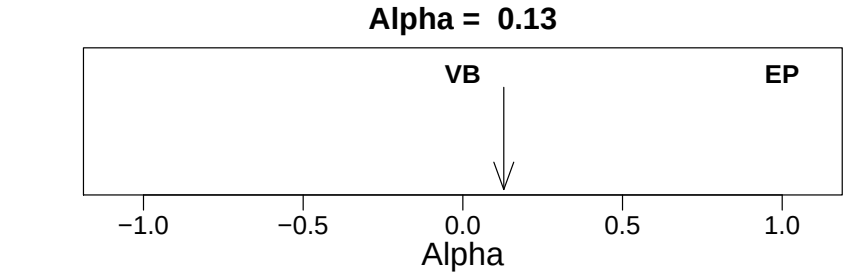
Example

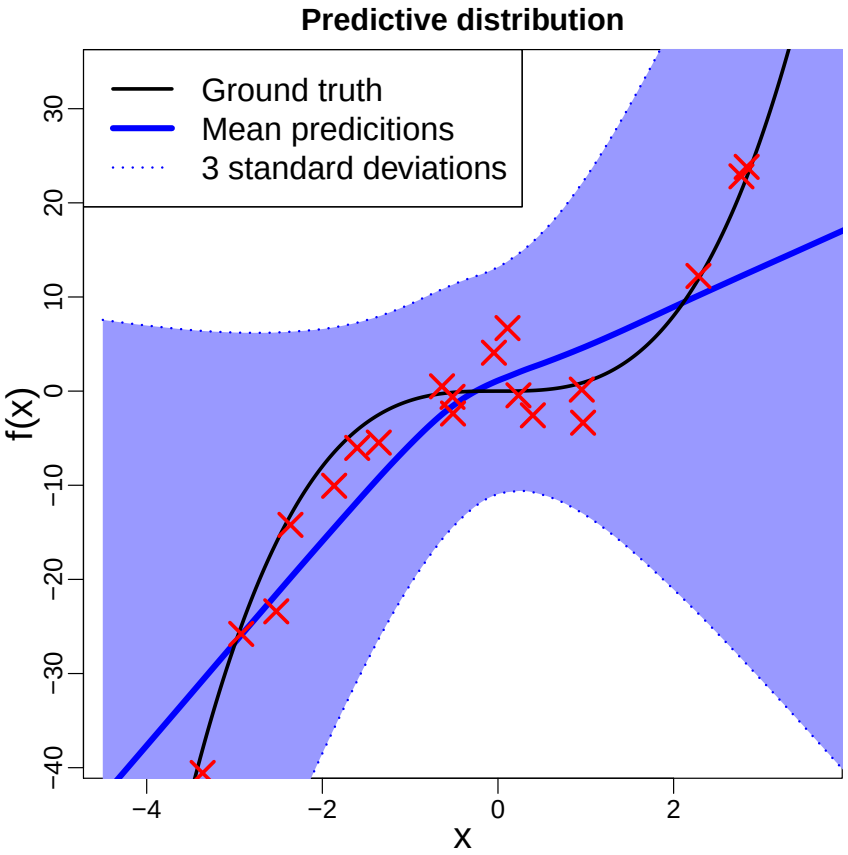

Alpha = 0.05
VB
EP
−1.0
−0.5
0.0
0.5
1.0
Alpha

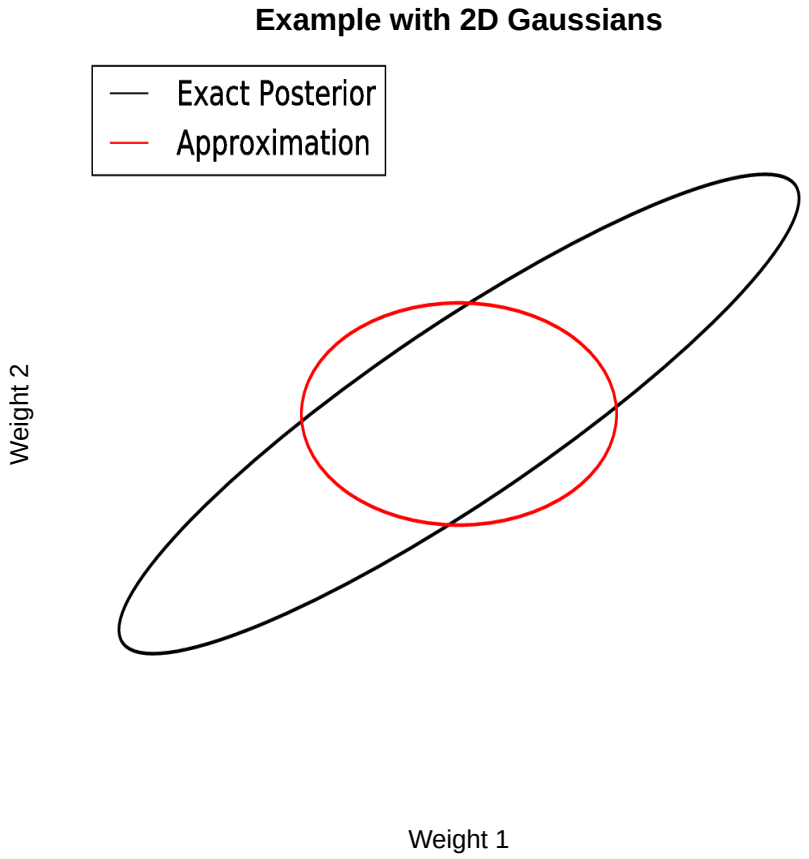

Predictive distribution
Ground truth
Mean predicitions
3 standard deviations
f(x)
30
20
10
0
−10
−20
−30
−40
−4
−2
0
2
X


Example with 2D Gaussians
Exact Posterior
Approximation
Weight 2
Weight 1

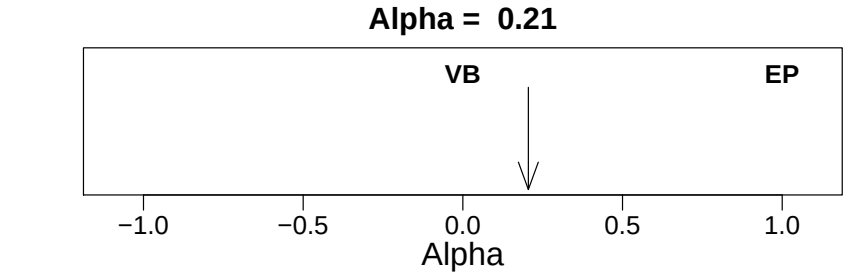
Example

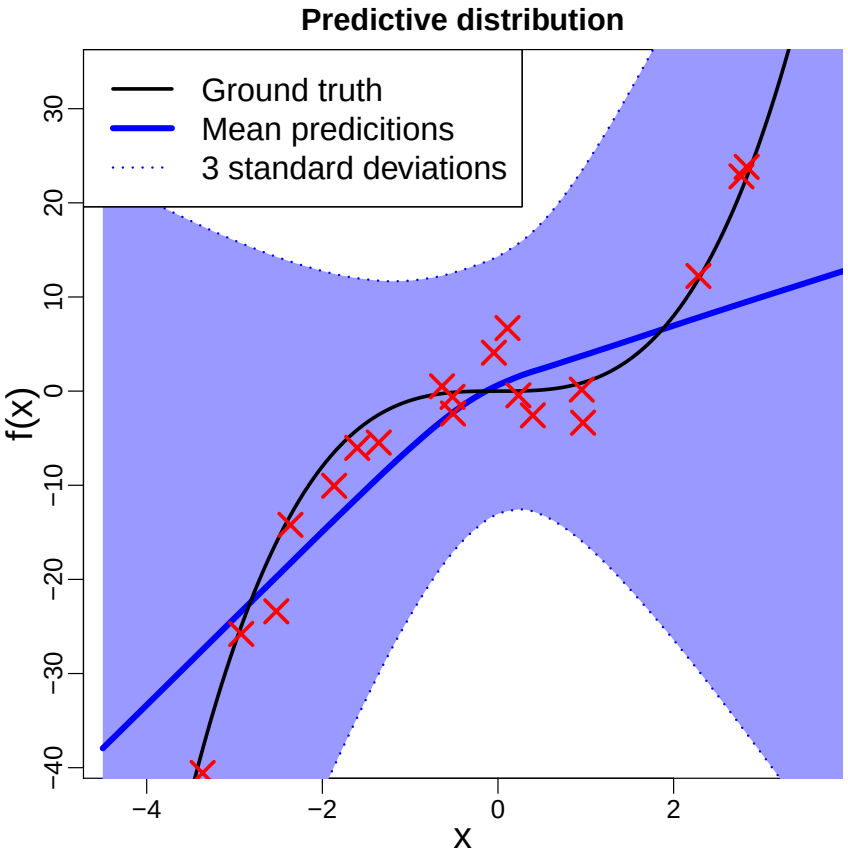

Alpha = 0.13
VB
EP
−1.0
−0.5
0.0
0.5
1.0
Alpha

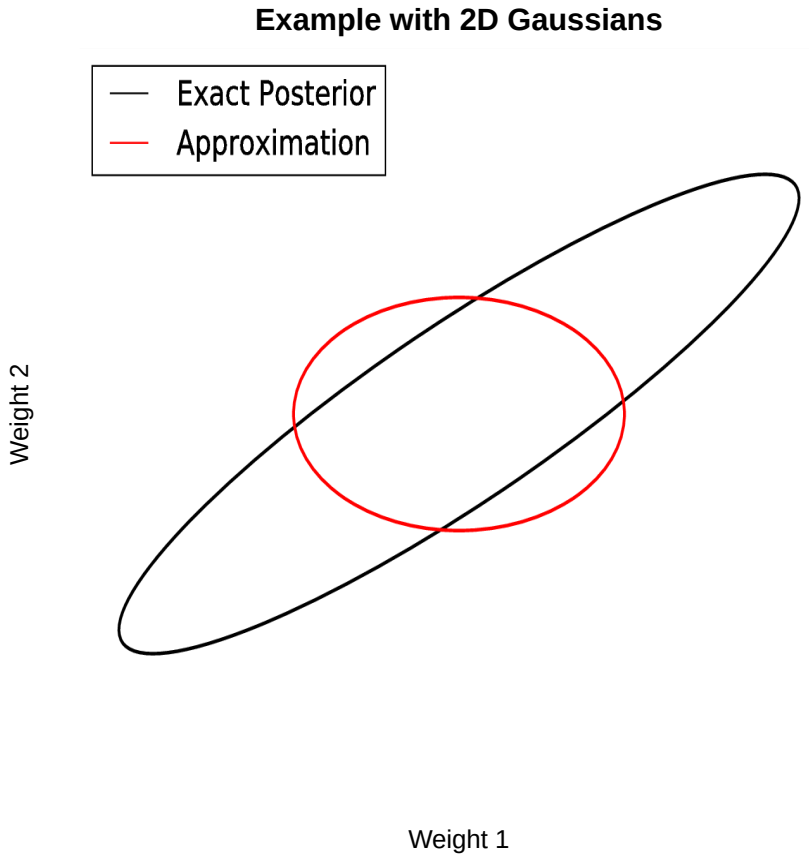

Predictive distribution
Ground truth
Mean predicitions
3 standard deviations
f(x)
30
20
10
0
−10
−20
−30
−40
−4
−2
0
2
X


