# Bayesian optimization w/ MATLAB

(Know your competition)

Le Minh Cristian

# Problem

Asymetric Hubbard model:

$$\begin{aligned}\hat{H}(t) =& t(\hat{c}^{\dagger}_{\sigma 1}\hat{c}_{\sigma 2} + \text{h.c.}) + U\hat{n}_{\uparrow i}\hat{n}_{\downarrow i} \\ &+ \frac{v_0}{2}(\hat{n}_2 - \hat{n}_1) + \frac{v_1(t)}{2}(\hat{n}_1 - \hat{n}_2)\end{aligned}$$

Hartree-Fock:

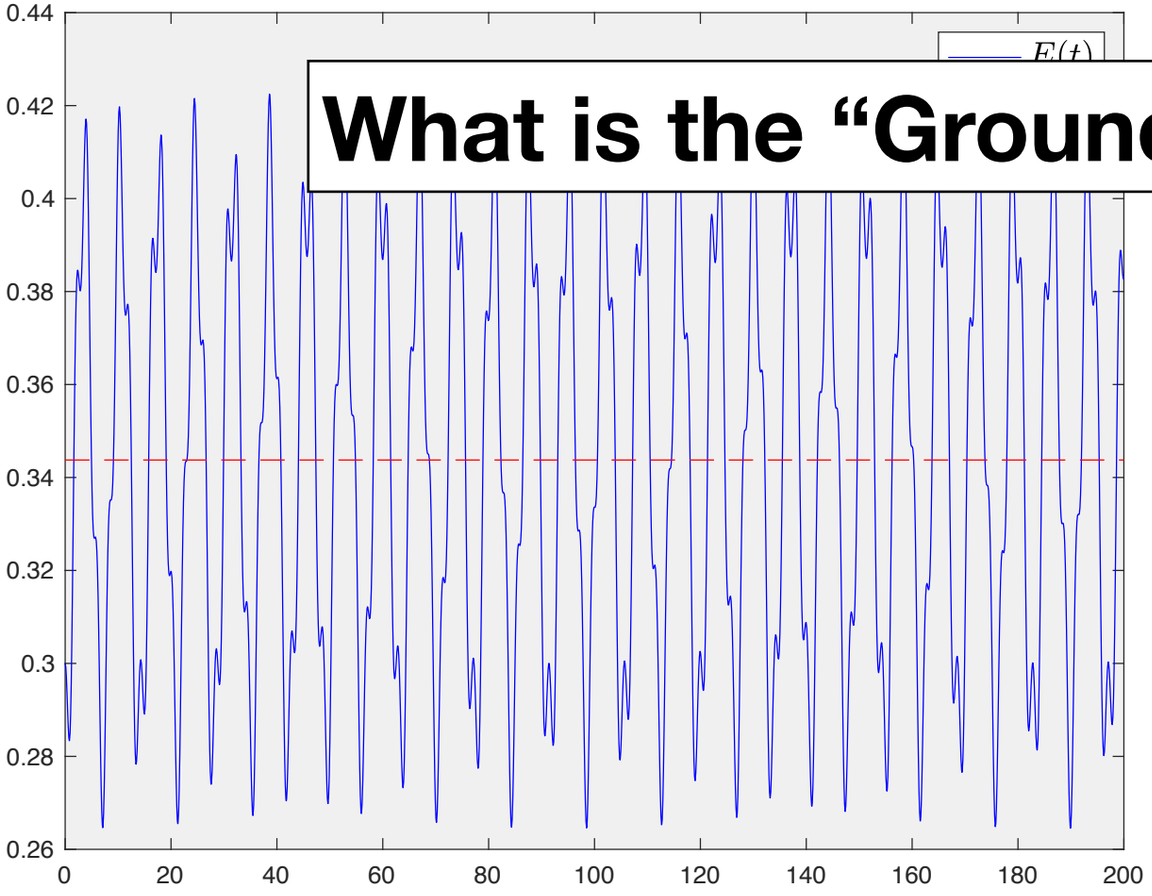
$$\hat{F}(t) = \begin{bmatrix} \frac{v_0}{2} + \frac{v_1(t)}{2} & t \\ t & -\frac{v_0}{2} - \frac{v_1(t)}{2} \end{bmatrix} + U \begin{bmatrix} |\phi_1(t)|^2 & 0 \\ 0 & |\phi_2(t)| \end{bmatrix}$$

