Machine Learning
CS342

Lecture 12: Probabilistic Classification: Logistic regression

Dr. Theo Damoulas

T.Damoulas@warwick.ac.uk

Office hours: Mon & Fri 10-11am @ CS 307

Before I forget…

- Use the ML forum to ask questions and post links etc. We are a team
- Assignment 1 FAQ on ML forum please use it
- Assignment 1 going well.. right? any questions?

- Assignment 2 will be released in week 7. Should be fun
- Open Kaggle accounts from now and explore if you want:
  - https://www.kaggle.com You choose your assignment if you want

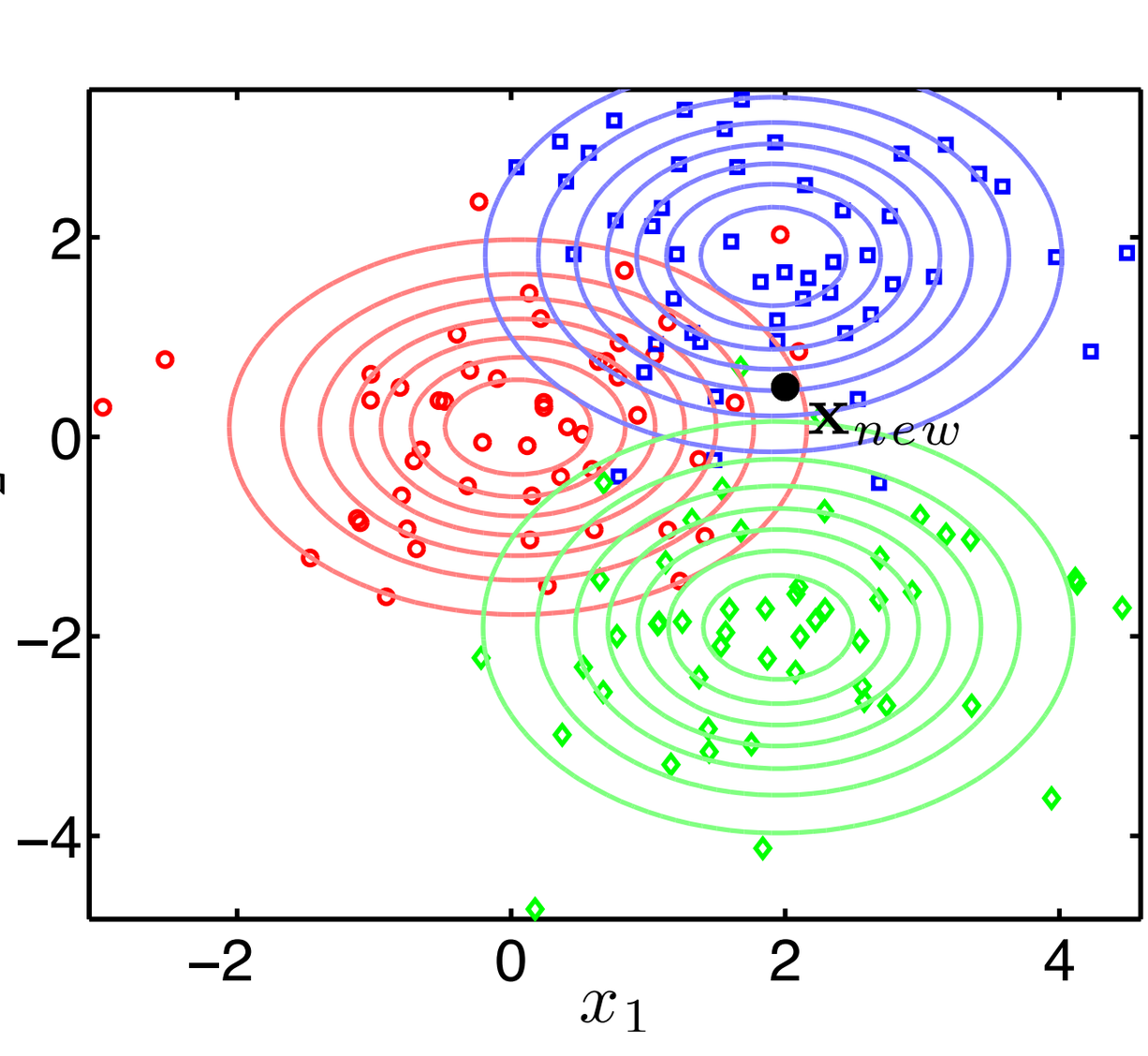
- Lab attendance is now monitored and might way in :(
- Attend all the lectures as I am carving a way through the material for you
- Not really “optional”.. if you don't attend you will struggle