Example with 2D Gaussians
Exact Posterior
Approximation
Weight 2
Weight 1

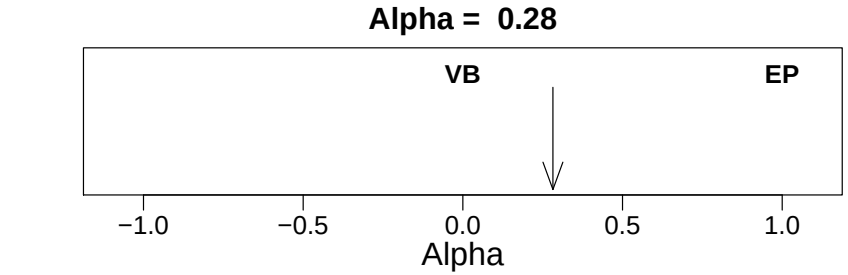
Example

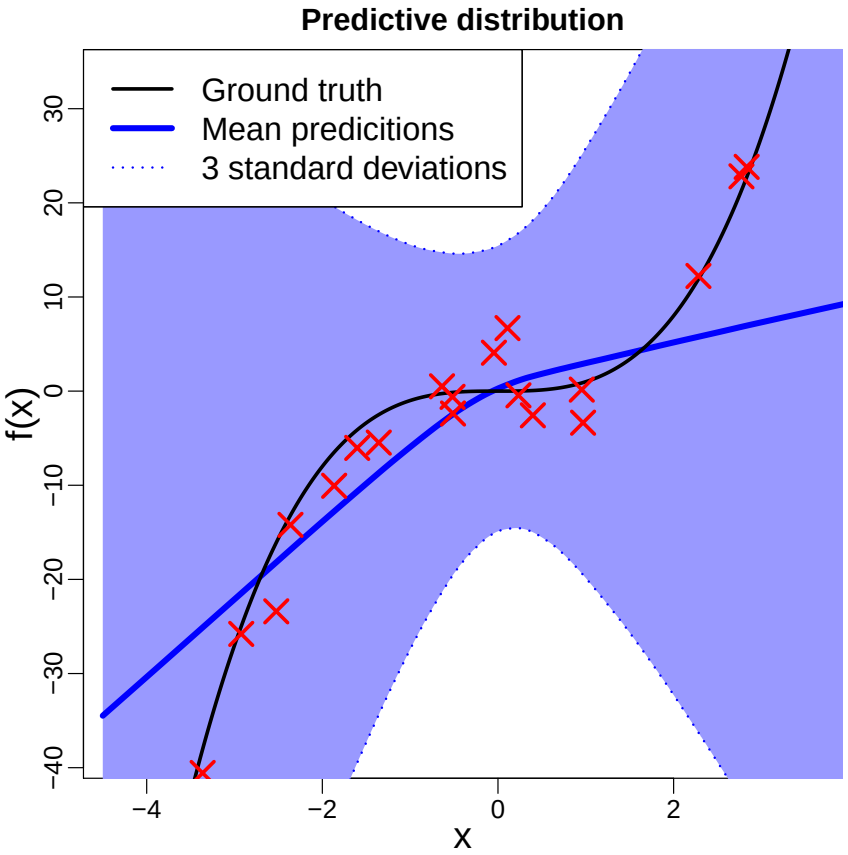

Alpha = 0.21
VB
EP
−1.0
−0.5
0.0
0.5
1.0
Alpha

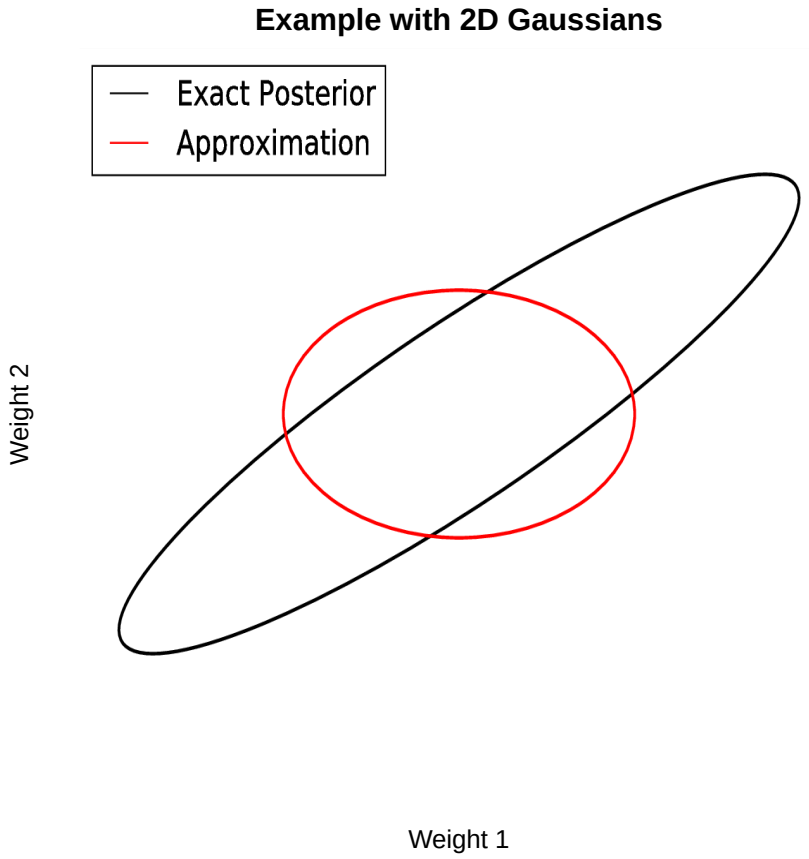

Predictive distribution
Ground truth
Mean predicitions
3 standard deviations
f(x)
30
20
10
0
−10
−20
−30
−40
−4
−2
0
2
X


Example with 2D Gaussians
Exact Posterior
Approximation
Weight 2
Weight 1

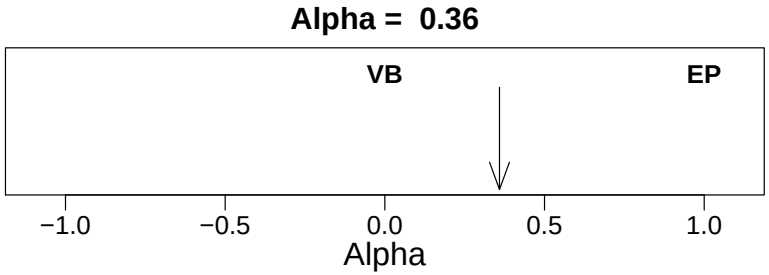
Example

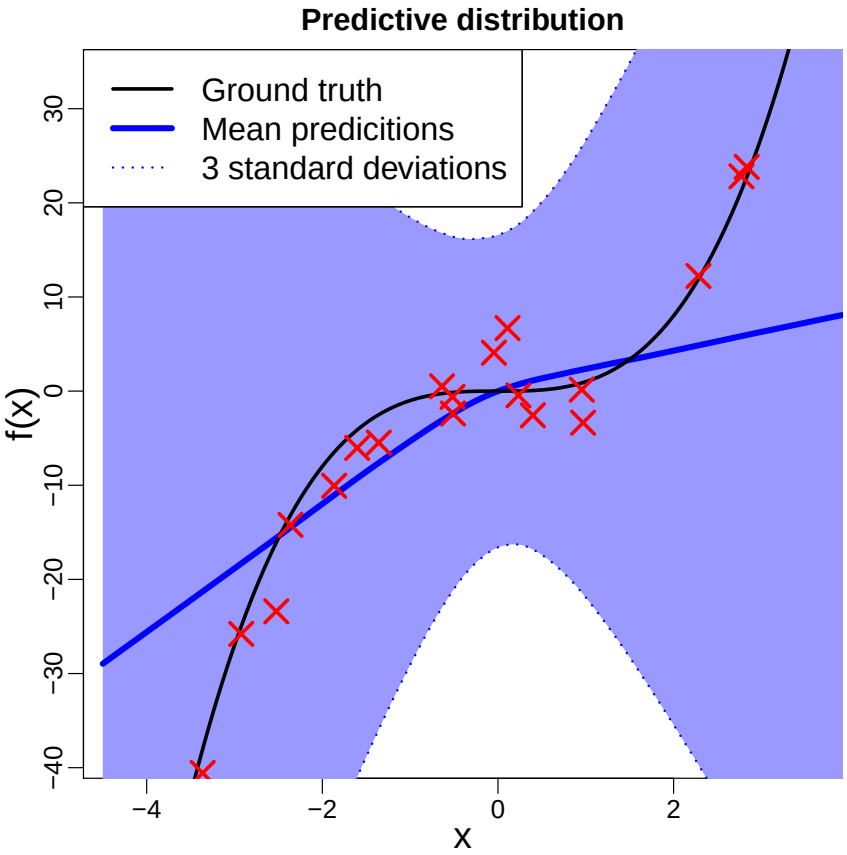

Alpha = 0.28
VB
EP
−1.0
−0.5
0.0
0.5
1.0
Alpha

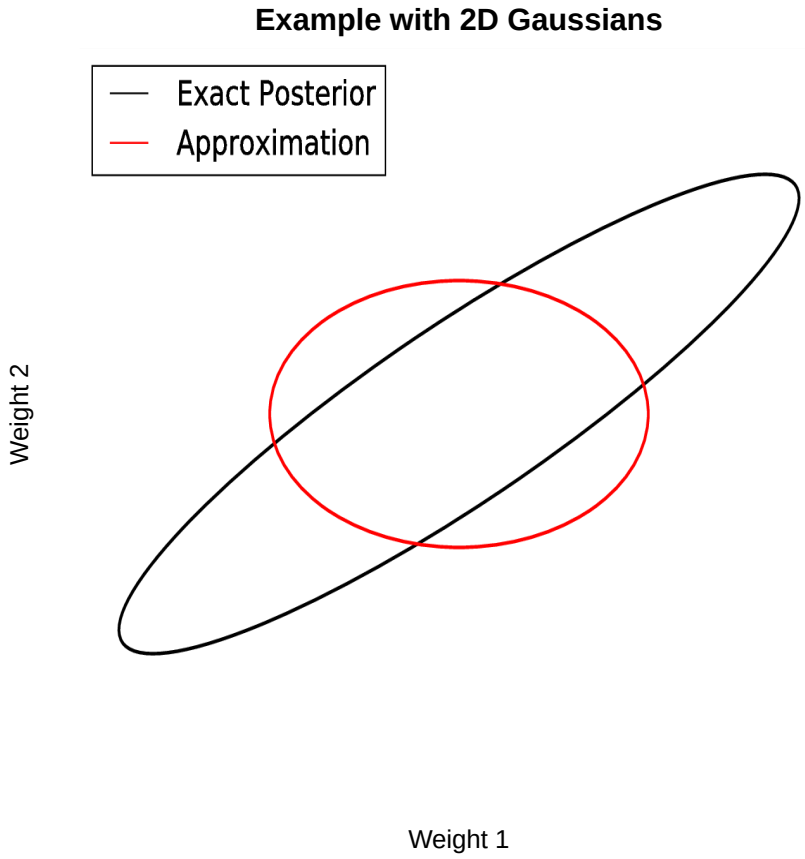

Predictive distribution
Ground truth
Mean predicitions
3 standard deviations
f(x)
30
20
10
0
−10
−20
−30
−40
−4
−2
0
2
x


