プログラミング演習2：ロジスティック回帰

機械学習

前書き

この演習では、ロジスティック回帰を実装し、2つに適用します
異なるデータセット。プログラミング演習を開始する前に、私たちは強く
ビデオ講義を見て、レビューの質問を完了することをお勧めします
関連トピックについて。

エクササイズを開始するには、スターターをダウンロードする必要があります
コードを作成し、その内容を完了したいディレクトリに解凍します
運動。必要に応じて、Octave / MATLABでcdコマンドを使用して、
この演習を開始する前に、このディレクトリ。

Octave / MATLABのインストール手順は、「En-
コースWebサイトの「セットアップ手順」を参照してください。

この演習に含まれるファイル

ex2.m-演習をステップごとに実行するOctave / MATLABスクリプト
ex2 reg.m-演習の後半のOctave / MATLABスクリプト
ex2data1.txt-演習の前半のトレーニングセット
ex2data2.txt-演習の後半のトレーニングセット
submit.m-サーバーにソリューションを送信する送信スクリプト
mapFeature.m-多項式特徴を生成する関数
plotDecisionBoundary.m-分類器の決定境界をプロットする関数-
アリ
[*] plotData.m-2D分類データをプロットする関数
[*] sigmoid.m-シグモイド関数
[*] costFunction.m-ロジスティック回帰コスト関数
[*] predict.m-ロジスティック回帰予測関数
[*] costFunctionReg.m-正規化されたロジスティック回帰コスト

は、完了する必要があるファイルを示します

2ページ

演習を通して、スクリプトex2.mおよびex2 reg.mを使用します。
これらのスクリプトは、問題のデータセットを設定し、関数を呼び出します
あなたが書くこと。どちらも変更する必要はありません。あなただけです
以下の手順に従って、他のファイルの関数を変更する必要があります
この割り当て。

助けを得る場所

このコースの演習では、Octaveを使用します[1]またはMATLAB、高レベルプログラム-
数値計算に適した明言語。お持ちでない場合
OctaveまたはMATLABがインストールされています。次のインストール手順を参照してください。
コースWebサイトの「環境設定手順」。

Octave / MATLABコマンドラインで、helpに続けてfunc-
名前には、組み込み関数のドキュメントが表示されます。たとえば、ヘルプ
plotは、プロットに関するヘルプ情報を表示します。の詳細なドキュメント
Octave関数は、Octaveのドキュメントページにあります。。マット-
LABドキュメントは、MATLABドキュメントページで見つけることができます。。

また、オンラインディスカッションを使用して、元の
他の生徒と一緒に ただし、書かれたソースコードは見ないでください
他人によって、またはソースコードを他人と共有する。

1ロジスティック回帰

演習のこの部分では、ロジスティック回帰モデルを作成して
学生が大学に入学するかどうかを予測します。

あなたが大学の学部の管理者であり、
あなたは彼らに基づいて各志願者の入学のチャンスを決定したい
2つの試験の結果。以前の応募者の履歴データがあります
ロジスティック回帰のトレーニングセットとして使用できます。 各トレーニングについて
たとえば、2つの試験と入学で申請者のスコアがあります
決定。

あなたの仕事は、申請者の推定モデルモデルを構築することです
これらの2つの試験のスコアに基づいた入学の確率。この概要
ex2.mのフレームワークコードが演習をガイドします。

[1] Octaveは、MATLABの無料の代替です。プログラミング演習では、無料です
OctaveまたはMATLABを使用します。

3ページ

1.1データの視覚化

学習アルゴリズムの実装を開始する前に、常に次のことを行ってください。
可能であればデータを視覚化します。ex2.mの最初の部分では、コードは
関数plotDataを呼び出して、データを2次元プロットに表示します。