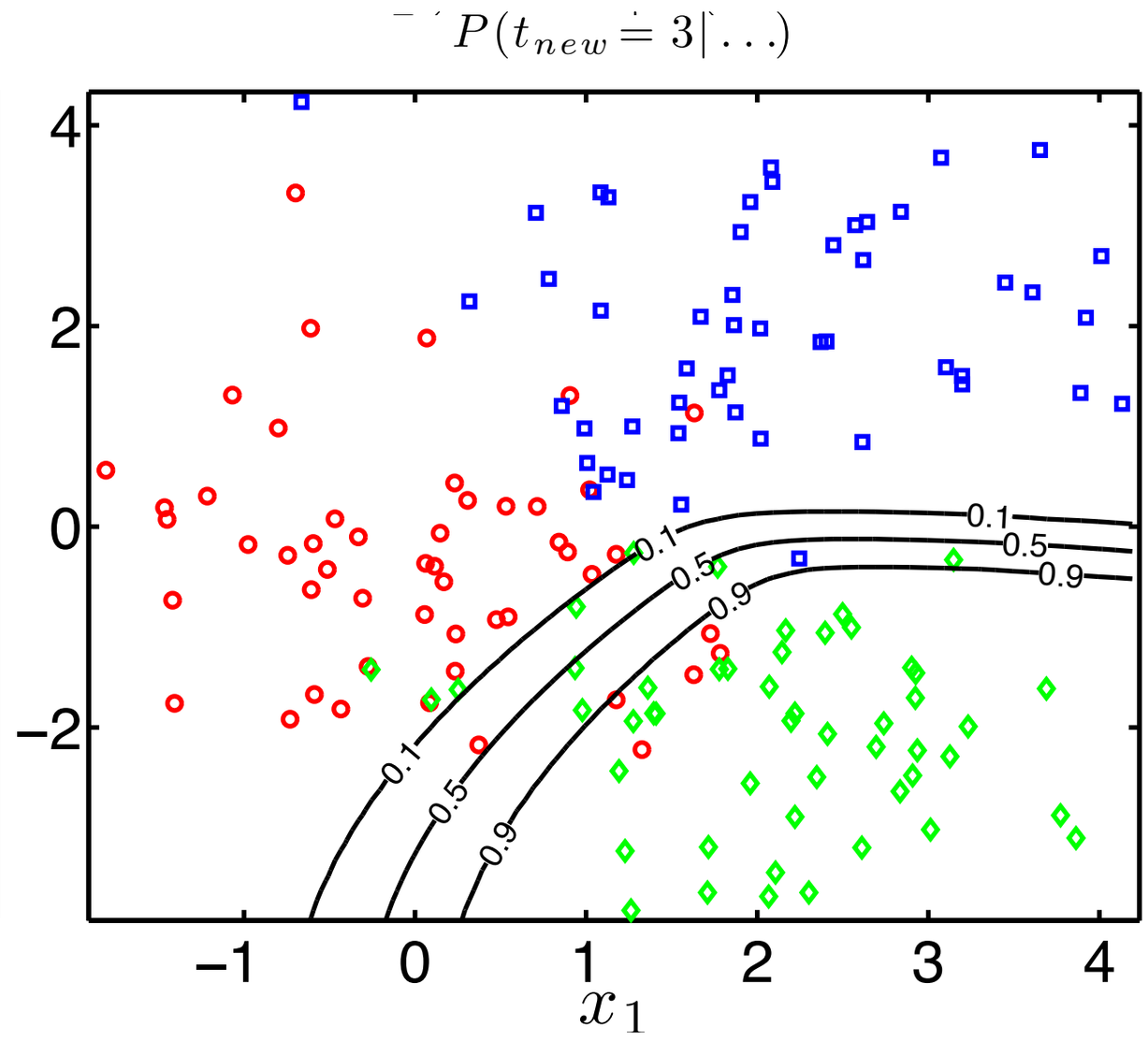
- Speed…
  - Happy to adjust but trade-off: coverage of the area - depth - speed
  - Ask questions (online and/or in person)
  - Monday we can do a slide-free Q&A/board/revision/focus in an area

Recap: Naive Bayes

Prediction phase

$$P(t^* = k|\mathbf{X}, \mathbf{t}, \mathbf{x}^*) = \frac{p(\mathbf{x}^*|t^* = k, \mathbf{X}, \mathbf{t})P(t^* = k)}{\sum_j p(\mathbf{x}^*|t^* = j, \mathbf{X}, \mathbf{t})P(t^* = j)}$$





Recap: Generative versus Discriminative models

Generative Framework

A Generative framework is one that tries to model the data generating process. In classification this means that it models the class-conditional densities of the data. You can *generate* new data from the model.

Might perform better when less evidence

Can generate fake data from it

Likelihood p(X|t,w,…)

Discriminative Framework

A Discriminative framework is one that tries to model a function that discriminates/separates the classes. This means that it models directly the decision boundary