Example with 2D Gaussians
Exact Posterior
Approximation
Weight 2
Weight 1

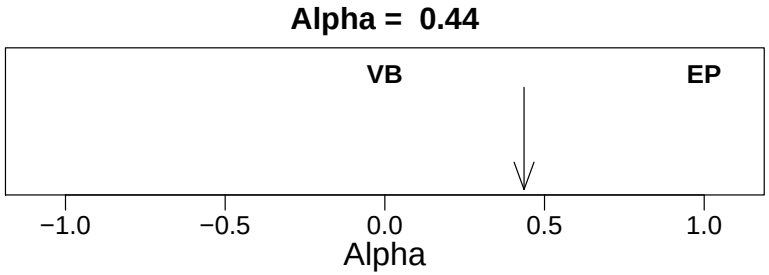
Example

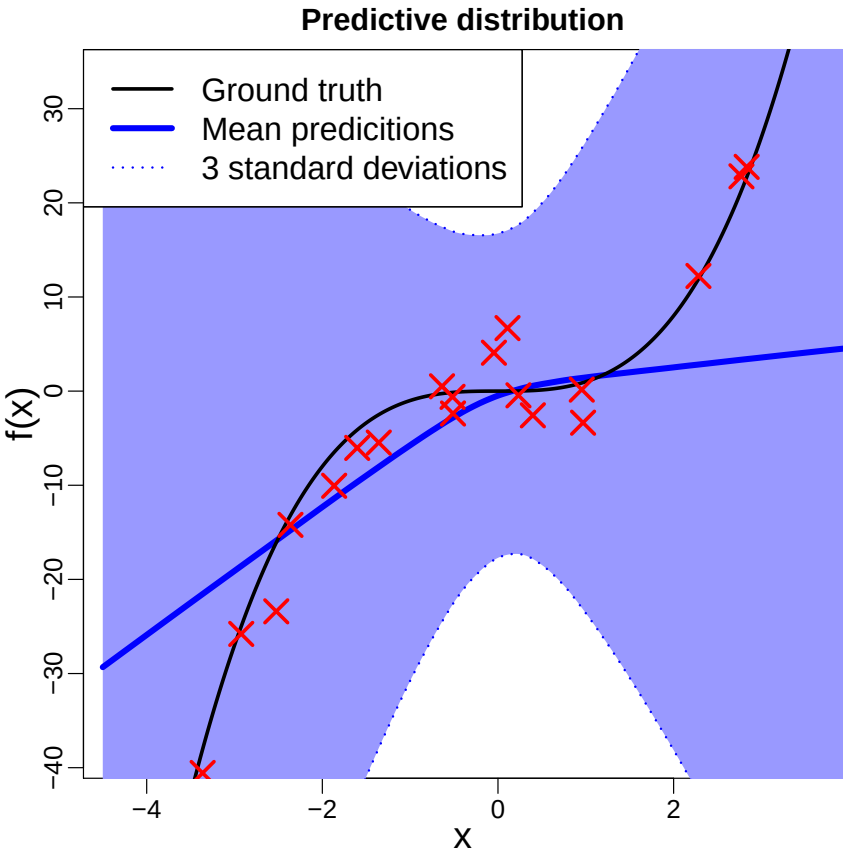

Alpha = 0.36
VB
EP
−1.0
−0.5
0.0
0.5
1.0
Alpha

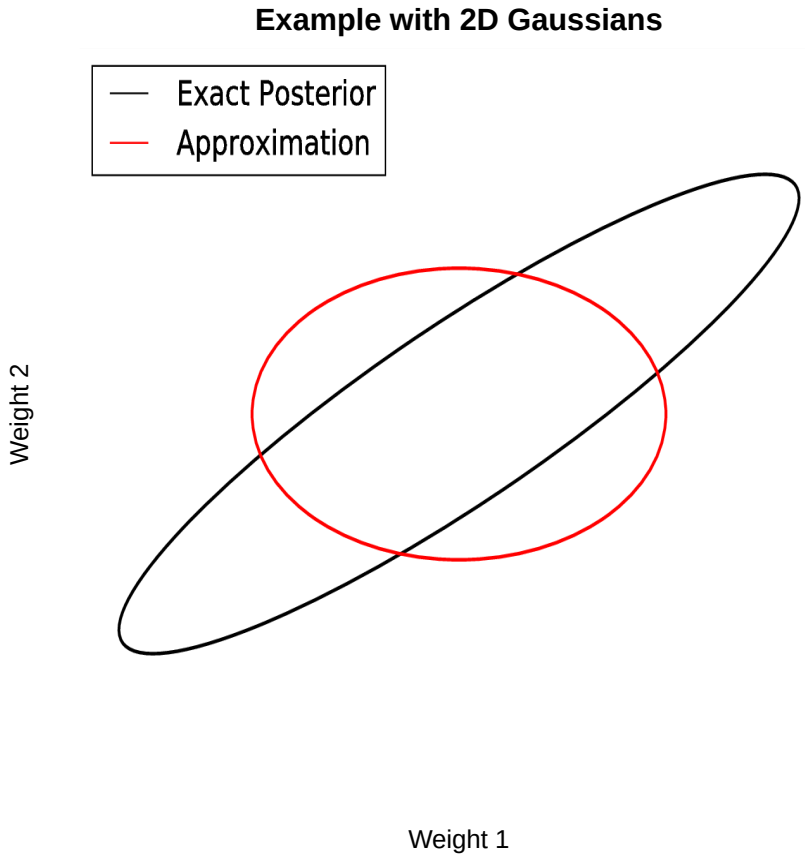

Predictive distribution
Ground truth
Mean predicitions
3 standard deviations
f(x)
30
20
10
0
−10
−20
−30
−40
−4
−2
0
2
x


Example with 2D Gaussians
Exact Posterior
Approximation
Weight 2
Weight 1

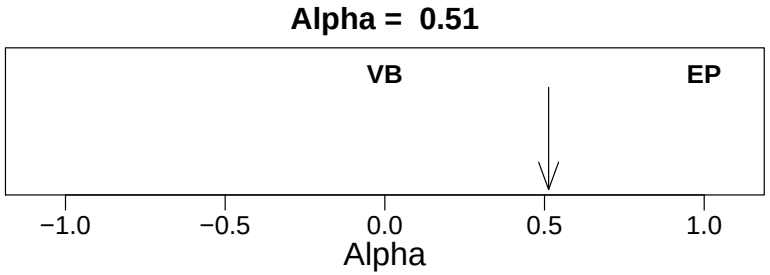
Example

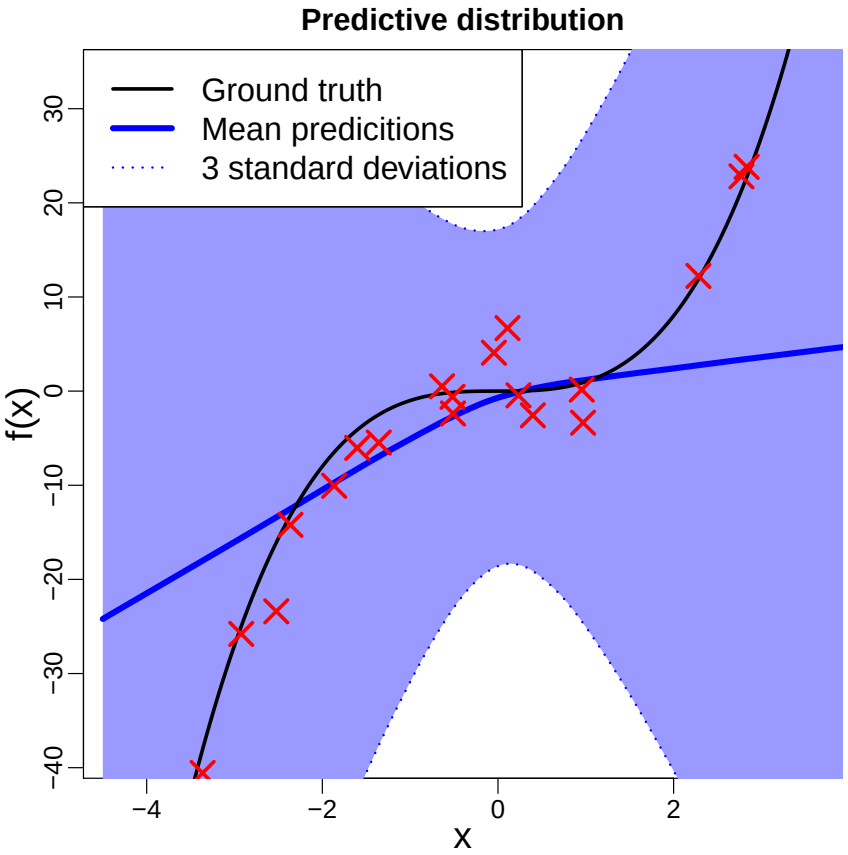

Alpha = 0.44
VB
EP
−1.0
−0.5
0.0
0.5
1.0
Alpha

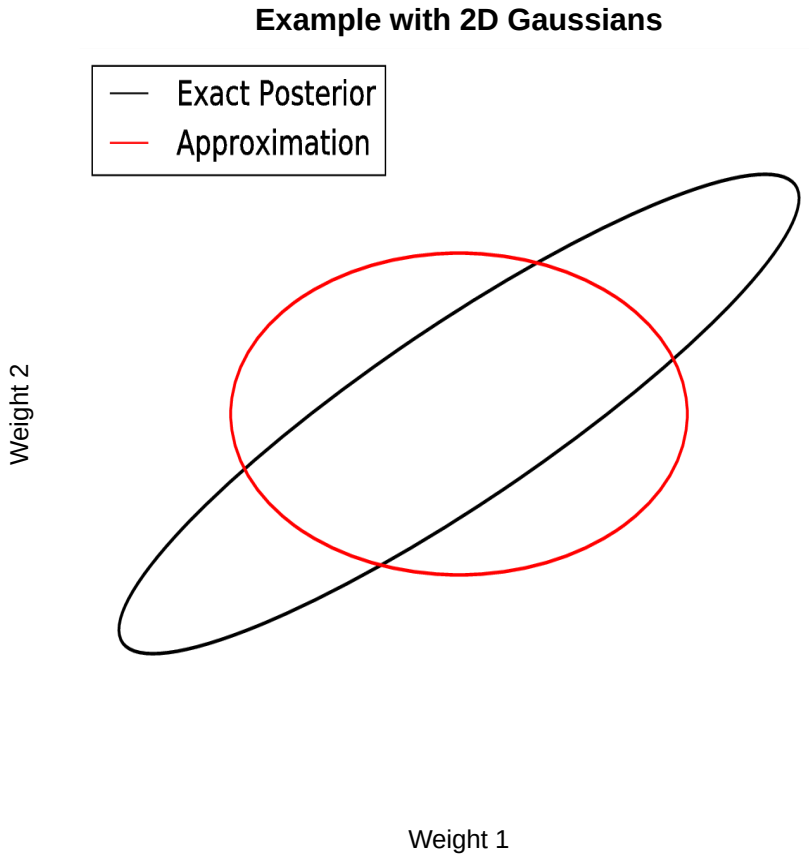

Predictive distribution
Ground truth
Mean predicitions
3 standard deviations
f(x)
30
20
10
0
−10
−20
−30
−40
−4
−2
0
2
x


Example with 2D Gaussians
Exact Posterior
Approximation
Weight 2
Weight 1

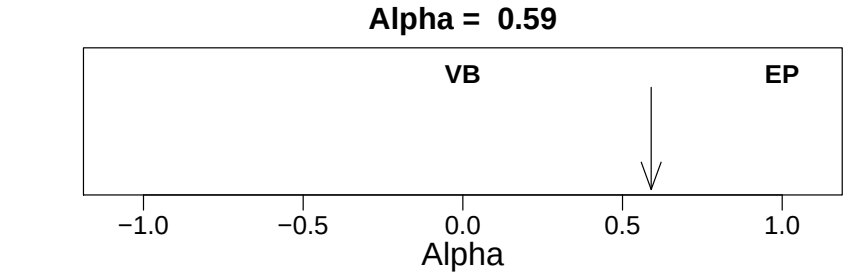
Example

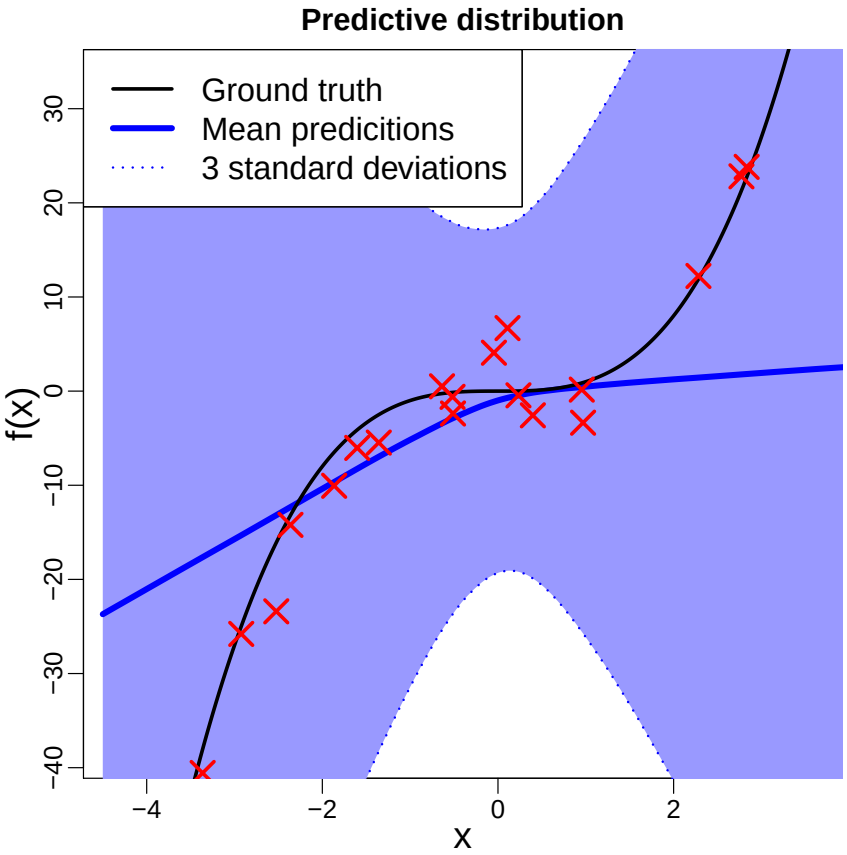

Alpha = 0.51
VB
EP
−1.0
−0.5
0.0
0.5
1.0
Alpha

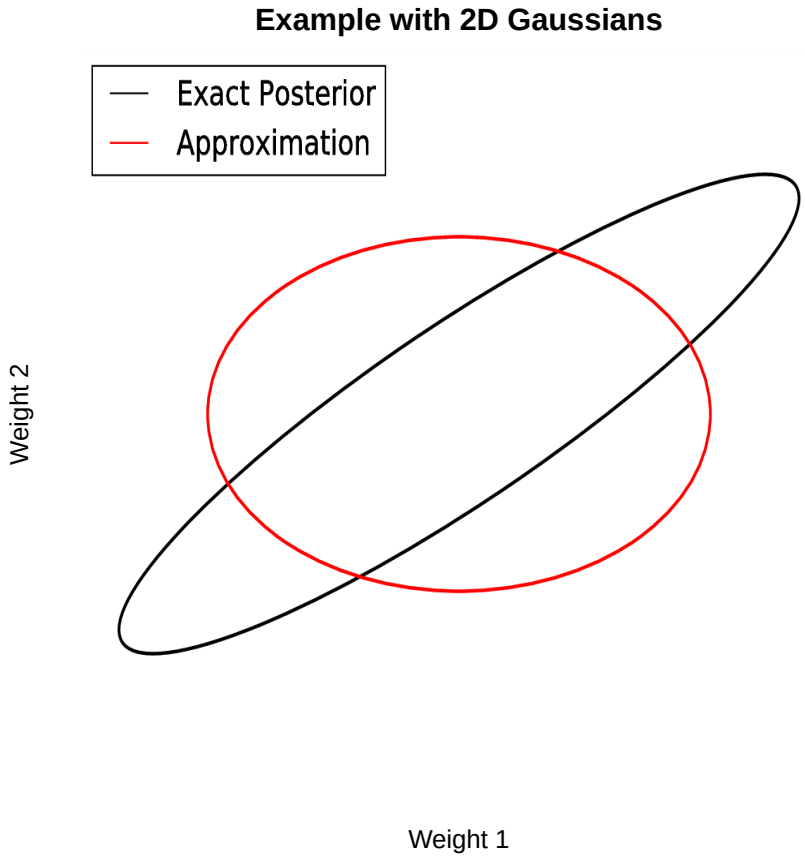

Predictive distribution
Ground truth
Mean predicitions
3 standard deviations
f(x)
30
20
10
0
−10
−20
−30
−40
−4
−2
0
2
X


