Regularization with Weight Decay

In the following experiments use the data provided in the files `in.dta`, `out.dta` as a training and test set respectively. Each line of the files corresponds to a two-dimensional input $\mathbb{x} = (x_1, x_2)$, so that $\mathcal{X} = \mathbb{R}^2$, followed by the corresponding label from $\mathcal{Y} = \{-1, 1\}$. We are going to apply Linear Regression with a non-linear transformation for classification. The nonlinear transformation is given by'

$$\varphi(x_1, x_2) = \left(1, x_1, x_2, x_1^2, x_2^2, x_1 x_2, |x_1 - x_2|, |x_1 + x_2|\right).$$

Recall that the classification error is defined as the fraction of misclassified points.

```
(* Clear globals *)
Clear[Generate𝒳, DoLinearRegressionExperiment, ClassificationE]
```

```
Generate𝒳[OptionsPattern[]] :=
 Module[{
   𝒟y, 𝒟, y, f, d = 2,
   f1, f2,
   m, a, b, c,
   t1, t2,
   dotTest1, dotTest2, DotTest
  },

  f1 = RandomReal[{-1, 1}, {1, d}][[1]];
  f2 = RandomReal[{-1, 1}, {1, d}][[1]];

       f1[[2]] - f2[[2]]
  m = ─────────────────;
       f1[[1]] - f2[[1]]
  a = -m;
  b = 1;
  c = m f1[[1]] - f1[[2]];
  f = {{c}, {a}, {b}};

  (* f should not dot to zero for our two original points! *)
  t1 = {1, f1[[1]], f1[[2]]};
  t2 = {1, f2[[1]], f2[[2]]};

  DotTest[v_, s_] := If[Abs[v] > 0.00000000001, Throw[s], 0];
  DotTest[t1.f, "t1 dot test failed"];
  DotTest[t2.f, "t2 dot test failed"];

  (*𝒟 = RandomReal[{-1,1},{OptionValue[𝒟Size],d}];*)
  𝒟y = Import[OptionValue[ℝ2DataFile], "Table"];
  𝒟 = 𝒟y[[All(*1;;15*), {1, d}]];
  y = 𝒟y[[All(*1;;15*), d + 1]];
  {𝒟, f, y}
 ]
Options[Generate𝒳] = {ℝ2DataFile → "in.dta"};
```

```
LinearTarget[f_, 𝕏_] := Sign[𝕏.f];
NoFeature[𝕏_] := 𝕏;
ϕ[𝕏_] := {1, #2, #3, #2^2, #3^2, #2 #3, Abs[#2 - #3], Abs[#2 + #3]} & @@@ 𝕏;
NoRegularizer[𝕏_, λ_] := PseudoInverse[𝕏] (*Inverse[𝕏ᵀ.𝕏].𝕏ᵀ*);
Tikhonov[𝕏_, λ_] := Inverse[𝕏ᵀ.𝕏 + λ IdentityMatrix[Dimensions[𝕏][[2]]]].𝕏ᵀ;
DoLinearRegressionExperiment[𝒳_, OptionsPattern[]] :=
 Module[{
    𝒟, f, y, 𝕪,
    𝕏, 𝕏f, 𝕏fdag,
    𝕨
   },
   {𝒟, f, y} = 𝒳;
   𝕏 = Function[x, Prepend[x, 1]] /@ 𝒟;
   𝕏f = OptionValue[DataFeature][𝕏];
   𝕪 = y;
   𝕏fdag = OptionValue[Regularizer][𝕏f, OptionValue[λ]];
   𝕨 = 𝕏fdag.𝕪;
   {𝕨, 𝕏, 𝕪, 𝒟, f}
 ]
Options[DoLinearRegressionExperiment] = {TargetFunction → LinearTarget,
    DataFeature → ϕ, Regularizer → NoRegularizer, λ → 0};

ClassificationE[𝕏_, 𝕨_, 𝕪_, OptionsPattern[]] :=
 (*
 Determine classification error for hypothesis 𝕨 when
    given data collection 𝕏 and labels 𝕪.  NOTE that this is a
    generic classification error function that is E_in/E_out-agnostic
 *)
 Module[{
    N, misses, sumOfMisses
   },
   N = Length[𝕪];
   misses =
    MapThread[If[#1 ≠ #2, 1, 0] &, {Sign[OptionValue[DataFeature][𝕏].𝕨], 𝕪}];
   sumOfMisses = Total[misses];
   1/N sumOfMisses
 ]
Options[ClassificationE] = {DataFeature → ϕ};
```

Linear regression with non-linear transform

Run Linear Regression on the training set after performing the non-linear transformation to determine E_{in} and E_{out}.

```
(* Clear globals *)
Clear[Experiment1, e1Ein, e1Eout]
```

```
Experiment1[] :=
 (*
 Perform linear regression using the provided in-sample file,
 then determine Ein from it and Eout from the provided out-
  of-sample data+labelage.
 *)
 Module[{
   Ein, Eout,
   𝒟in, 𝒟out,
   𝕏in, 𝕏out,
   𝕪in, 𝕪out,
   𝕨, f, d = 2
  },

  {𝕨, 𝕏in, 𝕪in, 𝒟in, f} =
   DoLinearRegressionExperiment[Generate𝒳[ℝ2DataFile → "in.dta"]];
  Ein = ClassificationE[𝕏in, 𝕨, 𝕪in];

  {𝒟out, f, 𝕪out} = Generate𝒳[ℝ2DataFile → "out.dta"];
  𝕏out = Function[x, Prepend[x, 1]] /@ 𝒟out;
  Eout = ClassificationE[𝕏out, 𝕨, 𝕪out];
  {Ein, Eout}
 ]

{e1Ein, e1Eout} = Experiment1[];
StringForm[
 "Linear regression results:\nEin=``, Eout=``", e1Ein × 1., e1Eout × 1.]
```