$$\hat{F}(t)\,|\phi(t)\rangle = i\partial_t\,|\phi(t)\rangle$$

# Problem

$$\bar{E} = \frac{1}{T}\int_0^T \langle \phi_\uparrow(t)\phi_\downarrow(t)|\hat{H}(t)|\phi_\uparrow(t)\phi_\downarrow(t)\rangle$$

$$|\phi(t+\delta_t)\rangle = |\phi(t)\rangle - i\delta_t \hat{F}(t)\,|\phi(t)\rangle$$




**What is the “Ground state”?**

$$|\phi(0)\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

# Propagator

## (Quick and dirty)

## ode45

Solve nonstiff differential equations — medium order method

### Syntax

```
[t,y] = ode45(odefun,tspan,y0)
[t,y] = ode45(odefun,tspan,y0,options)
[t,y,te,ye,ie] = ode45(odefun,tspan,y0,options)
sol = ode45( __ )
```

### Description

`[t,y] = ode45(odefun,tspan,y0)`, where `tspan = [t0 tf]`, integrates the system of differential equations $y' = f(t, y)$ from `t0` to `tf` with initial conditions `y0`. Each row in the solution array `y` corresponds to a value returned in column vector `t`. example

All MATLAB® ODE solvers can solve systems of equations of the form $y' = f(t, y)$, or problems that involve a mass matrix, $M(t, y)y' = f(t, y)$. The solvers all use similar syntaxes. The `ode23s` solver only can solve problems with a mass matrix if the mass matrix is constant. `ode15s` and `ode23t` can solve problems with a mass matrix that is singular, known as differential-algebraic equations (DAEs). Specify the mass matrix using the `Mass` option of `odeset`.

`ode45` is a versatile ODE solver and is the first solver you should try for most problems. However, if the problem is stiff or requires high accuracy, then there are other ODE solvers that might be better suited to the problem. See Choose an ODE Solver for more information.

`[t,y] = ode45(odefun,tspan,y0,options)` also uses the integration settings defined by `options`, which is an argument created using the `odeset` function. For example, use the `AbsTol` and `RelTol` options to specify absolute and relative error tolerances, or the `Mass` option to provide a mass matrix. example

`[t,y,te,ye,ie] = ode45(odefun,tspan,y0,options)` additionally finds where functions of $(t,y)$, called event functions, are zero. In the output, `te` is the time of the event, `ye` is the solution at the

# Bayesian Optimization

## bayesopt

Select optimal machine learning hyperparameters using Bayesian optimization

### Syntax

```
results = bayesopt(fun,vars)
results = bayesopt(fun,vars,Name,Value)
```

### Description

`results = bayesopt(fun,vars)` attempts to find values of `vars` that minimize `fun(vars)`. example

**Note**

To include extra parameters in an objective function, see Parameterizing Functions (MATLAB).

`results = bayesopt(fun,vars,Name,Value)` modifies the optimization process according to the `Name,Value` arguments. example

### Examples

collapse all

#### Create a `BayesianOptimization` Object Using `bayesopt`

This example shows how to create a `BayesianOptimization` object by using `bayesopt` to minimize cross-validation loss.