Example with 2D Gaussians
Exact Posterior
Approximation
Weight 2
Weight 1

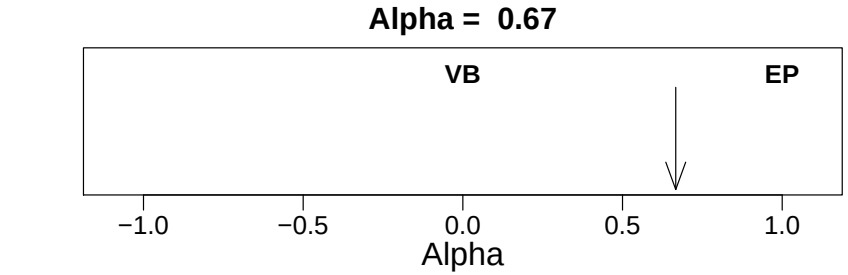
Example

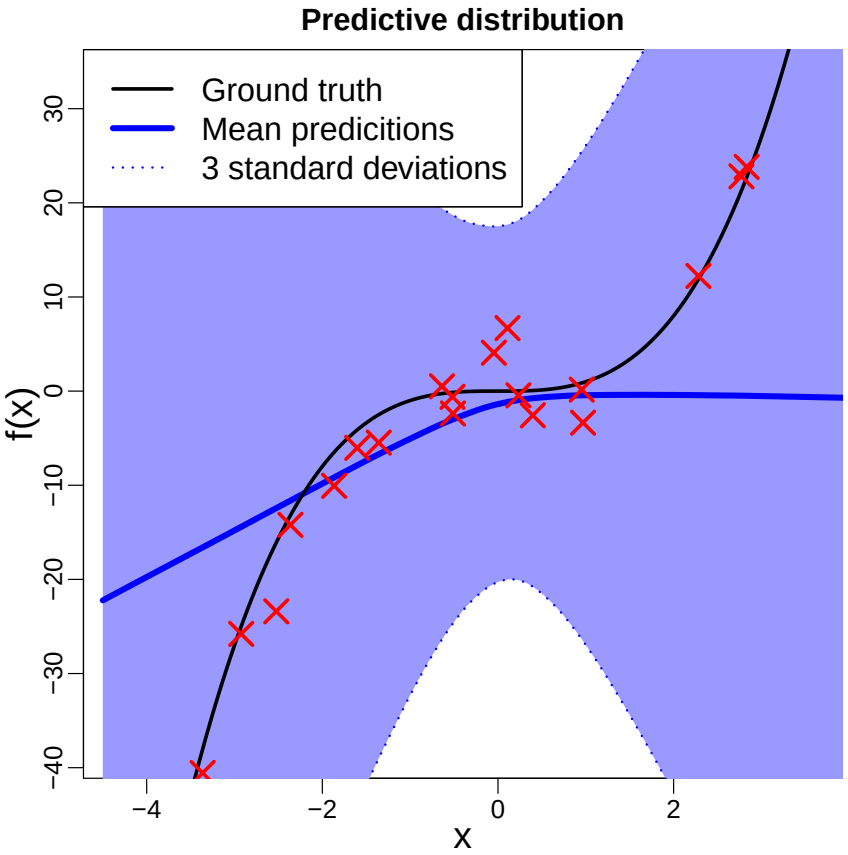

Alpha = 0.59
VB
EP
−1.0
−0.5
0.0
0.5
1.0
Alpha

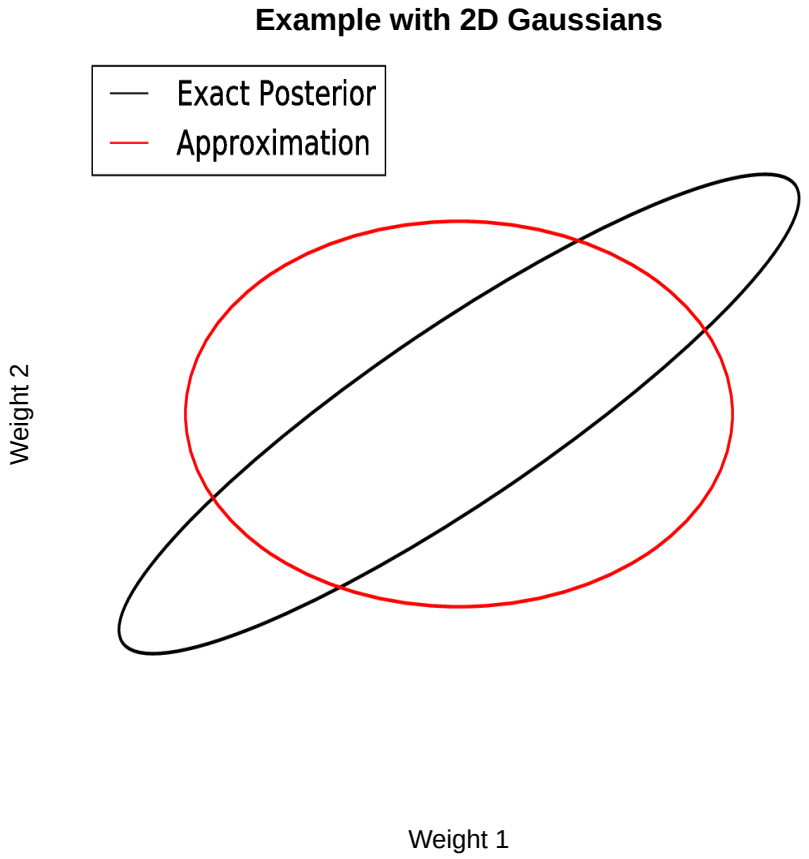

Predictive distribution
Ground truth
Mean predicitions
3 standard deviations
f(x)
30
20
10
0
−10
−20
−30
−40
−4
−2
0
2
X


Example with 2D Gaussians
Exact Posterior
Approximation
Weight 2
Weight 1

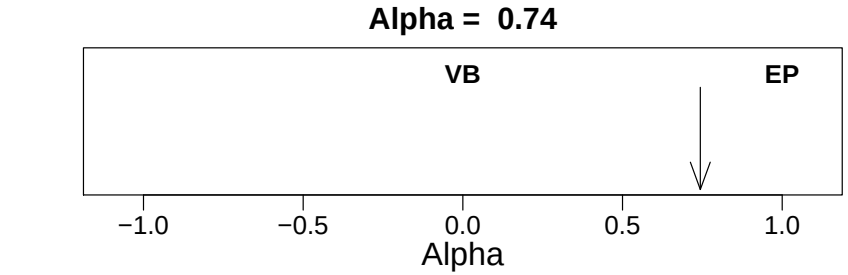
Example

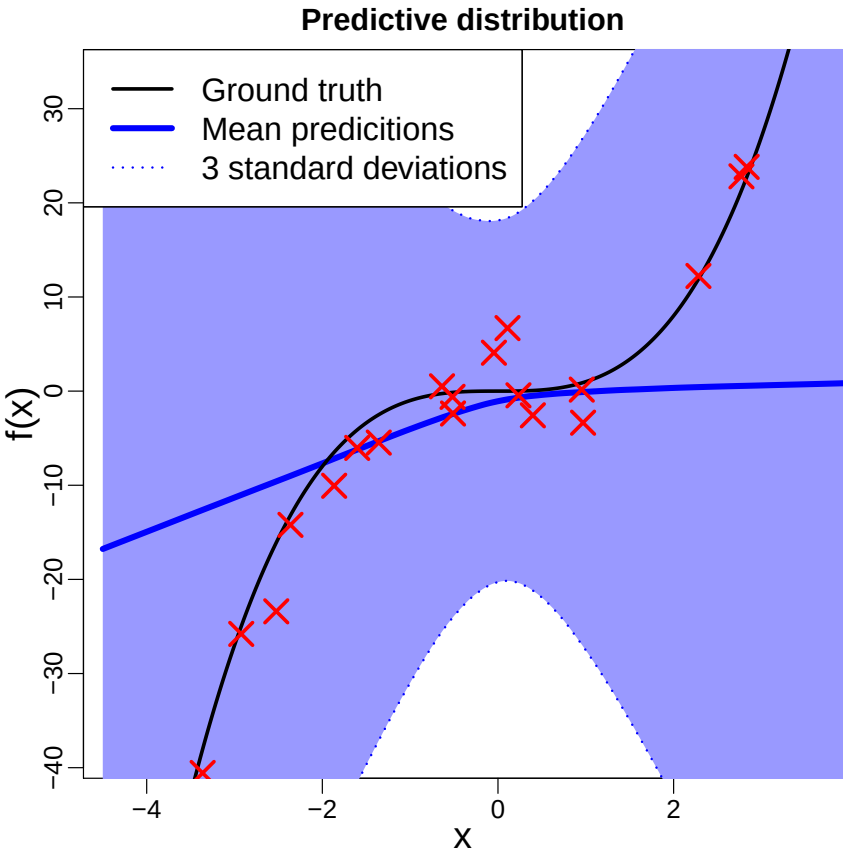

Alpha = 0.67
VB
EP
−1.0
−0.5
0.0
0.5
1.0
Alpha

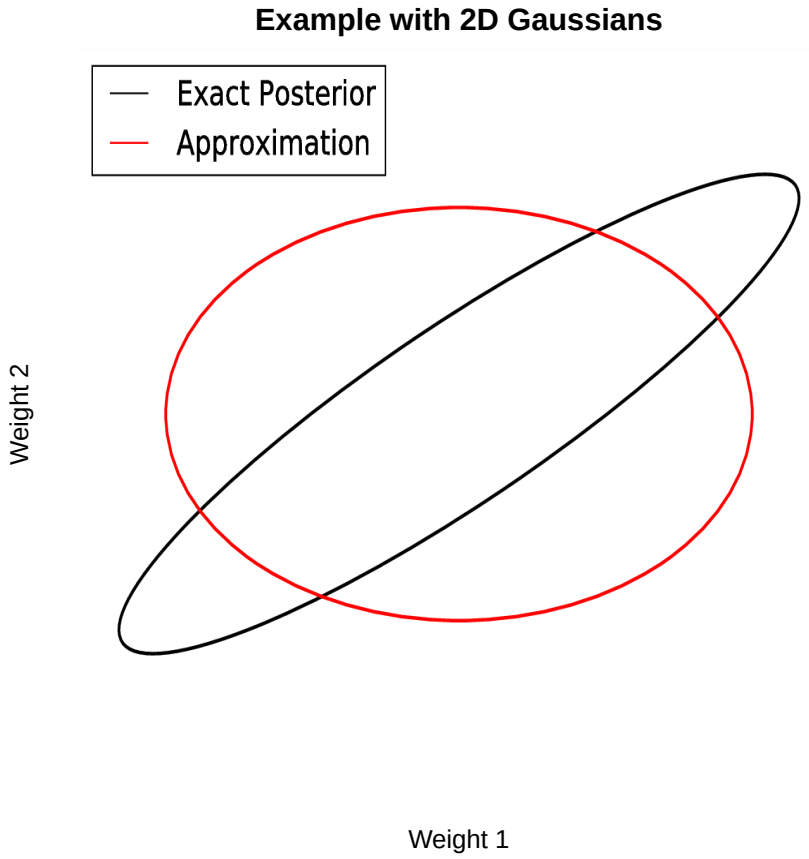

Predictive distribution
Ground truth
Mean predicitions
3 standard deviations
f(x)
30
20
10
0
−10
−20
−30
−40
−4
−2
0
2
x