Linear regression results:
E_{in}=0.02857142857142857`, $E_{out}$=0.084`

Linear regression with weight decay, k=-3

Now add weight decay to Linear Regression, that is, add the term $\frac{\lambda}{N} \sum_{i=0}^{7} w_i^2$ to the squared in-sample error, using $\lambda = 10^k$, k = -3.

```
(* Clear globals *)
Clear[Experiment2, e2Ein, e2Eout, e2λ]
```

```
Experiment2[regularizerλ_] :=
 (*
 Perform linear regression using the provided in-sample file,
 then determine E_in from it and E_out from the provided out-
  of-sample data+labelage.
 *)
 Module[{
   Ein, Eout,
   𝒟in, 𝒟out,
   𝕏in, 𝕏out,
   𝕪in, 𝕪out,
   𝕨, f, d = 2
  },

  {𝕨, 𝕏in, 𝕪in, 𝒟in, f} = DoLinearRegressionExperiment[
    Generate𝒳[ℝ2DataFile → "in.dta"], Regularizer → Tikhonov, λ → regularizerλ];
  Ein = ClassificationE[𝕏in, 𝕨, 𝕪in];

  {𝒟out, f, 𝕪out} = Generate𝒳[ℝ2DataFile → "out.dta"];
  𝕏out = Function[x, Prepend[x, 1]] /@ 𝒟out;
  Eout = ClassificationE[𝕏out, 𝕨, 𝕪out];
  {Ein, Eout}
 ]
```

```
e2λ = 10^-3;
{e2Ein, e2Eout} = Experiment2[e2λ];
StringForm[
 "Linear regression results (w/ λ=`` regularization):\nE_in=``, E_out=``",
 e2λ, e2Ein × 1., e2Eout × 1.]
```

Linear regression results (w/ $\lambda=\frac{1}{1000}$ regularization):
E_{in}=0.02857142857142857`, $E_{out}$=0.08`

Linear regression with weight decay, k=3

```
(* Clear globals *)
Clear[e3Ein, e3Eout, e3λ]
```

```
e3λ = 10^3;
{e3Ein, e3Eout} = Experiment2[e3λ];
StringForm[
 "Linear regression results (w/ λ=`` regularization):\nE_in=``, E_out=``",
 e3λ, e3Ein × 1., e3Eout × 1.]
```

Linear regression results (w/ λ=1000 regularization):
E_{in}=0.37142857142857144`, $E_{out}$=0.436`

Linear regression with weight decay, various k

Looking for value of k that achieves the smallest out-of-sample classification error, trying integer k from -2 to 2.

```
(* Clear globals *)
Clear[e4Ein, e4Eout, e4λ, i]
```

```
e4λ = {10^2, 10^1, 10^0, 10^-1, 10^-2};

i = 1;
{e4Ein, e4Eout} = Experiment2[e4λ[[i]]];
StringForm[
 "Linear regression results (w/ λ=`` regularization):\nE_in=``, E_out=``",
 e4λ[[i]], e4Ein × 1., e4Eout × 1.]

i = 2;
{e4Ein, e4Eout} = Experiment2[e4λ[[i]]];
StringForm[
 "Linear regression results (w/ λ=`` regularization):\nE_in=``, E_out=``",
 e4λ[[i]], e4Ein × 1., e4Eout × 1.]

i = 3;
{e4Ein, e4Eout} = Experiment2[e4λ[[i]]];
StringForm[
 "Linear regression results (w/ λ=`` regularization):\nE_in=``, E_out=``",
 e4λ[[i]], e4Ein × 1., e4Eout × 1.]

i = 4;
{e4Ein, e4Eout} = Experiment2[e4λ[[i]]];
StringForm[
 "Linear regression results (w/ λ=`` regularization):\nE_in=``, E_out=``",
 e4λ[[i]], e4Ein × 1., e4Eout × 1.]

i = 5;
{e4Ein, e4Eout} = Experiment2[e4λ[[i]]];
StringForm[
 "Linear regression results (w/ λ=`` regularization):\nE_in=``, E_out=``",
 e4λ[[i]], e4Ein × 1., e4Eout × 1.]
```

Linear regression results (w/ λ=100 regularization):
E_{in}=0.2\`, $E_{out}$=0.228\`

Linear regression results (w/ λ=10 regularization):
E_{in}=0.05714285714285714\`, $E_{out}$=0.124\`

Linear regression results (w/ λ=1 regularization):
E_{in}=0.\`, $E_{out}$=0.092\`

Linear regression results (w/ $\lambda=\frac{1}{10}$ regularization):
E_{in}=0.02857142857142857\`, $E_{out}$=0.056\`

Linear regression results (w/ $\lambda=\frac{1}{100}$ regularization):
E_{in}=0.02857142857142857\`, $E_{out}$=0.084\`