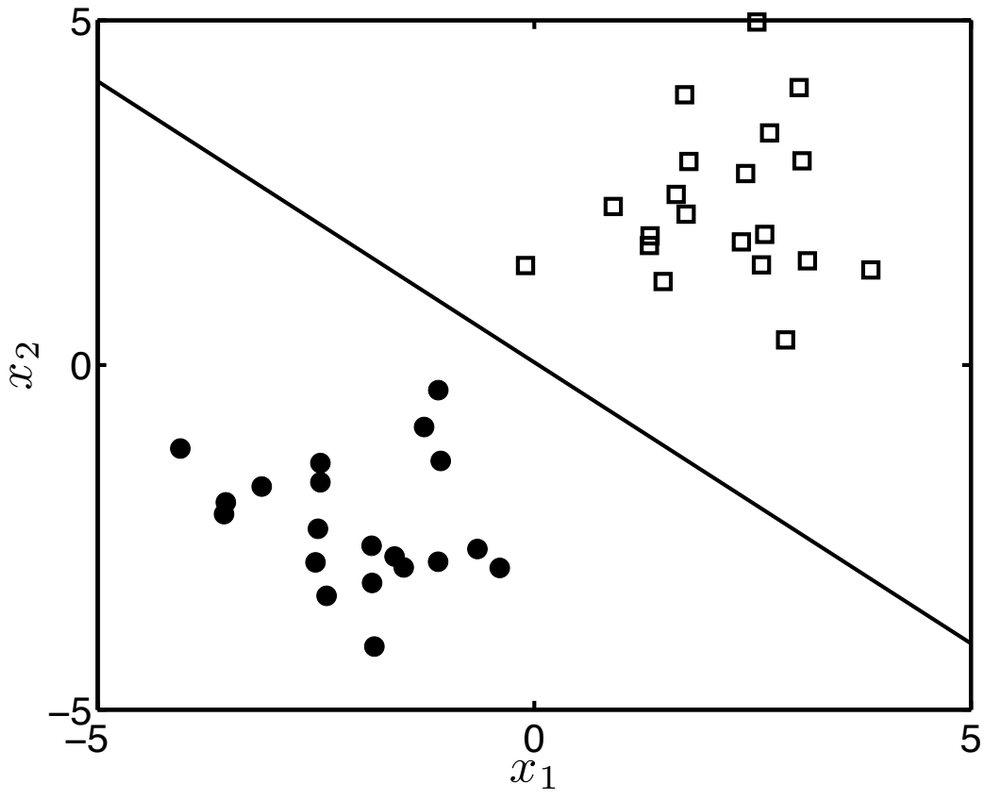
Might perform better when more evidence

Likelihood P(t|X,w,…)

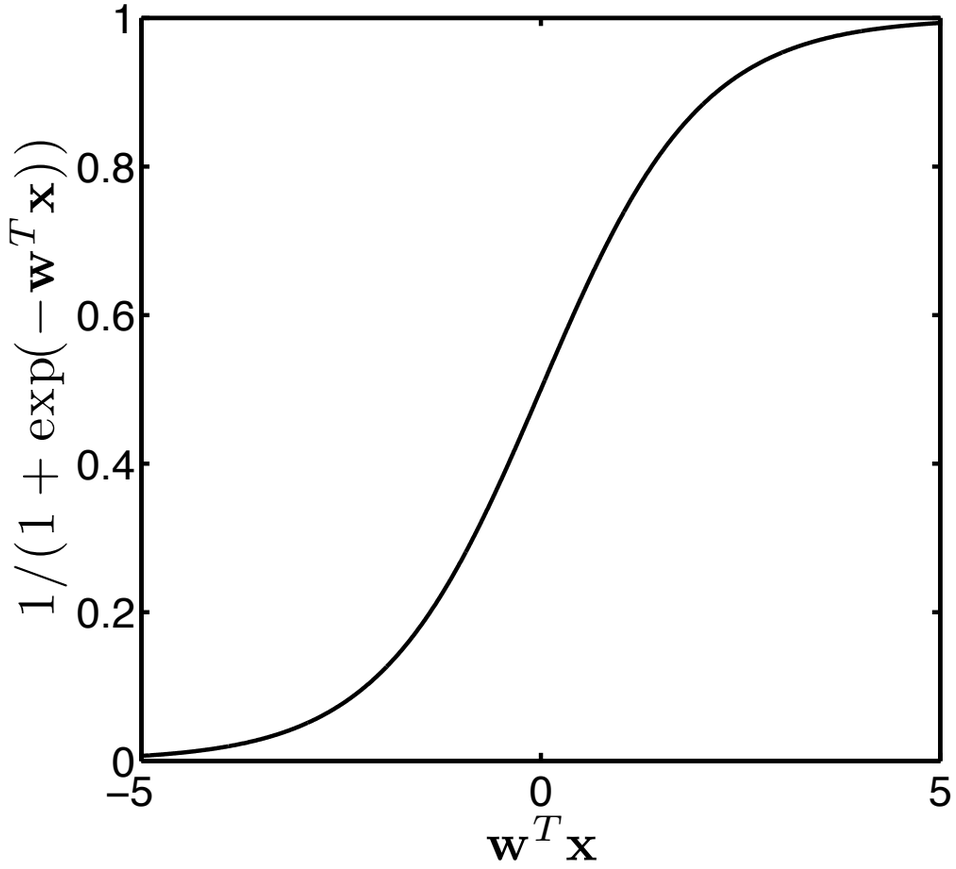
Logistic Regression: Binary classification

We want a **discriminative** function e.g. $f(\mathbf{w})=\mathbf{xw}=w_0+w_1x_1+w_2x_2$

Need to turn the continuous output of $f(\mathbf{w})$ to a probability!