Example with 2D Gaussians
Exact Posterior
Approximation
Weight 2
Weight 1

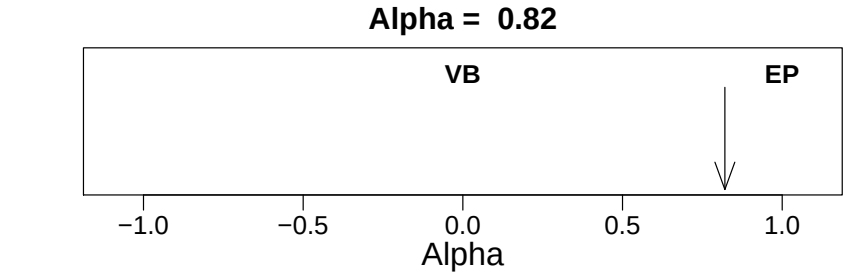
Example

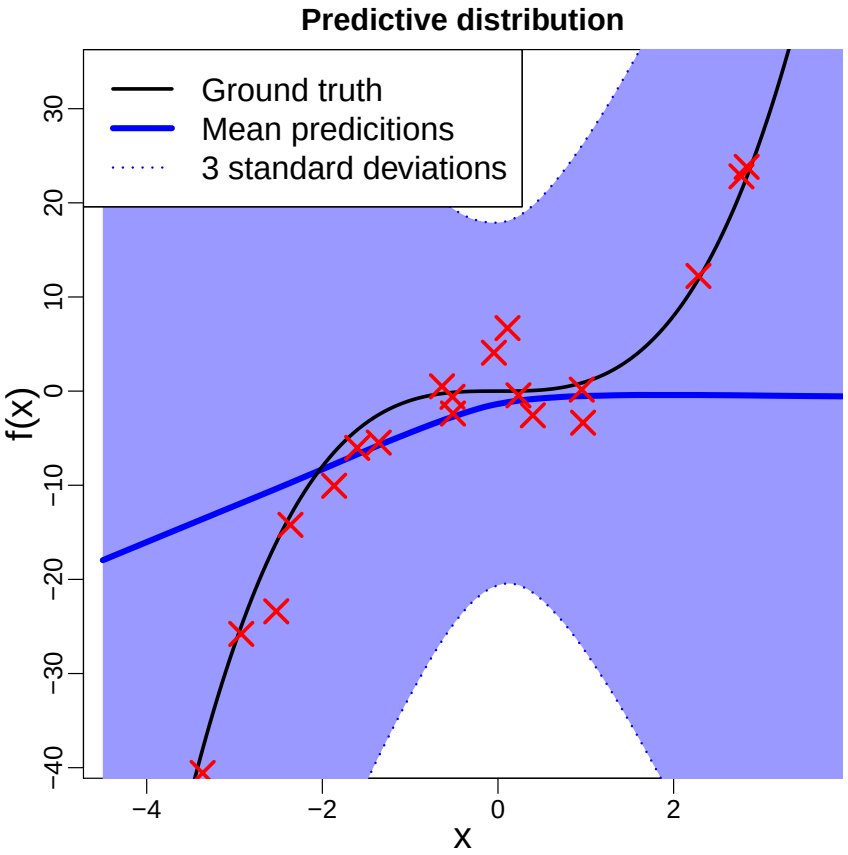

Alpha = 0.74
VB
EP
−1.0
−0.5
0.0
0.5
1.0
Alpha

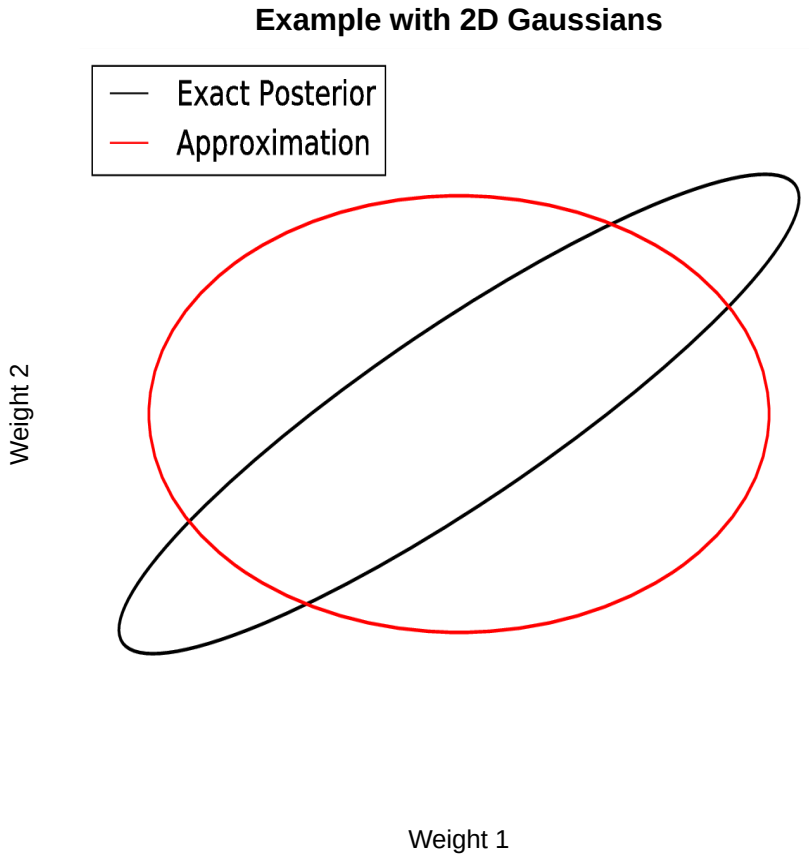

Predictive distribution
Ground truth
Mean predicitions
3 standard deviations
f(x)
30
20
10
0
−10
−20
−30
−40
−4
−2
0
2
X


Example with 2D Gaussians
Exact Posterior
Approximation
Weight 2
Weight 1

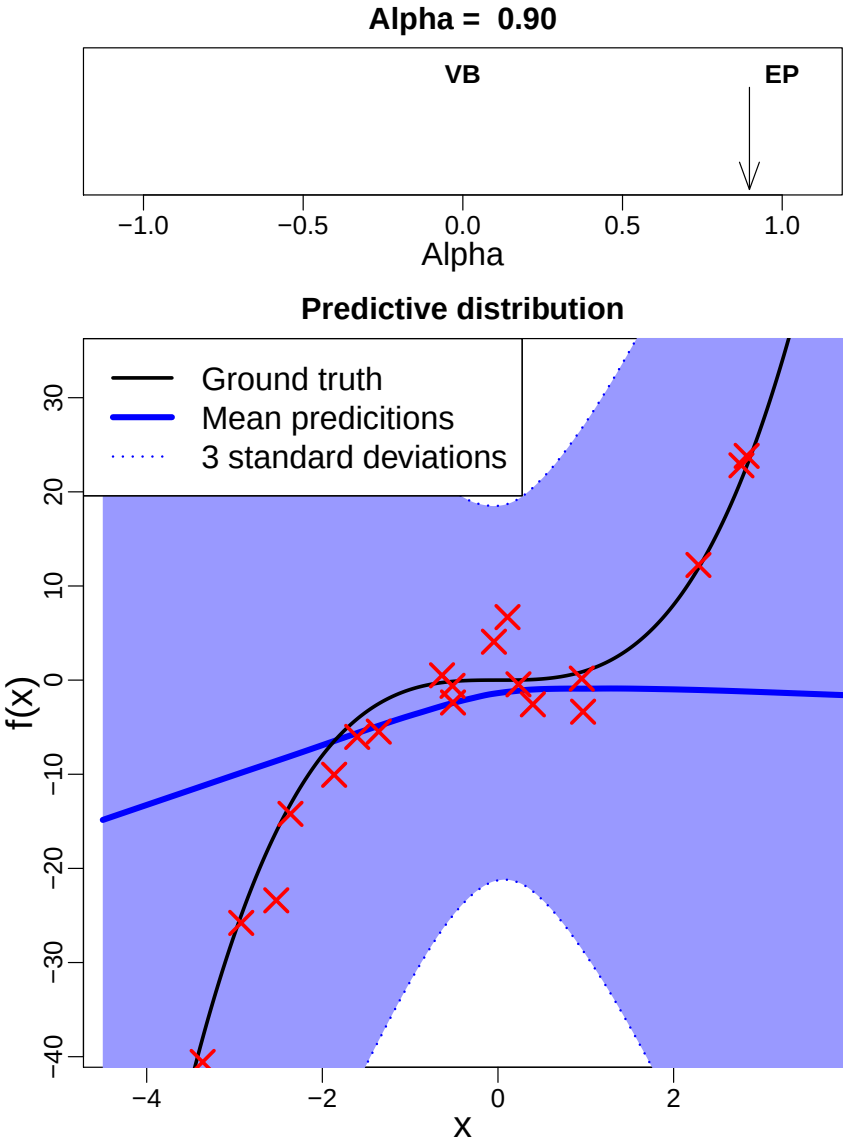
Example

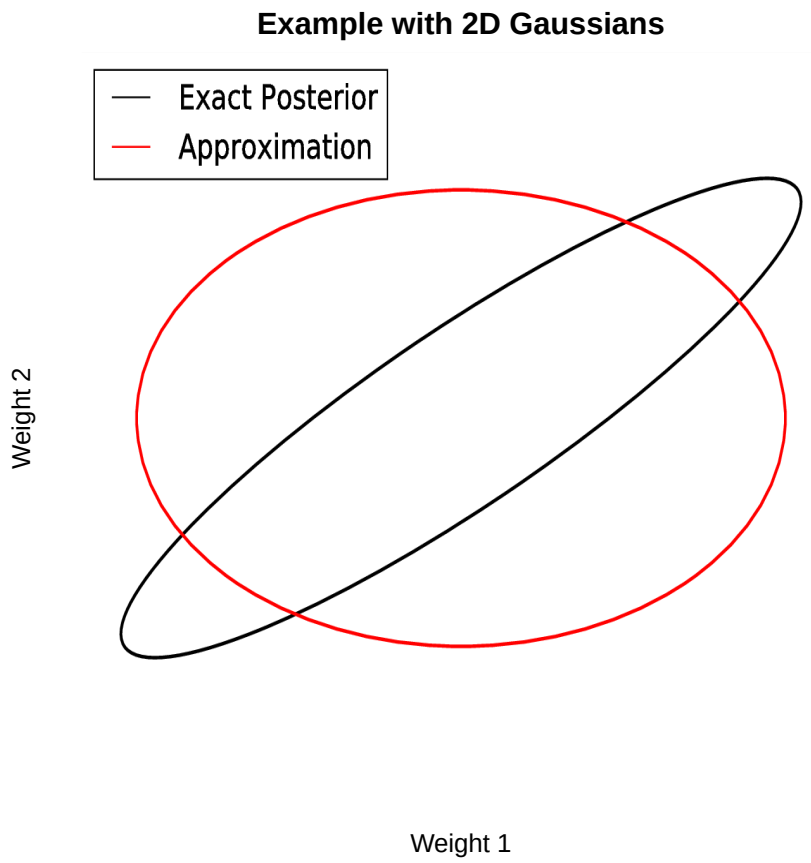

Alpha = 0.82
VB
EP
−1.0
−0.5
0.0
0.5
1.0
Alpha


Predictive distribution
Ground truth
Mean predicitions
3 standard deviations
f(x)
30
20
10
0
−10
−20
−30
−40
−4
−2
0
2
X