次に、plotDataのコードを完成させて、図を表示します。
図1のように、軸は2つの試験の得点、および正と
負の例は異なるマーカーで示されています。

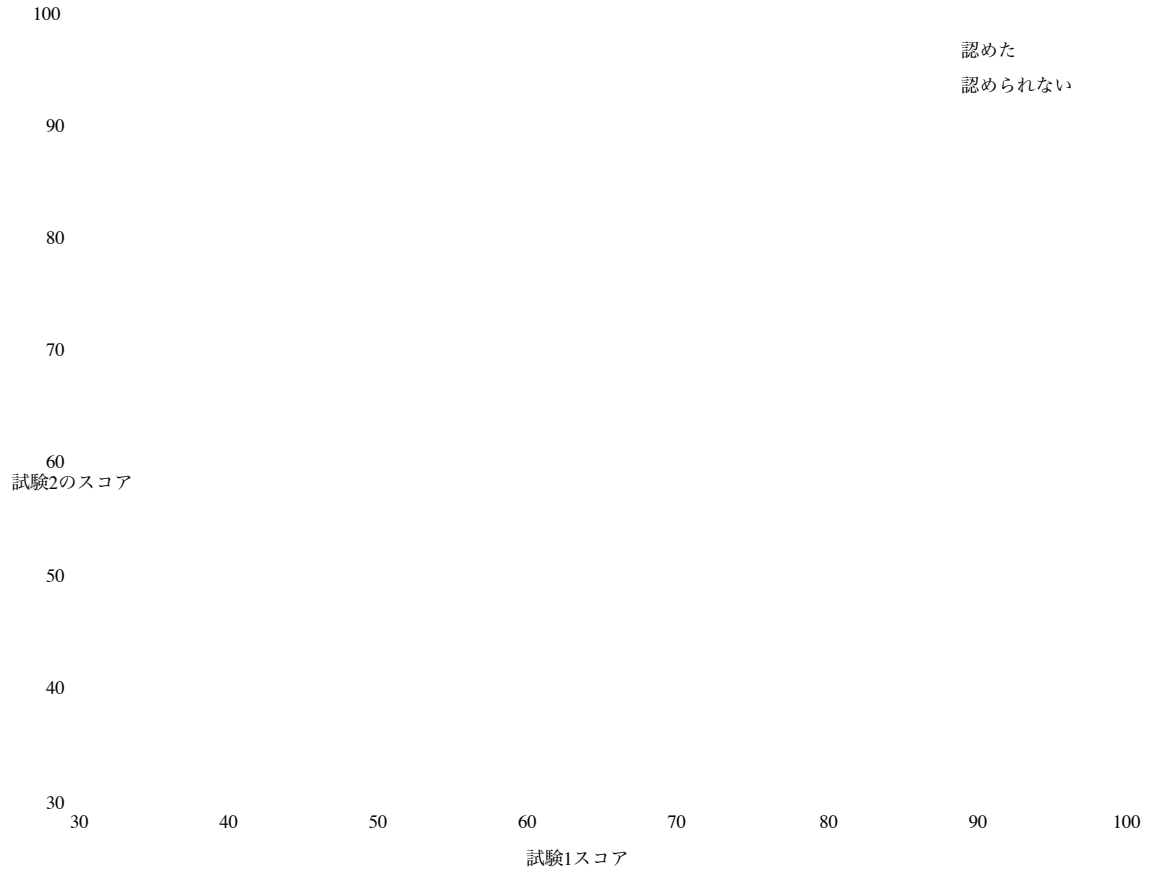

図1：トレーニングデータの散布図

プロットに慣れるために、plotData.mを残しました
空なので、自分で実装してみることができます。ただし、これはオプションです
（無段階）運動。また、以下の実装も提供します。
コピーするか、参照してください。 サンプルをコピーする場合は、必ず学習してください
Octave / MATLABを調べて、各コマンドが実行していること
ドキュメンテーション。

```
％ポジティブおよびネガティブな例のインデックスを見つける
pos = find（y == 1）; neg = find（y == 0）;

％プロットの例
plot（X（pos、1）、X（pos、2）、'k +'、'LineWidth'、2、...
          'MarkerSize' 、7）;
plot（X（neg、1）、X（neg、2）、'ko'、'MarkerFaceColor'、'y'、...
          'MarkerSize' 、7）;
```


1.2実装

1.2.1ウォームアップ演習：シグモイド関数

実際のコスト関数を開始する前に、ロジスティック回帰を思い出してください。
sion仮説は次のように定義されます：

$$時間_{\theta}（X）= G（\theta^{T} X）$$

ここで、関数gはシグモイド関数です。シグモイド関数は次のように定義されます：

$$g（z）= \frac{1}{1 + e^{-z}}$$ 。

最初のステップは、この関数をsigmoid.mに実装することです。
プログラムの残りの部分から呼び出されます。終了したら、いくつかテストしてみてください
Octave / MATLABコマンドラインでsigmoid（x）を呼び出して値を設定します。ために
xの大きな正の値、シグモイドは1に近いはずです。
負の値の場合、シグモイドは0に近いはずです。シグモイドの評価（0）
正確に0.5になるはずです。コードはベクターでも機能し、
マトリックス。行列の場合、関数はシグモイドを実行する必要があります
すべての要素で機能します。

Oc-でsubmitと入力して、採点のためにソリューションを送信できます。
tave / MATLABコマンドライン。提出スクリプトにより、
ログイン電子メールと送信トークンを入力し、必要なファイルを尋ねます
提出する。次のWebページから送信トークンを取得できます。
割り当て。