Ok with Derivation?



$$\log \frac{P(t=1|\mathbf{w},\mathbf{x})}{P(t=0|\mathbf{w},\mathbf{x})} = \mathbf{xw}$$

$$P(t=1|\mathbf{w},\mathbf{x}) = \frac{1}{1+\exp(-\mathbf{xw})}$$

Logistic Regression: Likelihood

Logistic (Binary)

$$P(t=1|\mathbf{w},\mathbf{x}) = \frac{1}{1+\exp(-\mathbf{x}\mathbf{w})}$$

Artificial Neural Networks: activation function

Softmax (Multiclass extension)

$$P(t=j|\mathbf{W},\mathbf{x}) = \frac{\exp(\mathbf{x}\mathbf{w}_j)}{\sum_{k=1}^{K}\exp(\mathbf{x}\mathbf{w}_k)}$$

K decision boundaries

Can also be used for normalising data with outliers
See *sigmoidal normalisation*

Logistic Regression: Likelihood

So far I have been talking about a single **x** observation really:

$$P(t = 1|\mathbf{w}, \mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{x}\mathbf{w})}$$

We have N observations so I should really use a subscript

$$P(t_n = 1|\mathbf{w}, \mathbf{x}_n) = \frac{1}{1 + \exp(-\mathbf{x}_n\mathbf{w})}$$

and for other class $P(t_n = 0|\mathbf{w}, \mathbf{x}_n) = 1 - P(t_n = 1|\mathbf{w}, \mathbf{x}_n)$

So what is the likelihood for all my data?

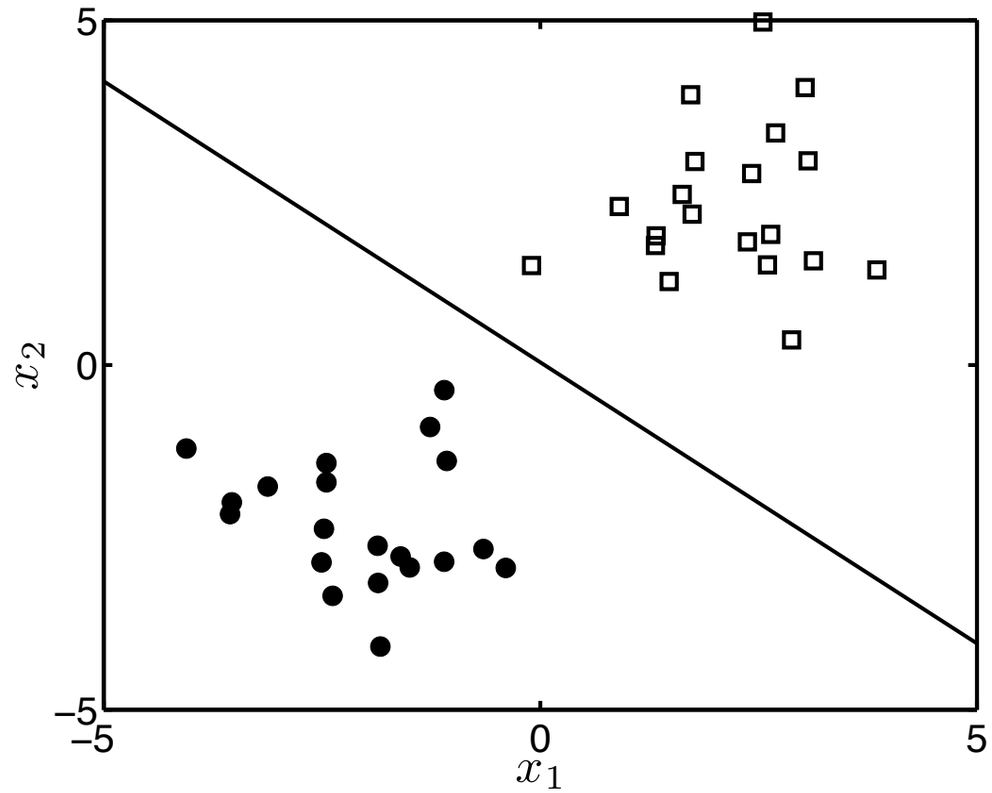
independence assumption (i.i.d) $$P(\mathbf{t}|\mathbf{X}, \mathbf{w}) = \prod_{n=1}^{N} P(t_n|\mathbf{x}_n, \mathbf{w})$$

Logistic Regression: Likelihood

$$P(\mathbf{t}|\mathbf{X},\mathbf{w}) = \prod_{n=1}^{N} P(t_n|\mathbf{x}_n,\mathbf{w})$$

Looking for an expression for this

Training/Learning phase: I know t_n (e.g. $t_n = 1$)



Assume I know **w**

$$P(t_n = 1|\mathbf{w},\mathbf{x}_n) = \frac{1}{1+\exp(-\mathbf{x}_n\mathbf{w})}$$

$$P(t_n = 0|\mathbf{w},\mathbf{x}_n) = 1 - P(t_n = 1|\mathbf{w},\mathbf{x}_n)$$

Logistic Regression: Likelihood

Lets take the nth observation from training data. I know both $\mathbf{x}_n$ and t_n

Let's say it is class 1 so $t_n = 1$

So how likely is this under my model (specific **w** and this likelihood)?

Thats what my model says

$$P(t_n = 1|\mathbf{w}, \mathbf{x}_n) = \frac{1}{1 + \exp(-\mathbf{x}_n\mathbf{w})}$$

The closer this probability is to 1 the better for this model.
What if the true class was 0?

I know that overall I should take product of these across data
But some observations will be 0 and some 1 so how can I write for all?