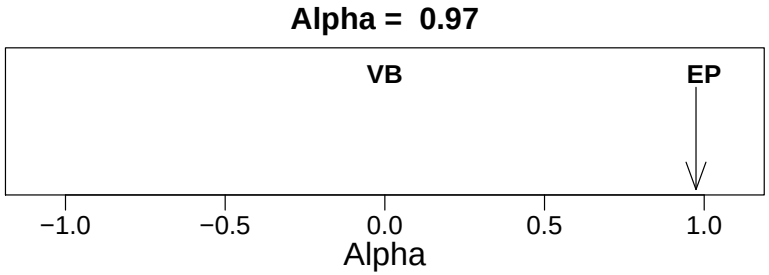

Example with 2D Gaussians
Exact Posterior
Approximation
Weight 2
Weight 1

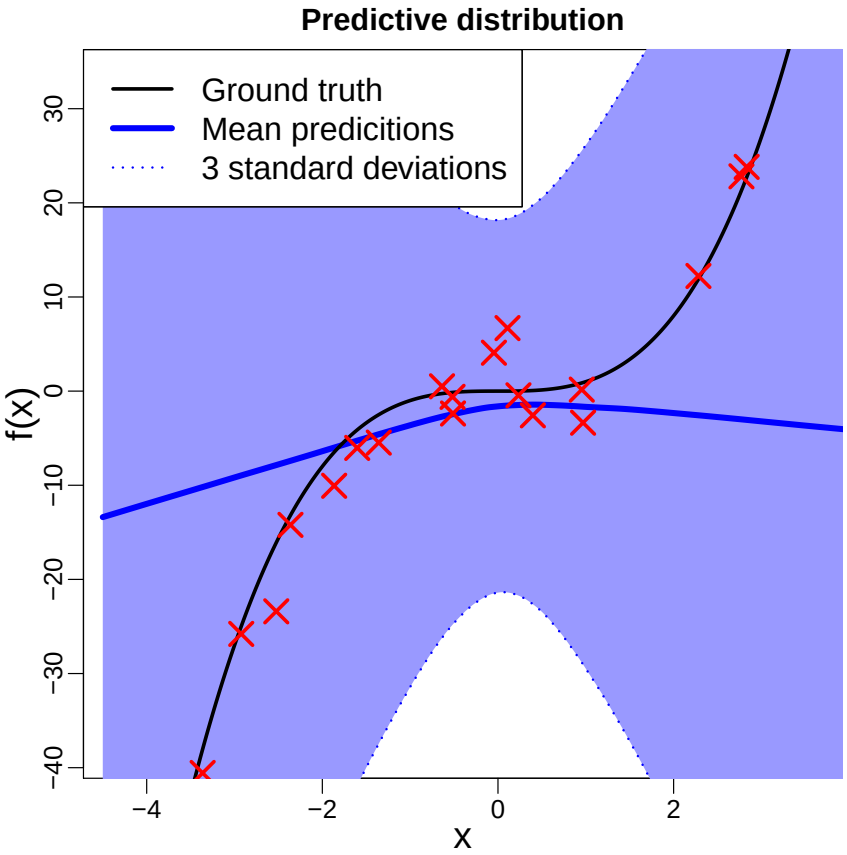
Example

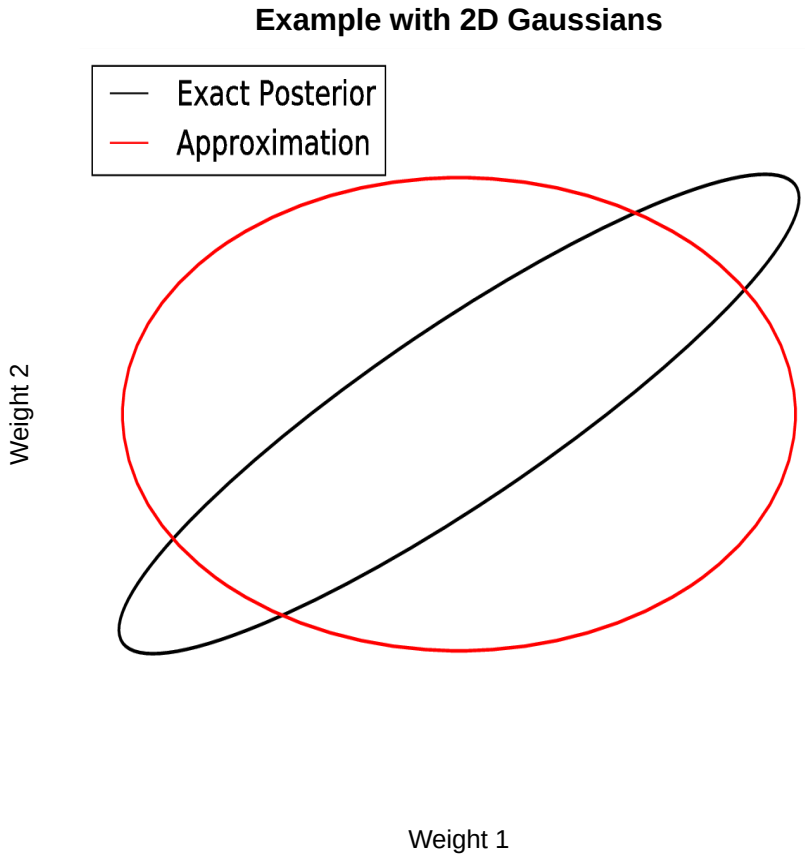

Alpha = 0.90
VB
EP
−1.0
−0.5
0.0
0.5
1.0
Alpha
Predictive distribution
Ground truth
Mean predicitions
3 standard deviations
f(x)
30
20
10
0
−10
−20
−30
−40
−4
−2
0
2
x


Example with 2D Gaussians
Exact Posterior
Approximation
Weight 2
Weight 1

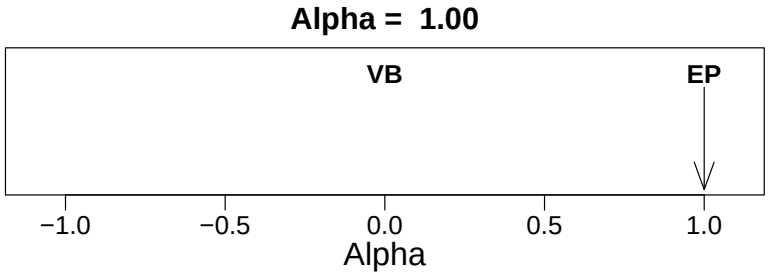
Example

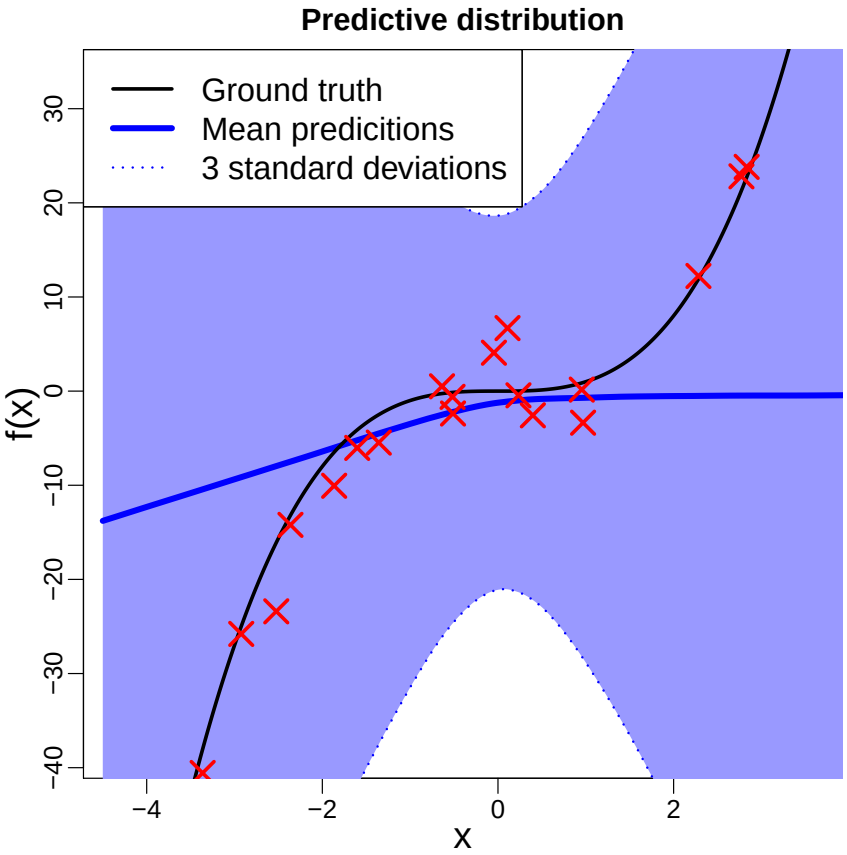

Alpha = 0.97
VB
EP
−1.0
−0.5
0.0
0.5
1.0
Alpha

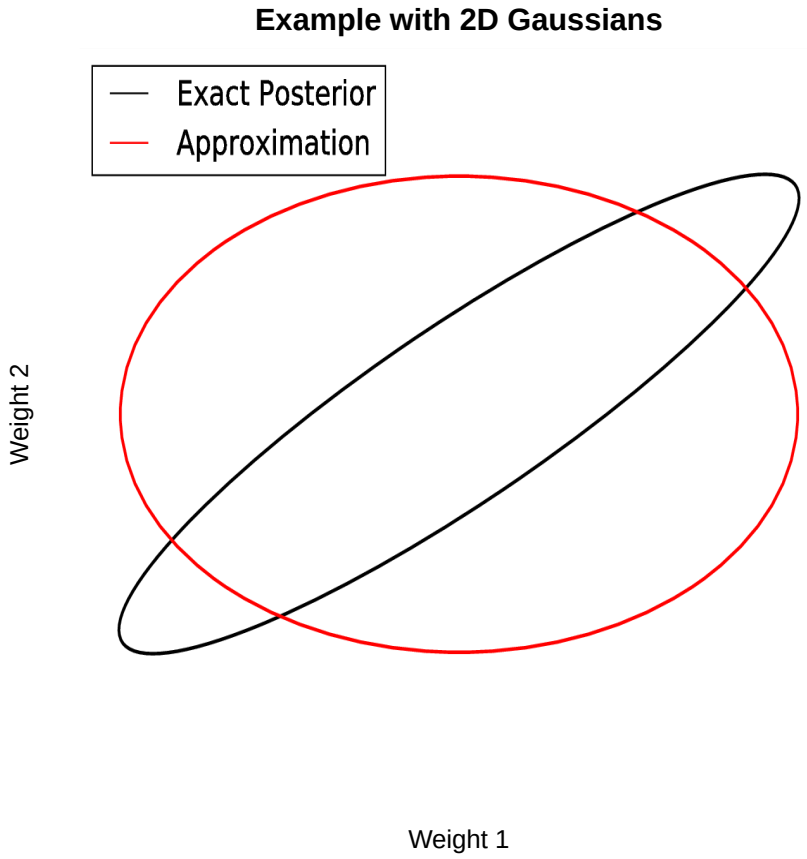

Predictive distribution
Ground truth
Mean predicitions
3 standard deviations
f(x)
30
20
10
0
−10
−20
−30
−40
−4
−2
0
2
x


Example with 2D Gaussians
Exact Posterior
Approximation
Weight 2
Weight 1

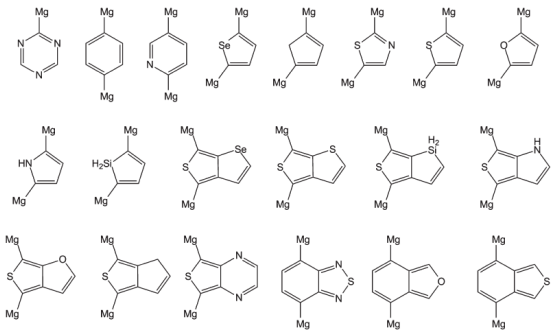
Example

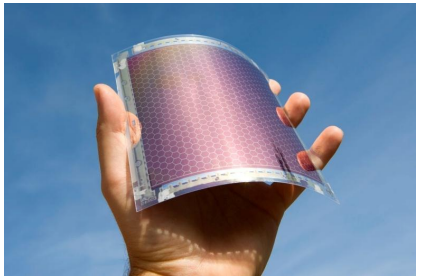

Alpha = 1.00
VB
EP
−1.0
−0.5
0.0
0.5
1.0
Alpha


Predictive distribution
Ground truth
Mean predicitions
3 standard deviations
f(x)
30
20
10
0
−10
−20
−30
−40
−4
−2
0
2
x


Example with 2D Gaussians
Exact Posterior
Approximation
Weight 2
Weight 1

Harvard clean energy project. Data from 60.000 **organic photovoltaics**.





Bayesian deep neural networks (BDNN), 2 hidden layers with 400 units.