Open Live Script

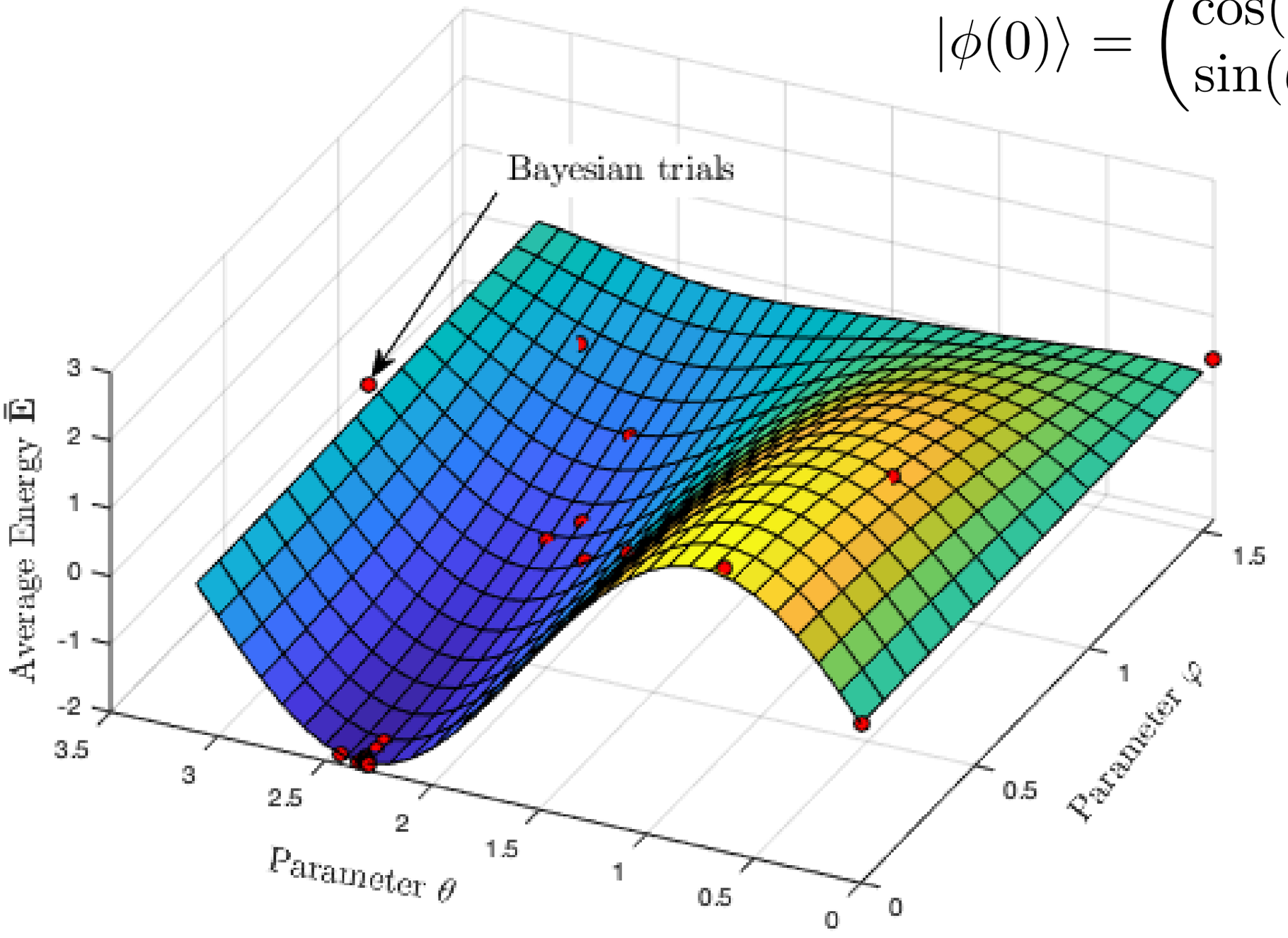
Optimize hyperparameters of a KNN classifier for the `ionosphere` data, that is, find KNN hyperparameters that minimize the cross-validation loss. Have `bayesopt` minimize over the following hyperparameters:

- Nearest-neighborhood sizes from 1 to 30
- Distance functions `'chebychev'`, `'euclidean'`, and `'minkowski'`.