Bernoulli Likelihood

Here is a compact way to combine all the terms:
for an overall Likelihood across the data

$$P(\mathbf{t}|\mathbf{X},\mathbf{w}) = \prod_{n=1}^{N} P(t_n = 1|\mathbf{x}_n,\mathbf{w})^{t_n}(1 - P(t_n = 1|\mathbf{x}_n,\mathbf{w}))^{1-t_n}$$

This is known as the Bernoulli distribution

That's all you might care to know about the likelihood in logistic regression

Logistic regression

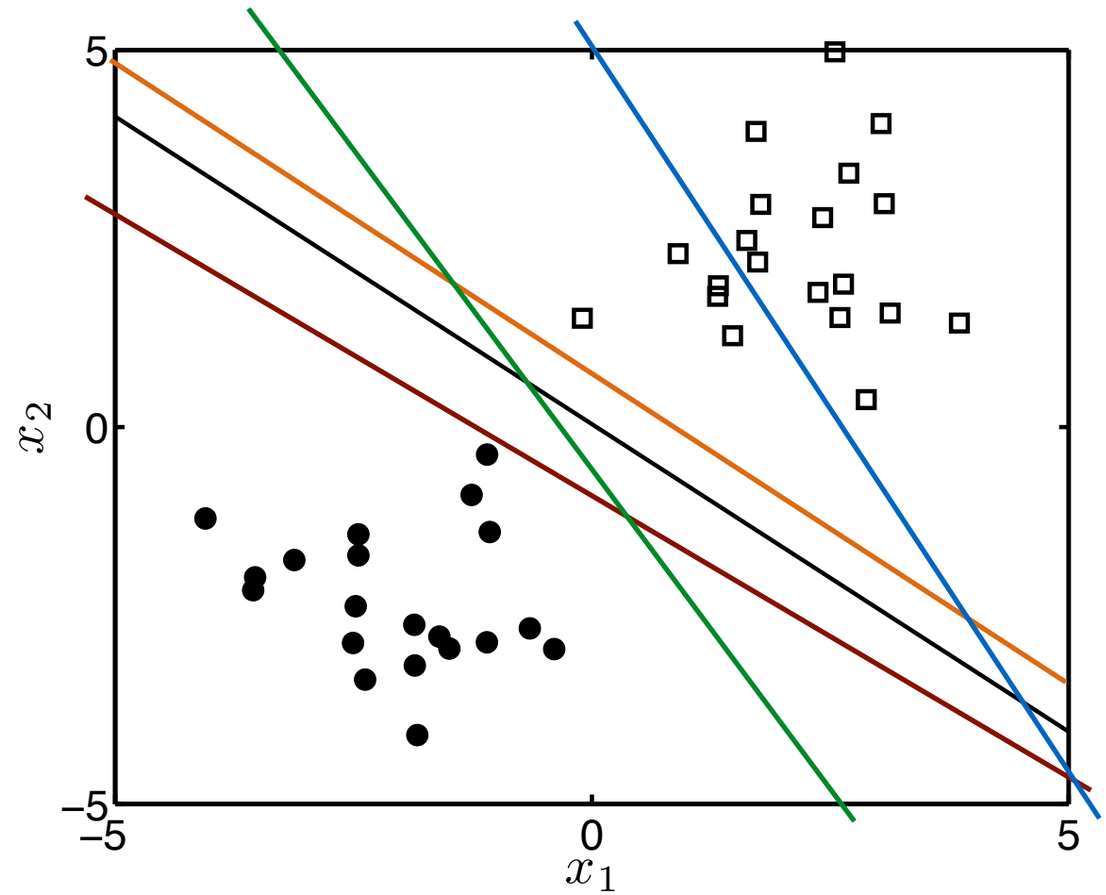
We are Bayesian this week so parameters are RVs and we use priors

We will place a prior density on **w** and attempt to get the posterior density!

Training/Learning

$$p(\mathbf{w}|\mathbf{X},\mathbf{t}) = \frac{P(\mathbf{t}|\mathbf{w},\mathbf{X})p(\mathbf{w})}{p(\mathbf{t}|\mathbf{X})}$$

Likelihood & Prior distribution(s)



Prediction

$$P(t^* = k|\mathbf{x}^*,\mathbf{X},\mathbf{t}) = \int P(t^* = k|\mathbf{x}^*,\mathbf{w})p(\mathbf{w}|\mathbf{X},\mathbf{t})$$

Logistic Regression a-la Bayes

Prior over **w**?

- In Bayesian Linear Regression we placed a Gaussian/Normal
- It was conjugate with our Gaussian likelihood
- So posterior density was also Gaussian! (closed form)
- Can we do something similar with this Likelihood?

$$p(\mathbf{w}|\mathbf{X},\mathbf{t}) = \frac{P(\mathbf{t}|\mathbf{w},\mathbf{X})p(\mathbf{w})}{p(\mathbf{t}|\mathbf{X})}$$

No :(

There is no density on **w** that we know of that is a conjugate pair

Non-conjugate Likelihood-prior pair

- Most sophisticated/complicated models will be like that
- We do not know the form of the posterior density!
- We can't compute the normalising constant (marginal likelihood)
- What can we do in general (beyond LogReg)?