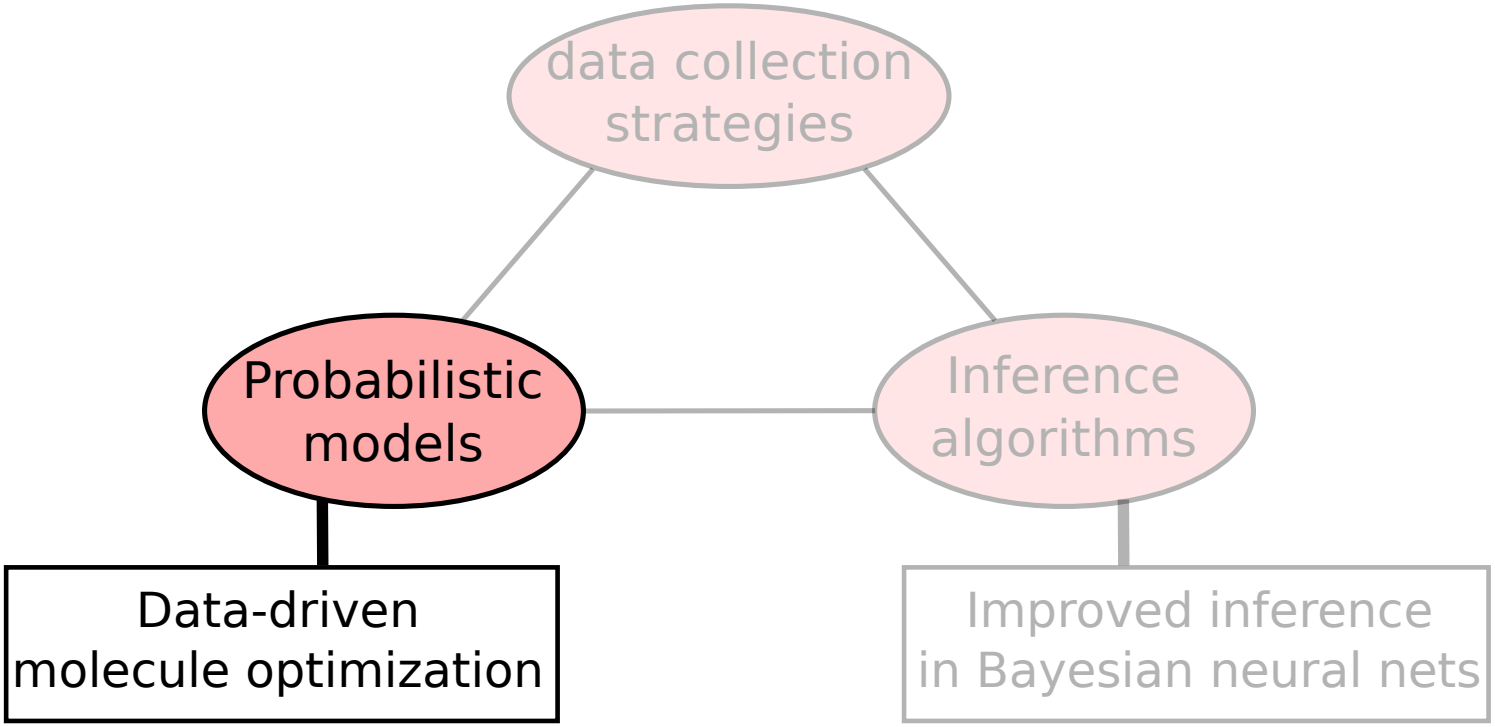
Table: Avg. Test Error and Test Log-likelihood.

	α=1.0 **(EP)**	α=0.5	α=0 **(VB)**	**Sparse GP**
Error	1.28±0.01	**1.08±0.01**	1.13±0.01	1.37±0.01
Log-likelihood	-0.93±0.01	**-0.74±0.01**	-1.39±0.03	-1.23±0.01

Take home messages

Alpha divergence minimization...

1. generalizes existing inference methods (VB and EP).
2. better than any of them in regression problems with neural nets.

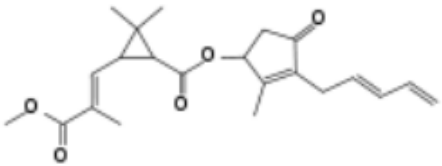
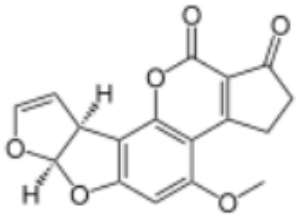
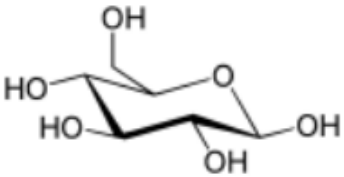

data collection strategies
Probabilistic models
Inference algorithms
Data-driven molecule optimization
Improved inference in Bayesian neural nets

How to represent molecules? We use the SMILES specification.

Molecules are encoded as short **character strings**.

How to represent molecules? We use the SMILES specification.

Molecules are encoded as short **character strings**.

Graph	SMILES string
	CCC[C@@H](O)CC\C=C\C=C\C#CC#C\C=C\CO
	COC(=O)C(\C)=C\C1C(C)(C)[C@H]1C(=O)O[C@@H]2C(C)=C(C(=O)C2)CC=CC=C
	O1C=C[C@H]([C@H]1O2)c3c2cc(OC)c4c3OC(=O)C5=C4CCC(=O)5
	OC[C@@H](O1)[C@@H](O)[C@H](O)[C@@H](O)[C@@H](O)1

Optimization over molecules is **challenging**. Existing methods either use

- Exhaustive search through a fixed library.
- Genetic algorithms.

These require manual specification.

Optimization over molecules is **challenging**. Existing methods either use

- Exhaustive search through a fixed library.
- Genetic algorithms.

These require manual specification.

A continuous, data-driven representation allows us to use **gradient based optimization** and can **leverage chemical databases** to avoid manual tuning.

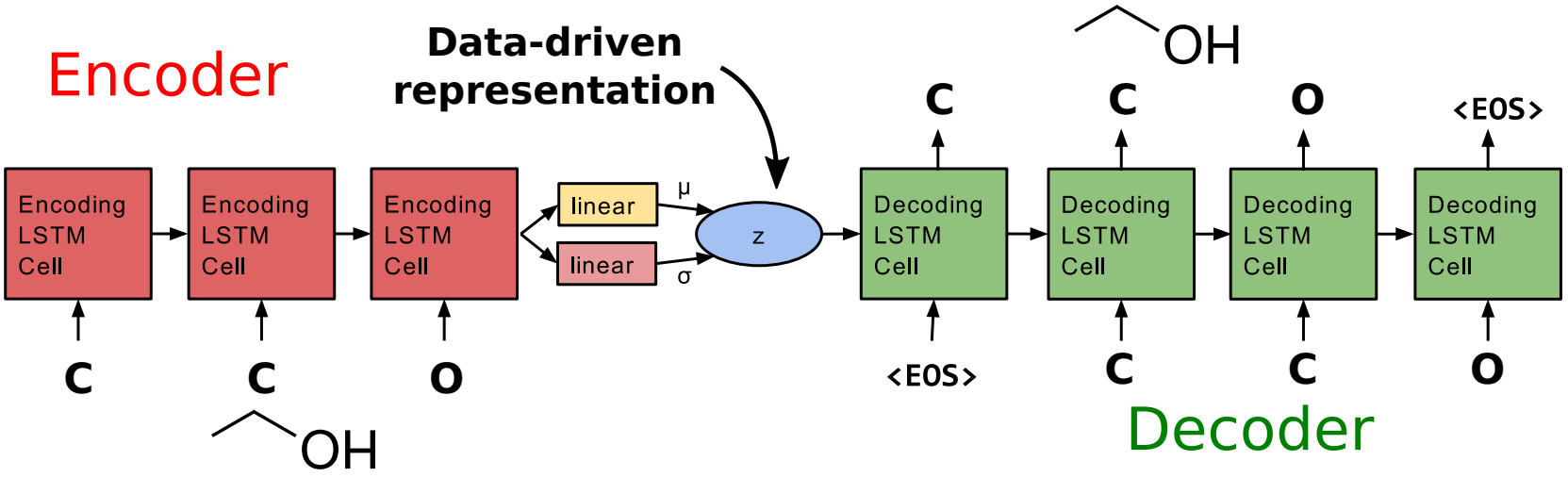
Optimization over molecules is **challenging**. Existing methods either use

- Exhaustive search through a fixed library.
- Genetic algorithms.

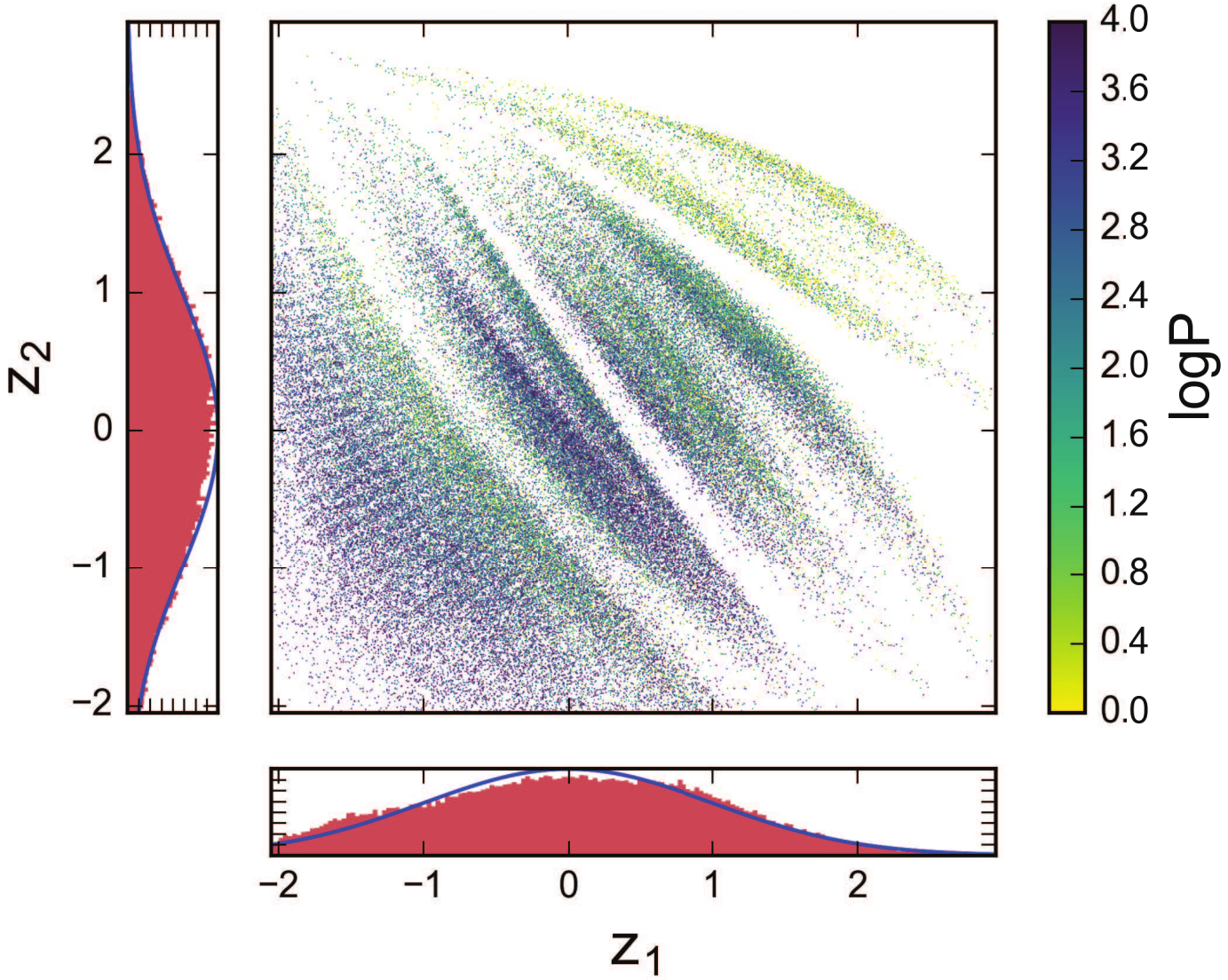
These require manual specification.

A continuous, data-driven representation allows us to use **gradient based optimization** and can **leverage chemical databases** to avoid manual tuning.