ここで、ソリューションを送信する必要があります。

1.2.2コスト関数と勾配

次に、ロジスティック回帰のコスト関数と勾配を実装します。
costFunction.mのコードを完成させて、コストと勾配を返します。

ロジスティック回帰のコスト関数は

$$J（\theta）= \frac{1}{m} \sum_{i=1}^{m} [-y_{(i)} \log（h_{\theta}（x_{(i)}）） -（1 - y_{(i)}）\log（1 - h_{\theta}（x_{(i)}））]、$$

コストの勾配は、$\theta_{番目}$と同じ長さのベクトルです。
要素（j = 0,1、...、nの場合）は次のように定義されます。

$$\frac{\partial J（\theta）}{\partial \theta_{J}} = \frac{1}{m} \sum_{i=1}^{m} （h_{\theta}（x_{(i)}）- y_{(i)_j}）^{(私)} x$$

この勾配は線形回帰の勾配と同じように見えますが、
dient、線形とロジスティック回帰のため、式は実際に異なります
h_{θ}（x）の定義が異なります。

θの完了したタ。ex2.mは初期値を使用してcostFunctionを呼び出します

ここで、ソリューションを送信する必要があります。

1.2.3 fminuncを使用した学習パラメーター

前の割り当てで、線形再構成の最適なパラメーターを見つけました。
勾配降下法の実装による退行モデル。コスト関数を書きました
勾配を計算し、それに応じて勾配降下ステップを実行しました。
今回は、勾配降下ステップを実行する代わりに、Octave /-を使用します
fminuncと呼ばれるMATLAB組み込み関数。

Octave / MATLABのfminuncは、最小値を見つける最適化ソルバーです。
制約のない2つの機能。ロジスティック回帰の場合、
パラメータθでコスト関数J（θ）を最適化します。

具体的には、fminuncを使用して最適なパラメーターθを見つけます。
（Xおよびyの固定データセットが与えられた場合のロジスティック回帰コスト関数
値）。次の入力をfminuncに渡します。

- 最適化しようとしているパラメーターの初期値。
- トレーニングセットと特定のθを指定すると、計算する関数
  データセットのθに関するロジスティック回帰コストと勾配
  （X、y）

ex2.mでは、fminuncを正しいコードで呼び出すために記述されたコードが既にあります
引数。

2 最適化の制約は、多くの場合、パラメーターの制約を参照します。たとえば、
θが取り得る可能な値を制限する制約（例、θ≤1）。ロジスティック回帰
θは実際の値を取ることができるため、このような制約はありません。


```
%fminuncのオプションを設定します
options = optimset（'GradObj'、'on'、'MaxIter'、400）；

%fminuncを実行して最適なシータを取得
%この関数はthetaとコストを返します
[シータ、コスト] = ...
        fminunc（@（t）（costFunction（t、X、y））、初期シータ、オプション）；
```

このコードスニペットでは、最初にfminuncで使用するオプションを定義しました。
具体的には、GradObjオプションをonに設定します。これは、fminuncに、
関数は、コストと勾配の両方を返します。これにより、fminuncが
関数を最小化するときに勾配を使用します。さらに、
MaxIterオプションを400に設定すると、fminuncは最大で400ステップ前に実行されます
終了します。

最小化する実際の関数を指定するには、「ショートハンド」を使用します
@（t）（costFunction（t、X、y））で関数を指定するため。この
引数tを使用して、costFunctionを呼び出す関数を作成します。この
fminuncで使用するためにcostFunctionをラップできます。

costFunctionを正しく完了した場合、fminuncは収束します
適切な最適化パラメータで、コストの最終値を返す
およびθ。fminuncを使用することにより、ループを記述する必要がないことに注意してください。
または、勾配降下の場合と同じように学習率を設定します。これがすべてです
fminuncによる処理：コストを計算する関数を提供するだけで済みます
グラデーション。

fminuncが完了すると、ex2.mはcostFunction関数を呼び出します
θの最適なパラメーターを使用します。あなたはコストが約であることを見る必要があります
0.203。

この最終的なθ値は、次に決定境界をプロットするために使用されます
トレーニングデータ。図2のような図になります。私たちも奨励します
plotDecisionBoundary.mのコードを見て、そのようなプロット方法を確認してください
θ値を使用した境界。

1.2.4ロジスティック回帰の評価

パラメーターを学習した後、モデルを使用して、
特定の学生は認められます。試験1スコアの学生向け
45の試験2と85の試験2のスコアは、入学を期待する必要があります
0.776の確率。