More Bayesian

1. Find the most likely value of **w** (point estimator again)
2. Approximate the posterior with something easier
   - e.g. Assume it is Gaussian (Laplace approximation)
3. Sample from posterior density (Markov Chain Monte Carlo)

$$p(\mathbf{w}|\mathbf{X},\mathbf{t}) = \frac{P(\mathbf{t}|\mathbf{w},\mathbf{X})p(\mathbf{w})}{p(\mathbf{t}|\mathbf{X})}$$

We will talk a bit about 1 and 2 in the remainder

MAP point estimator

What can we compute?

$$p(\mathbf{w}|\mathbf{X},\mathbf{t}) = \frac{P(\mathbf{t}|\mathbf{w},\mathbf{X})p(\mathbf{w})}{p(\mathbf{t}|\mathbf{X})}$$

- We can compute: g(**w**) = p(**t**|**w**,**X**)p(**w**)
- First approach: find **w** that maximises the posterior density p(**w**|**X**,**t**)
- This is called the Maximum-A-Posteriori (MAP) solution
- Very similar to Maximum Likelihood but also includes a prior
- Obviously we don't have the posterior density but:

$$p(\mathbf{w}|\mathbf{X},\mathbf{t}) \propto P(\mathbf{t}|\mathbf{w},\mathbf{X})p(\mathbf{w})$$

Since marginal likelihood is constant wrt **w**

MAP point estimator

So….

$$\hat{\mathbf{w}}_{\text{MAP}} \leftarrow \underset{\mathbf{w}}{\text{argmax}}\ p(\mathbf{w}|\mathbf{X},\mathbf{t}) = \underset{\mathbf{w}}{\text{argmax}}\ P(\mathbf{t}|\mathbf{w},\mathbf{X})p(\mathbf{w})$$

So if Maximum Likelihood has analogies with minimising the Loss..
What is the addition of a prior analogous to?

Contrary to maximum likelihood on linear regression we cannot find this solution exactly with some linear algebra (no closed form update)

We resort to numerical optimisation (e.g. Newton-Raphson R&G Ch.4.):

1. Guess some initial **w**
2. Change it a bit in a way that increases the argument
3. Repeat until no further increase is possible

Those interested in optimisation: **Dr. Alina Ene's module 4th year**

MAP point estimator

Ok… but how do we predict?

$$P(t^* = 1 | \hat{\mathbf{w}}_{\text{MAP}}, \mathbf{x}^*) = \frac{1}{1 + \exp(-\mathbf{x}^* \hat{\mathbf{w}}_{\text{MAP}})}$$

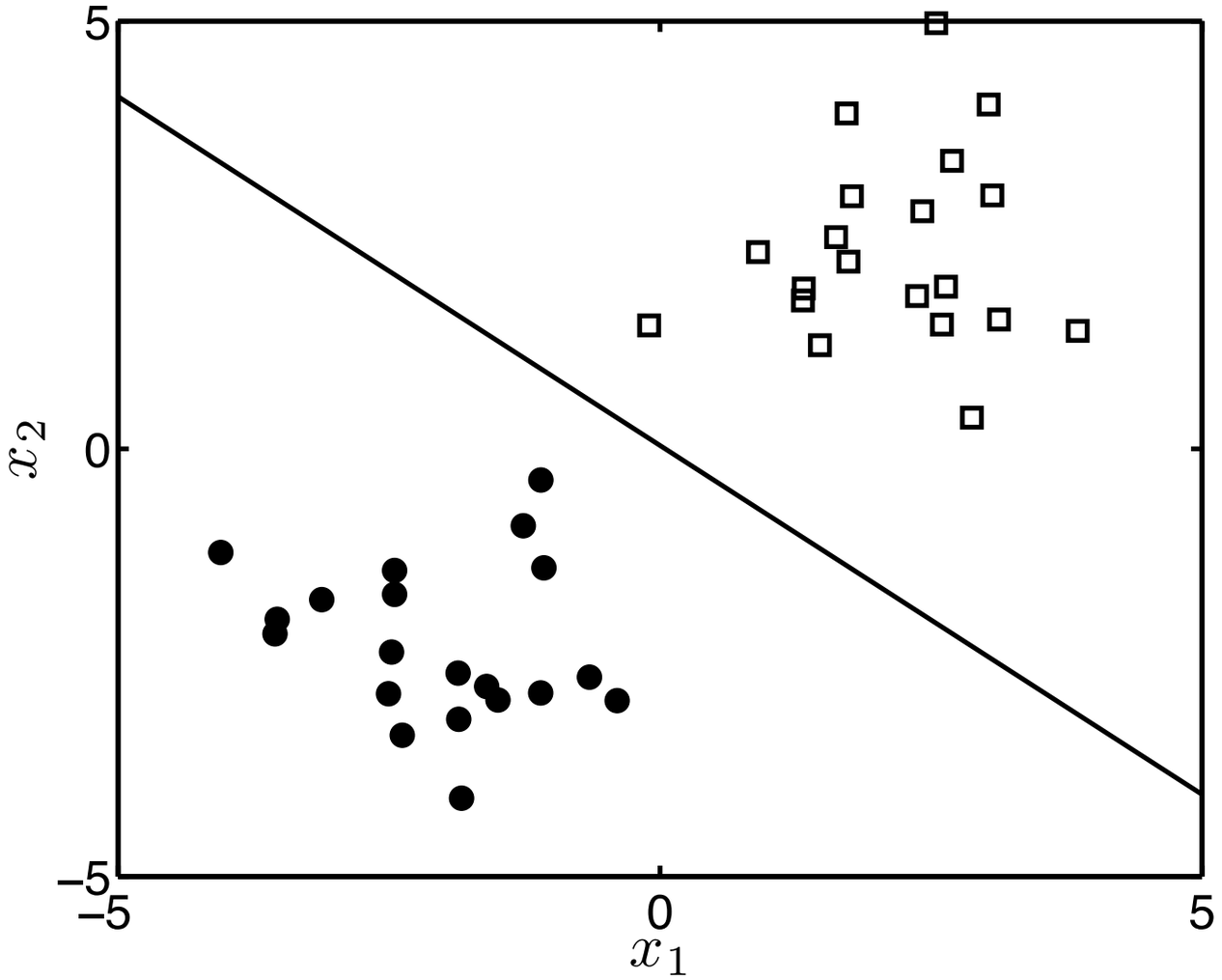
Point-estimator = single “best” parameter **w**