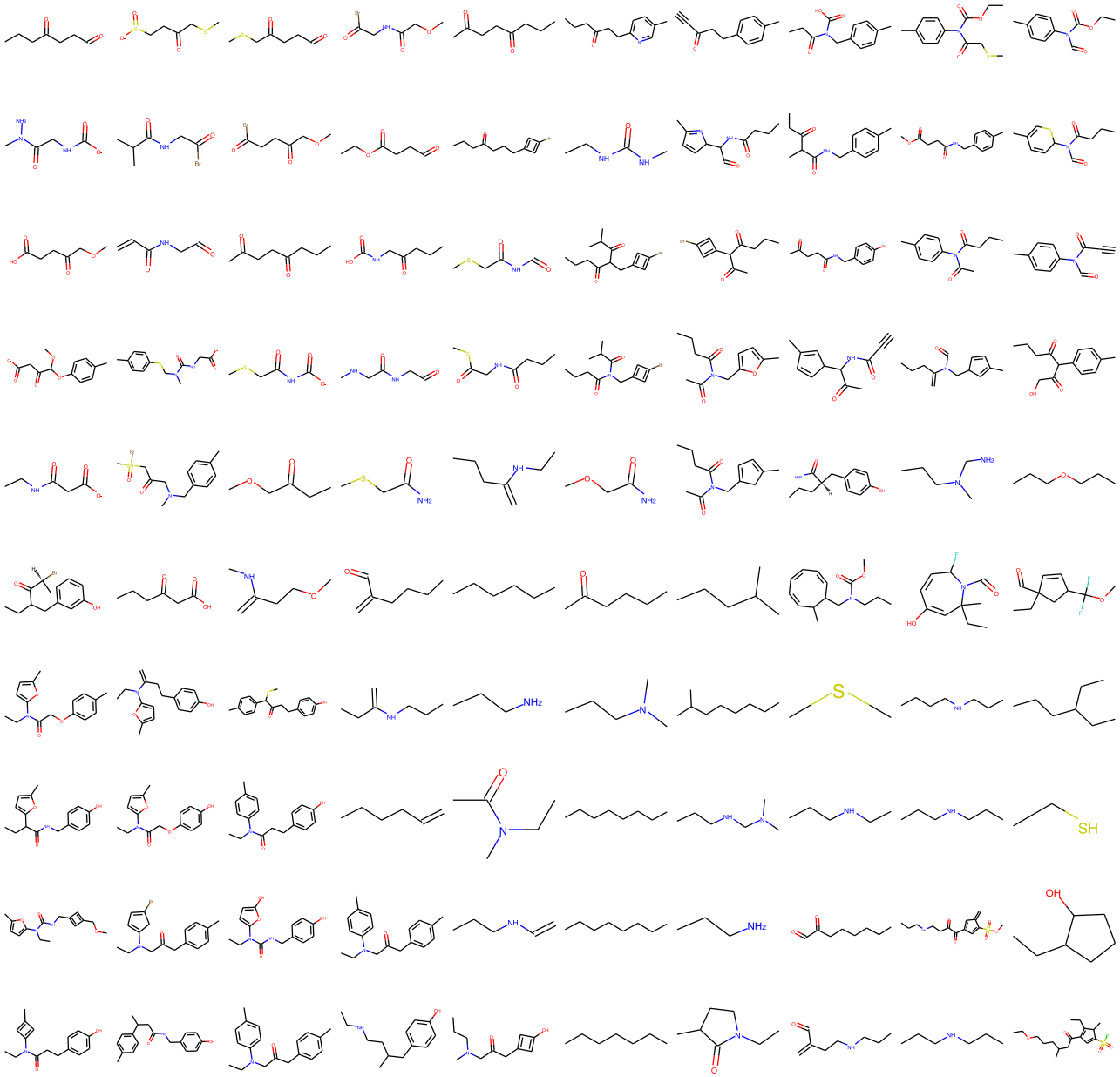
Implemented by using seq2seq variational autoencoder with RNNs on SMILES.



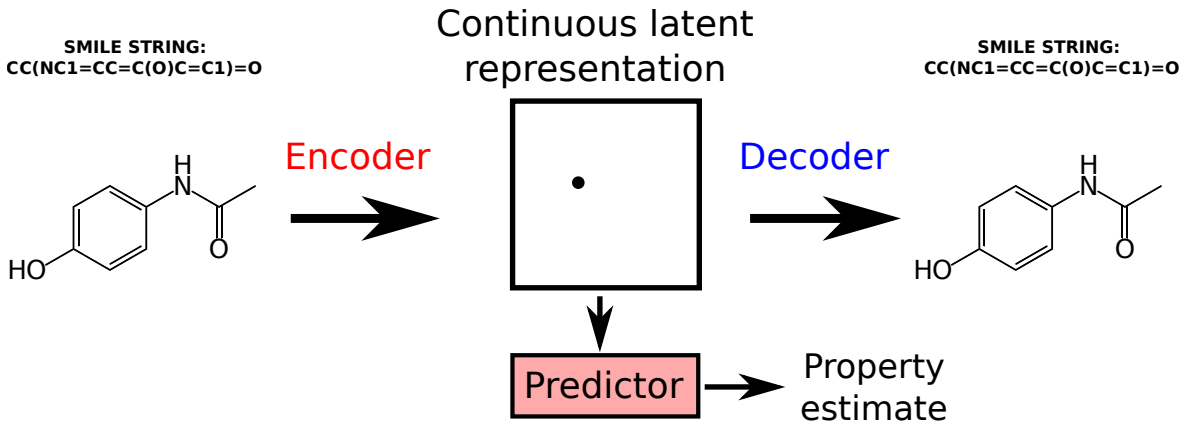
Latent space representation of 220,000 drug-like molecules



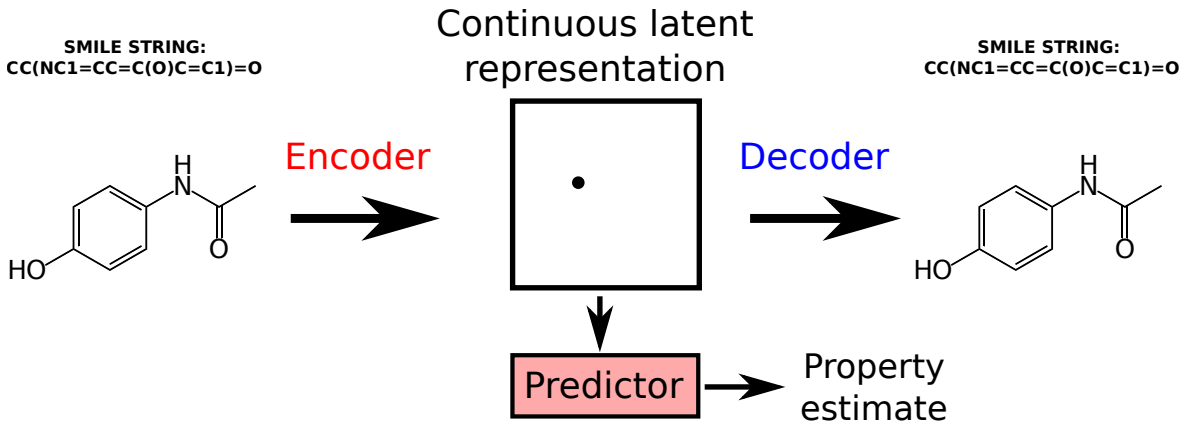
Interpolation in latent space



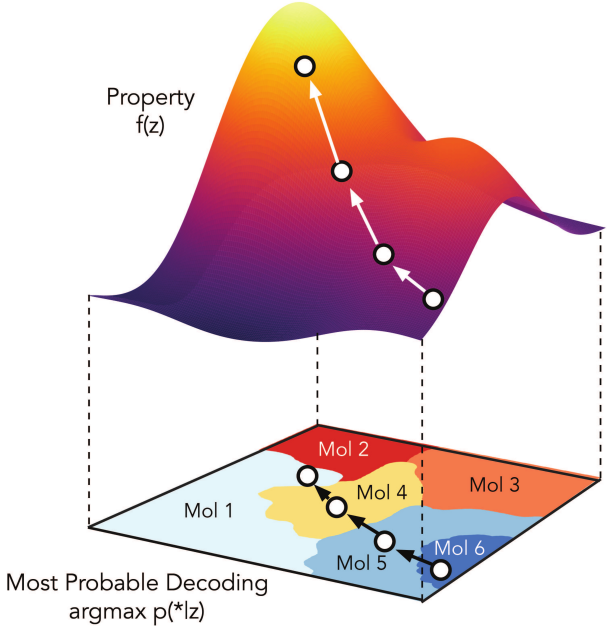
Easy to add a **surrogate model** from latent space to property.



Easy to add a **surrogate model** from latent space to property.



Gradient-based optimization can be used in latent space.



Data: 250,000 molecules from Zinc database. **Surrogate model**: sparse GP.

We aim to maximize the water-octanol partition coefficient (**logP**):

$$J(m) = \text{logP}(m) - \text{SA}(m) - \text{ring-penalty}(m)\,,$$

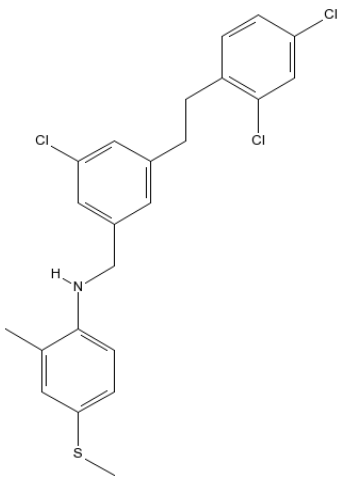
We perform 10 iterations of BO, collecting 50 points per iteration.

Data: 250,000 molecules from Zinc database. **Surrogate model**: sparse GP.

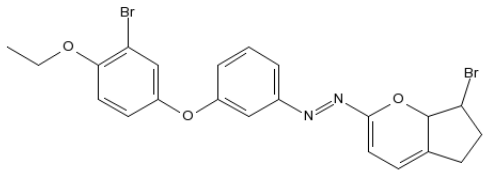
We aim to maximize the water-octanol partition coefficient (**logP**):

$$J(m) = \text{logP}(m) - \text{SA}(m) - \text{ring-penalty}(m)\,,$$

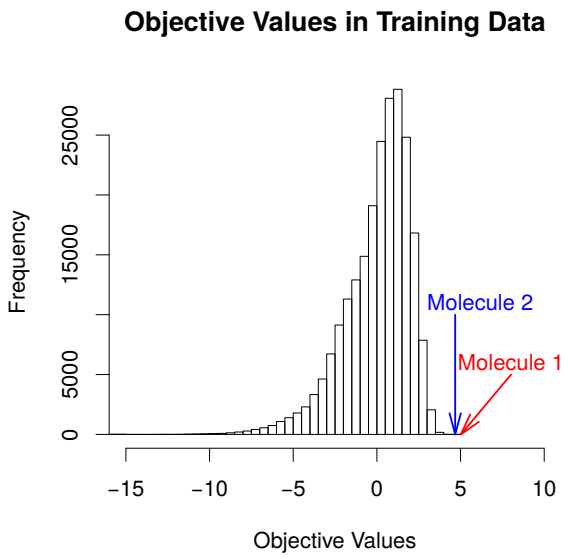
We perform 10 iterations of BO, collecting 50 points per iteration.



Molecule 1



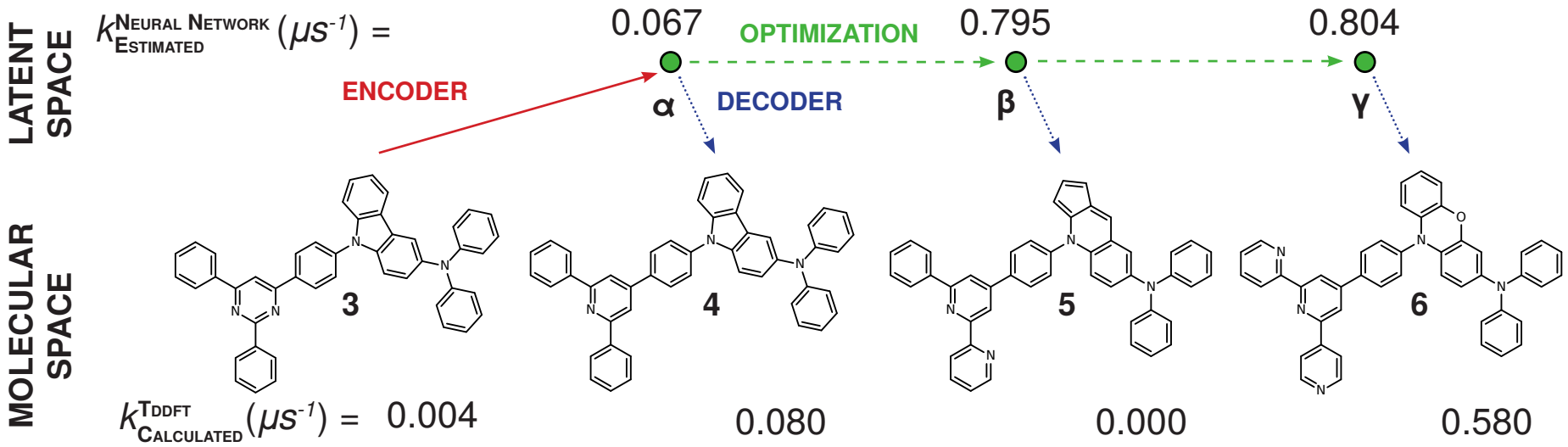
Molecule 2



Found two molecules better than anything seen before.

Local optimization of OLED molecules

We optimize the delayed fluorescence decay rate κ_{TADF}, as estimated from TDDFT computations on 150,000 molecules.



Take home messages

Alpha divergence minimization...

1. generalizes existing inference methods (VB and EP).
2. better than any of them in regression problems with neural nets.

Data-driven molecule optimization...

1. New compounds generated by optimizing in latent space.
2. Can leverage large lists of unlabeled chemical compounds.

Collaborators

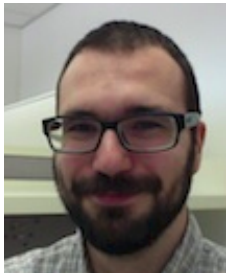
R. P. Adams

J. Aguilera-Ipaguirre

A. Aspuru-Guzik

T. B. Bui

D. Duvenaud



R. Gomez-Bonbardelli

D. Hernández-Lobato

T. Hirzel

Y. Li

M. Rowland

R. E. Turner

Thanks!