見つけたパラメーターの品質を評価する別の方法
学習したモデルがトレーニングセットでどの程度予測できるかを確認することです。 これで

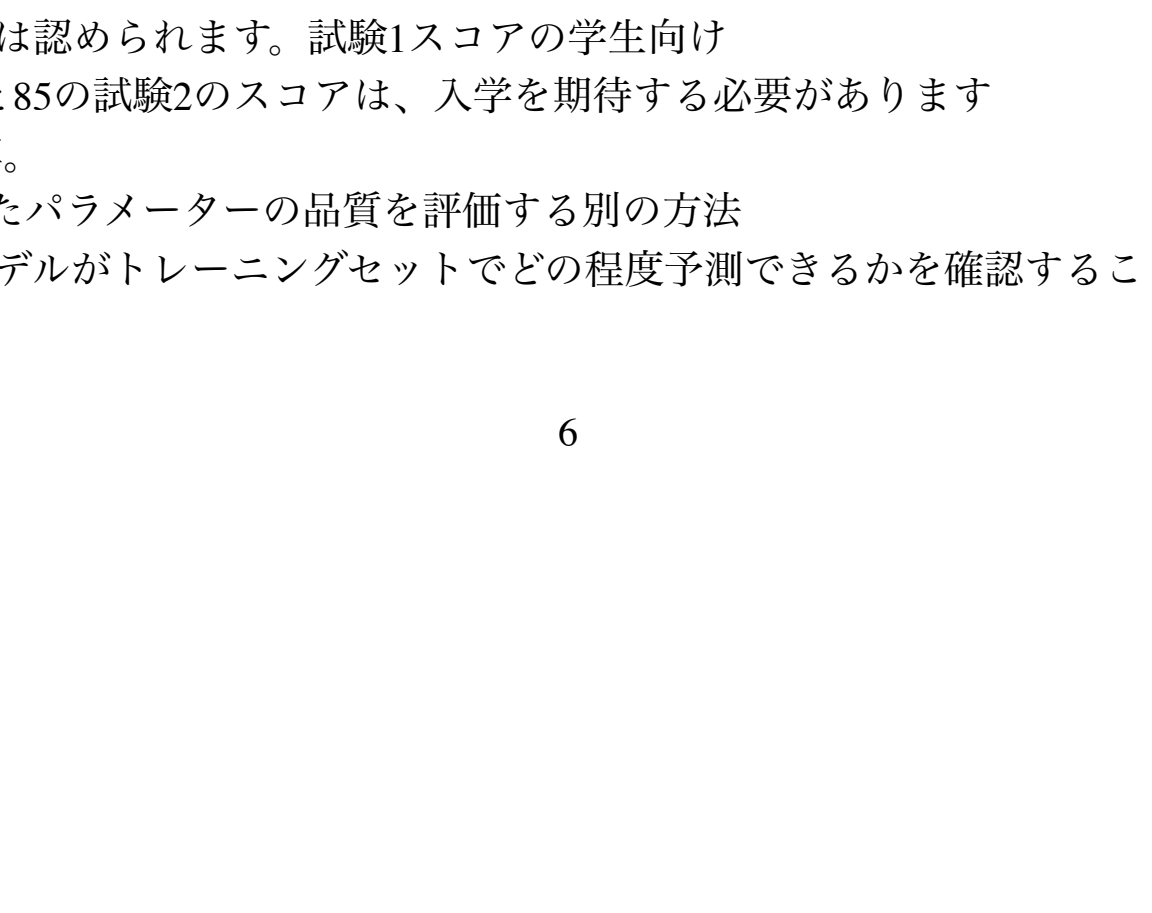

図2：決定境界を含むトレーニングデータ

一部、あなたの仕事は、predict.mのコードを完成させることです。予測関数

データセットと学習パラメータが与えられると、「1」または「0」の予測が生成されます
ベクトルθ

predict.mのコードを完了すると、ex2.mスクリプトは
を計算して、分類器のトレーニング精度を報告します。
正しい例の割合。

ここで、ソリューションを送信する必要があります。

2正規化されたロジスティック回帰

演習のこの部分では、正規化されたロジスティック回帰を実装します
製造工場のマイクロチップが品質保証に合格しているかどうかを予測する
ance（QA）。QA中に、各マイクロチップはさまざまなテストを経て、
正しく機能しています。

あなたが工場の製品管理者であり、あなたが持っているとします
2つの異なるテストでの一部のマイクロチップのテスト結果。 これら2つのテストから、
マイクロチップを受け入れるかどうかを決定したい
拒否されました。意思決定を支援するために、テスト結果のデータセットがあります
過去のマイクロチップで、ロジスティック回帰モデルを構築できます。

別のスクリプトex2 reg.mを使用して、
運動。

2.1データの視覚化

この演習の前の部分と同様に、plotDataを使用して、
図3のような図、軸は2つのテストスコア、および正
（y = 1、受け入れられた）および負の（y = 0、拒否された）の例を以下に示します
異なるマーカー。