MAP inference very common/successful point estimators for variety of modelling frameworks and problems

You might get a probability out but they are not really probabilistic..

- **Point estimate so uncertainty is not propagated**
- Remember that uncertainty is quantified in the ***covariance of the posterior density***

MAP inference



Decision Boundary: P(t=1|..) = 0.5

Line corresponds to:

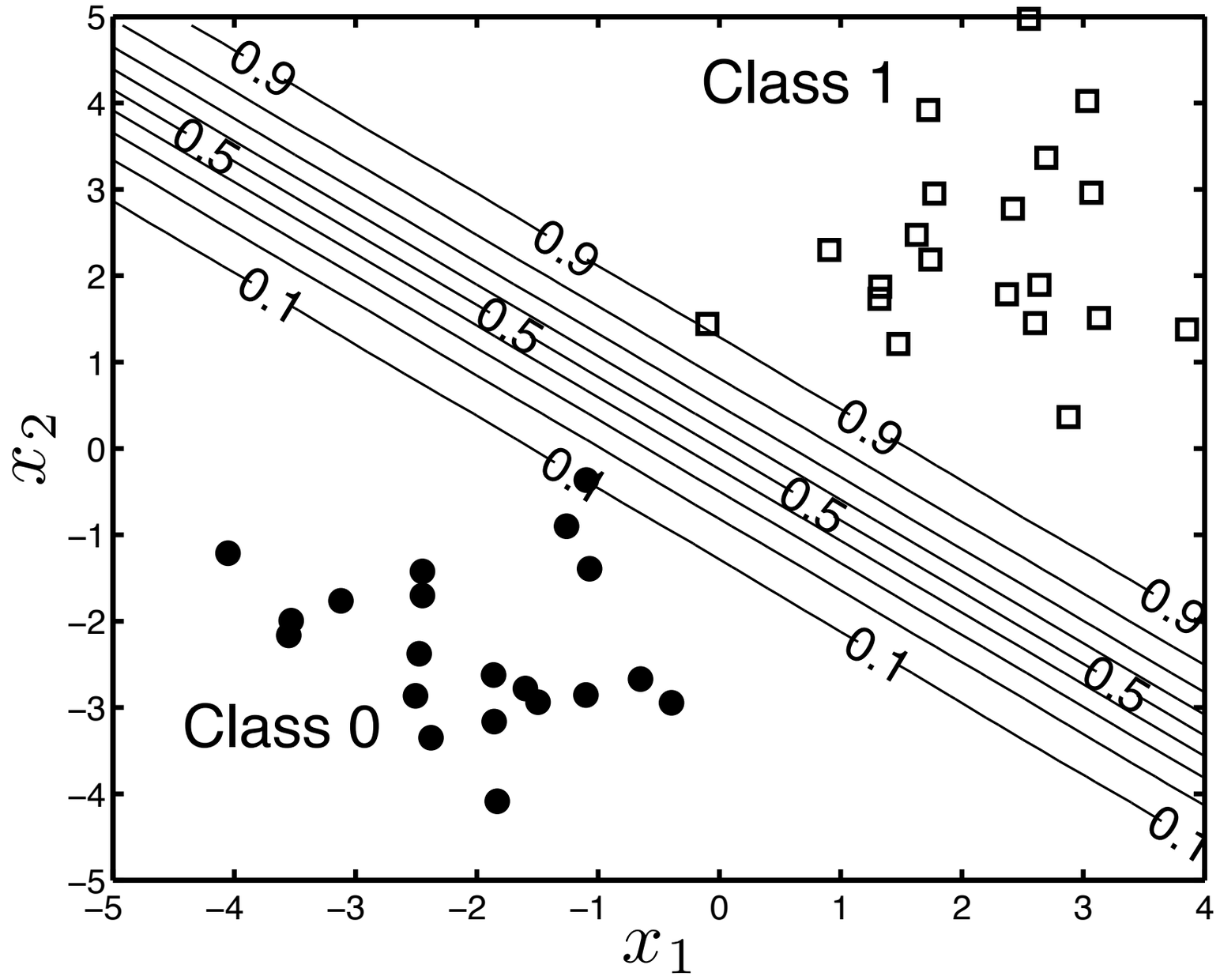
$$0.5 = \frac{1}{1 + \exp(-\mathbf{x}\hat{\mathbf{w}}_{\mathrm{MAP}})}$$

So the line is: $\mathbf{x}\hat{\mathbf{w}}_{\mathrm{MAP}} = 0$

MAP inference

Contours of

$$P(t_n = 1 | \mathbf{x}_n, \hat{\mathbf{w}}_{\text{MAP}})$$



Do you like these contours?