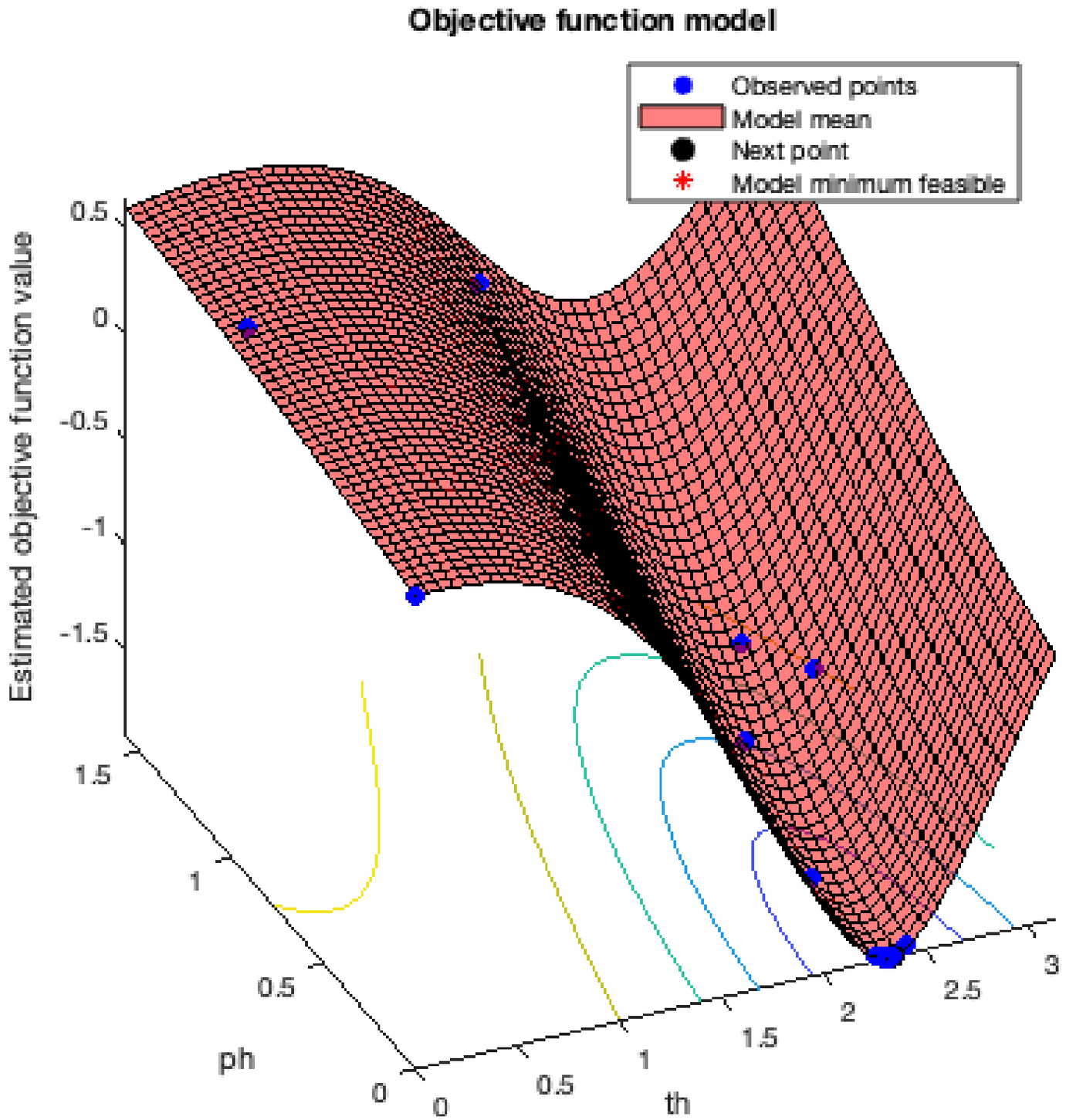
For reproducibility, set the random seed, set the partition, and set the `AcquisitionFunctionName` option to

# Bayesian Optimization

$$|\phi(0)\rangle = \begin{pmatrix} \cos(\theta) \\ \sin(\theta)e^{-i\varphi} \end{pmatrix}$$



# Bayesian Optimization



# Bayesian Optimization

## w/ MATLAB

**Advantage cf. COMBO:**

- Dynamic grid
- Intuitive on-the-fly visualization
- MATLAB integration (Easier to code?)

# Bayesian Optimization

## w/ MATLAB

**Current example:**

- Too simple: only one global minimum

- Parameter dimension is within limits < 50

➡Gradient methods out-perform
[*fminunc* (20s) > *bayseopt* (40s)]