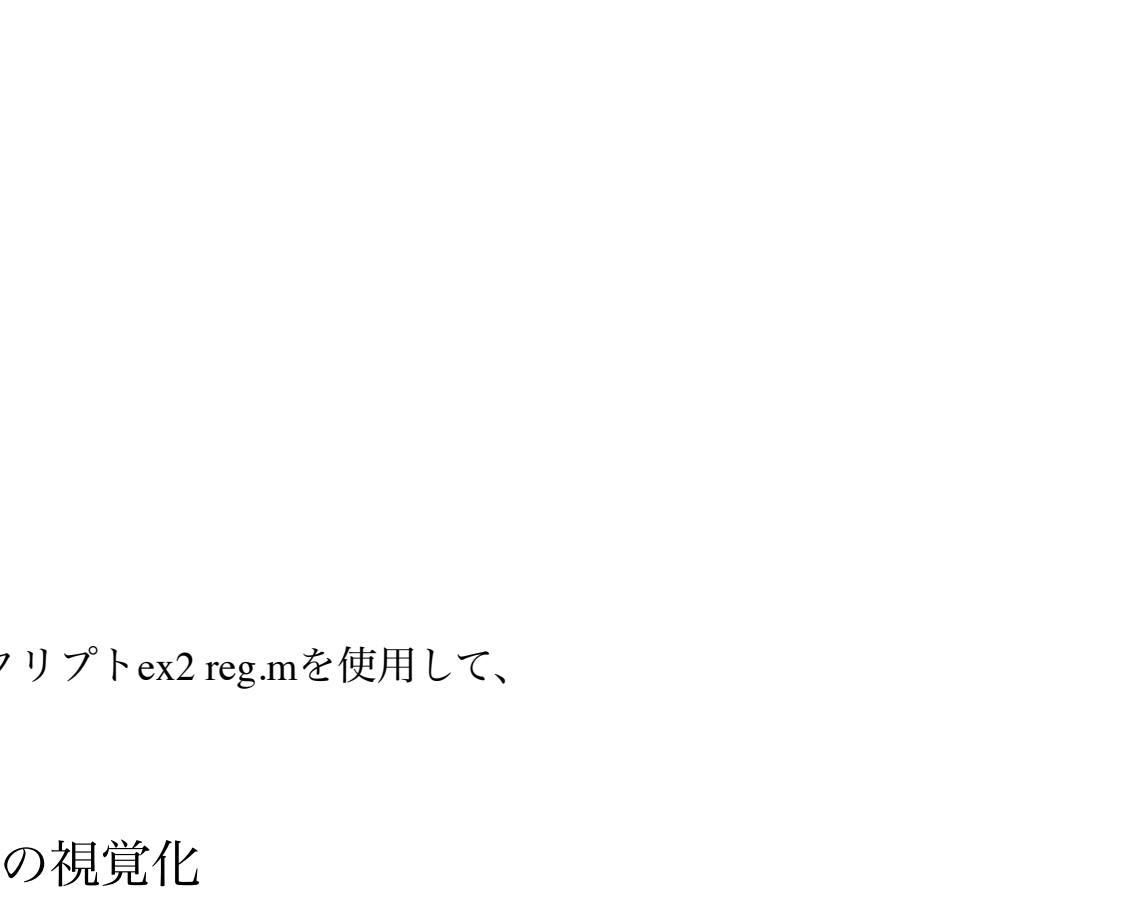

図3：トレーニングデータのプロット

図3は、データセットをポジティブと
プロットを通る直線による負の例。したがって、ストレート
ロジスティック回帰のフォワードアプリケーションは、このデータセットではうまく機能しません。
ロジスティック回帰では線形決定境界のみを見つけることができるためです。

2.2機能マッピング

データをより適切に適合させる1つの方法は、各データからより多くの機能を作成することです
ポイント。提供されている関数mapFeature.mで、機能を
x_1およびx_2の6乗までのすべての多項式項。

$$\text{mapFeature}(x) = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_1^2 \\ x_1 x_2 \\ x_2^2 \\ x_1^3 \\ \cdots \\ x_1 x_2^5 \\ x_2^6 \end{bmatrix}$$

このマッピングの結果、2つの特徴のベクトル（上のスコア
2つのQAテスト）が28次元のベクトルに変換されました。ロジスティック
この高次元の特徴ベクトルで訓練された回帰分類器は、
より複雑な決定境界で、描かれたときに非線形に見える
2次元プロット。

