Ariadne

Quick how-to

This page provides a very brief overview of main functions available in Ariadne for Gaussian process regression.

First step is to download Ariadne from NuGet and load it into F# interactive. We will also define some data in the Gaussian process format. The rest of this document shows how to do specific tasks with Gaussian processes.

```
 1: #r "Ariadne.dll"
 2: #r "MathNet.Numerics.dll"
 3: open Ariadne
 4: open Ariadne.GaussianProcess
 5: open Ariadne.Kernels
 6: open Ariadne.Optimization
 7:
 8: let data =
 9:   [{Locations = (...)
10:     Observations = (...) }]
```

Create covariance function (kernel)

```
1: let lengthscale = 3.0
2: let signalVariance = 15.0
3: let noiseVariance = 1.0
4:
5: let sqExp = SquaredExp.SquaredExp(lengthscale, signalVariance, noiseVariance)
```

Create Gaussian process with a covariance function

```
1: // method 1
2: let gp1 = GaussianProcess(sqExp.Kernel, Some sqExp.NoiseVariance)
3:
4: // method 2
5: let gp2 = sqExp.GaussianProcess()
```

Compute log likelihood

```
1:  let loglik = gp1.LogLikelihood data
```

Use Metropolis-Hastings to find values for hyperparameters

```
 1: open MathNet.Numerics
 2:
 3: // specify prior distribution
 4: let lengthscalePrior = Distributions.LogNormal.WithMeanVariance(2.0, 4.0)
 5: let variancePrior = Distributions.LogNormal.WithMeanVariance(1.0, 5.0)
 6: let noisePrior = Distributions.LogNormal.WithMeanVariance(0.5, 1.0)
 7: let prior = SquaredExp.Prior(lengthscalePrior, variancePrior, noisePrior)
 8:
 9: // get posterior mean values for hyperparameters
10: let sqExpMH =
11:     sqExp
12:     |> SquaredExp.optimizeMetropolis data MetropolisHastings.defaultSettings prior
```

Use gradient descent to find values for hyperparameters

```
1: let settings =
2:     { GradientDescent.Iterations = 10000;
3:       GradientDescent.StepSize = 0.01}
```


DOCUMENTATION

```
 4:
 5: let gradientFunction parameters = SquaredExp.fullGradient data parameters
 6:
 7: // Run gradient descent
 8: let sqExpGD =
 9:     gradientDescent gradientFunction settings (sqExp.Parameters)
10:     |> SquaredExp.ofParameters
```

Compute prediction for new time points (locations)

```
1: let allYears = [| 1985.0 .. 2015.0 |]
2: let predictValues, predictVariance = allYears |> gp1.Predict data
```

Compute predictive log likelihood for new observed data

```
1: let newObservation = {
2:     Locations = [| 2015.0 |]; Observations = [| 10.0 |]}
3:
4: let predictiveLoglik = gp1.PredictiveLogLikelihood data newObservation
```

Plot the fitted Gaussian process

```
1:  #r "FSharp.Charting.dll"
2:
3:  GaussianProcess.plot data gp1
4:  GaussianProcess.plotRange (1980.0, 2020.0) data gp1
```