機能マッピングにより、より表現力豊かな分類子を構築できますが、
また、過剰適合しやすくなります。演習の次の部分では、あなたは
データに合わせて正規化されたロジスティック回帰を実装し、
正則化が過適合問題との闘いにどのように役立つかをご自身で確認してください

2.3コスト関数と勾配

次に、次のコスト関数と勾配を計算するコードを実装します
正規化されたロジスティック回帰。costFunctionReg.mのコードを完了して
コストと勾配を返します。

ロジスティック回帰の正規化されたコスト関数は

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left[-y^{(i)}\log\left(h_\theta(x^{(i)})\right) - \left(1-y^{(i)}\right)\log\left(1-h_\theta(x^{(i)})\right)\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

あなたは、パラメータθ正則ないように注意してください0を。オクターブ/マットで
LAB、インデックス付けが1から始まることを思い出してください。したがって、正則化すべきではありません
（θに対応するシータ（1）パラメータ0コードで）。勾配
コスト関数のは、j 番目の要素が次のように定義されているベクトルです。

$$\frac{\partial J(\theta)}{\partial \theta_0} = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x_j^{(私)} \qquad j = 0の場合$$

10ページ

$$\frac{\partial J(\theta)}{\partial \theta_J} = \left(\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x_j^{(私)}\right) + \frac{\lambda}{m}\theta_J \qquad j\geq 1の場合$$

完了したら、ex2 reg.mはcostFunctionReg関数を呼び出します
θの初期値（すべてゼロに初期化）を使用します。あなたはそれを見るはずです
コストは約0.693です。

ここで、ソリューションを送信する必要があります。

2.3.1 fminuncを使用した学習パラメーター

前の部分と同様に、fminuncを使用して最適なものを学習します
パラメータθ 正規化のコストと勾配を完了した場合
ロジスティック回帰（costFunctionReg.m）が正しく、ステップできるようになります
ex2 reg.mの次の部分で、fminuncを使用してパラメーターθを学習します。

2.4決定境界のプロット

この分類子によって学習されたモデルを視覚化するために、
（非線形）をプロットする関数plotDecisionBoundary.mを提供しました
正と負の例を分離する決定境界。に
plotDecisionBoundary.m、comにより非線形決定境界をプロットします。
分類器の予測を等間隔のグリッドに配置してから描画
予測がy = 0からy = 1に変化する場所の等高線図。
パラメーターθを学習した後、ex reg.mの次のステップは、
図4のような決定境界。

11ページ

2.5オプション（未評価）の演習

演習のこの部分では、異なる正則化を試すことができます。
データセットのパラメーターを使用して、正則化がオーバー
フィッティング。

λを変えると、決定境界の変化に注意してください。 小さい
λ、分類器がほぼすべてのトレーニング例を取得することを見つける必要があります
正しいが、非常に複雑な境界を描くため、データが過剰適合
（図5）。これは適切な決定境界ではありません。たとえば、予測する
x =（–0.25,1.5）の点が受け入れられ（y = 1）、これは
トレーニングセットが与えられた場合の誤った決定。

λが大きい場合、より単純な決定を示すプロットが表示されます。
まだポジティブとネガをかなりうまく分離している境界。 どうやって-
これまで、λの値が大きすぎると、適切な適合と判断が得られません。
境界はデータにあまりうまく追従しないため、データが不十分です（図
6）。

これらのオプション（未評価）のソリューションを提出する必要はありません。
演習。

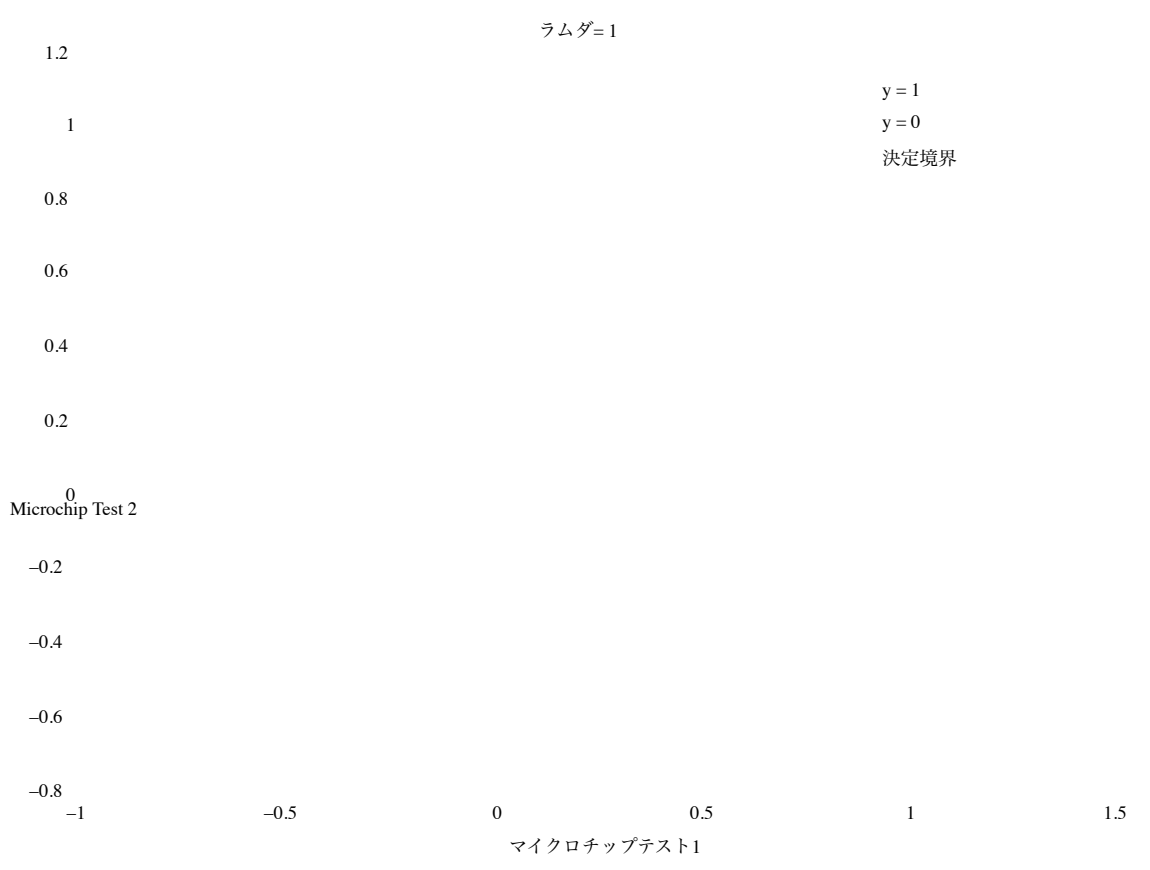

図4：決定境界を使用したトレーニングデータ（λ= 1）

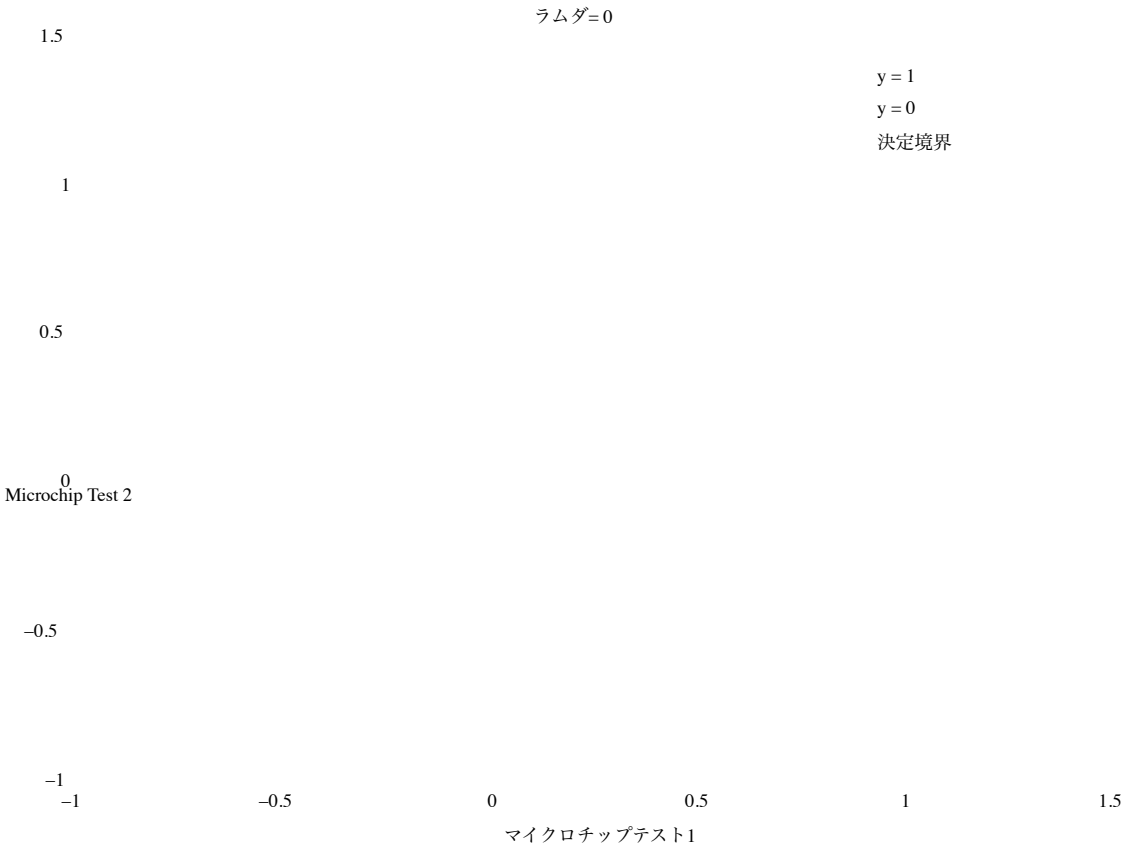

図5：正則化なし（オーバーフィット）（λ= 0）

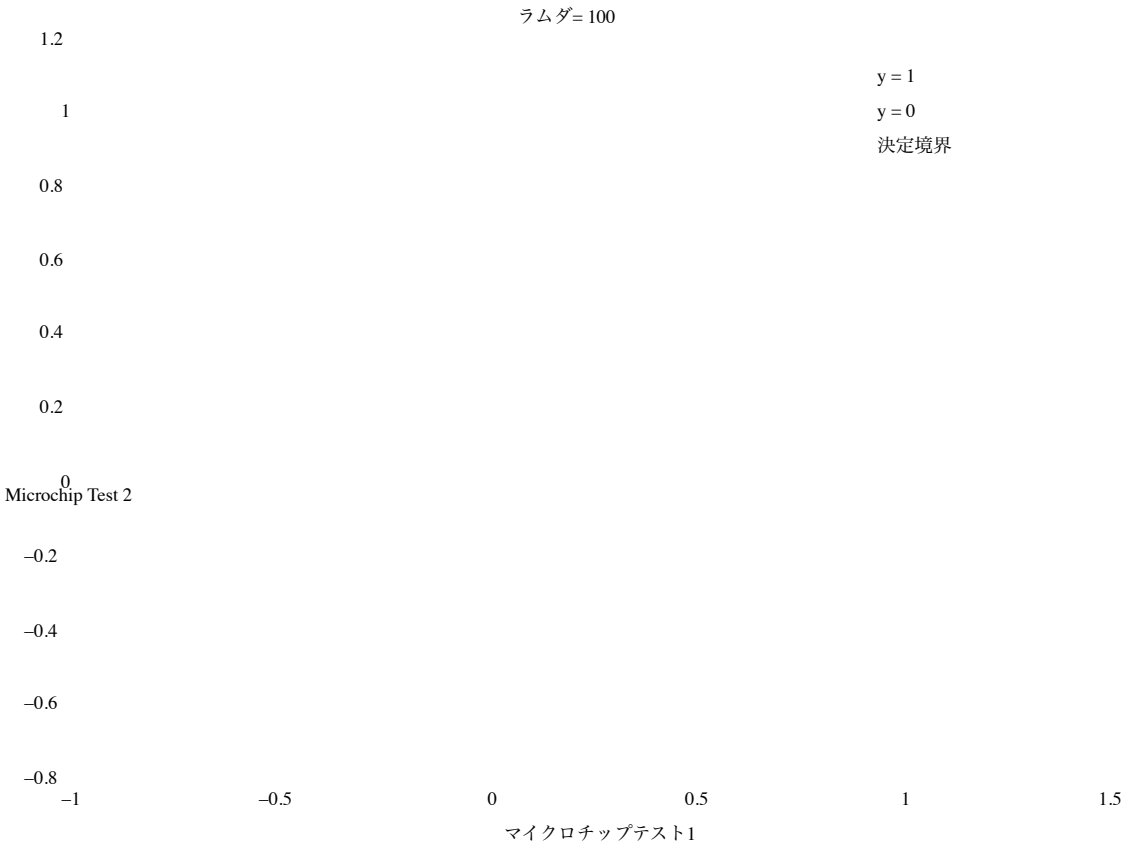

図6：正則化が多すぎる（アンダーフィット）（λ= 100）

提出と採点

課題のさまざまな部分を完了したら、必ず送信を使用してください
ソリューションを当社のサーバーに提出するための機能システム。 以下は
この演習の各部分の採点方法の内訳。

部	提出されたファイル	ポイント
シグモイド関数	sigmoid.m	5ポイント
ロジスティック回帰のコストを計算	costFunction.m	30ポイント
ロジスティック回帰の勾配	costFunction.m	30ポイント
予測機能	predict.m	5ポイント
正規化されたLRのコストを計算する	costFunctionReg.m	15ポイント
正規化されたLRの勾配	costFunctionReg.m	15ポイント
合計点		100ポイント

ソリューションを複数回送信することが許可されています。
最高のスコアのみが考慮されます。