Lets try and be a bit more Bayesian

More Bayesian

- What can we do in general (beyond LogReg)?
  1. Find the most likely value of **w** (MAP point estimator)
  2. **Approximate the posterior with something easier**
     - e.g. Assume it is Gaussian (Laplace approximation)
  3. Sample from posterior density (Markov Chain Monte Carlo)

Approximate posterior p(**w**|**X**,**t**) with another density/function q(**w**|**X**,**t**) that is "similar":

- Has the same **mode** (highest point)
- Maybe its similar in **shape**?
- Mathematical convenience: Something we can easily work with

Laplace approximation

Not state-of-the-art but easy to understand and work with

Laplace approximation: Approximate the posterior density p(**w**|**X**,**t**) with… a Gaussian distribution q(**w**|**X**,**t**)!

$$q(\mathbf{w}|\mathbf{X},\mathbf{t}) = \mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})$$

Just to get “an idea”
Not exam material
Ch. 4 R&G book

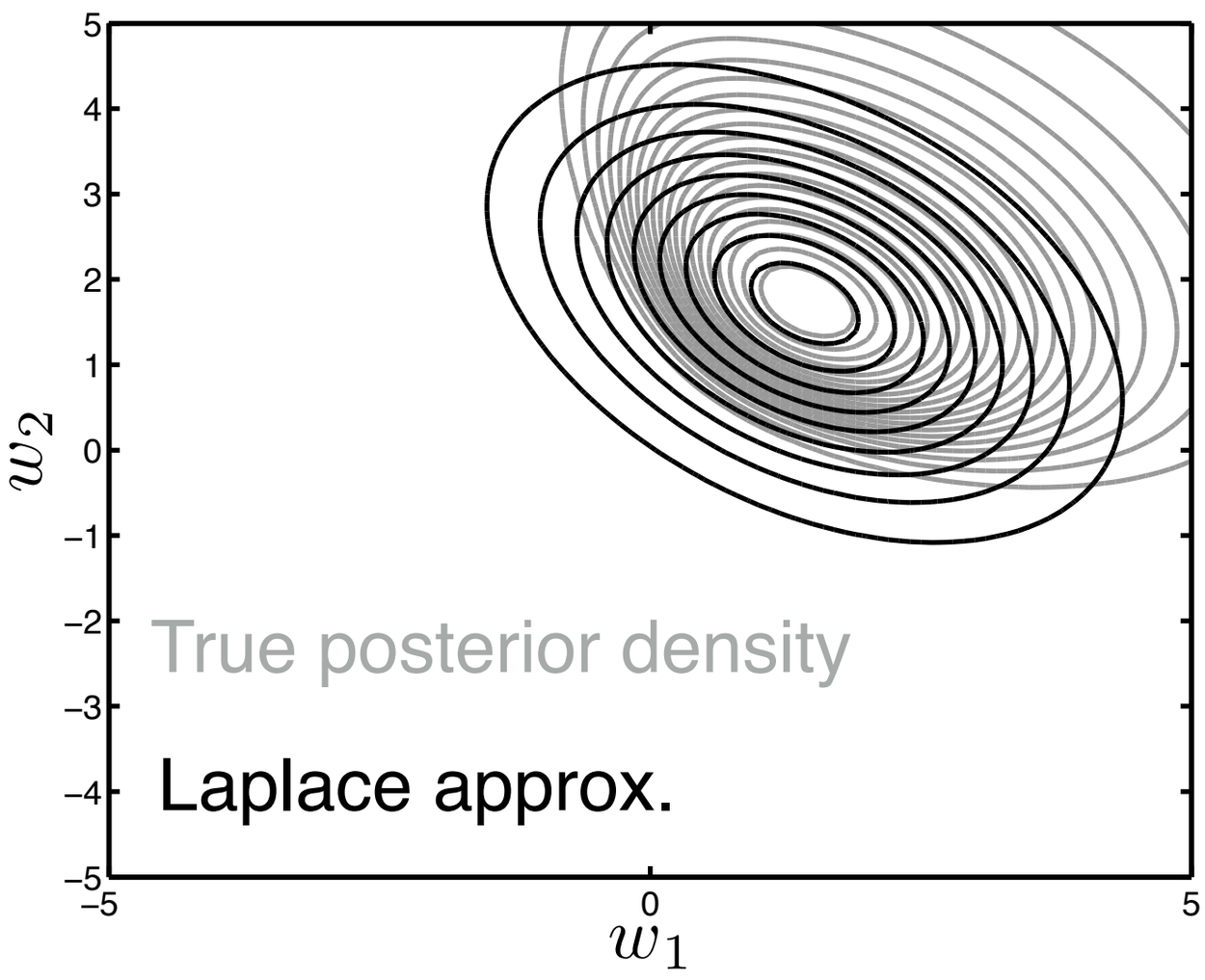
where:

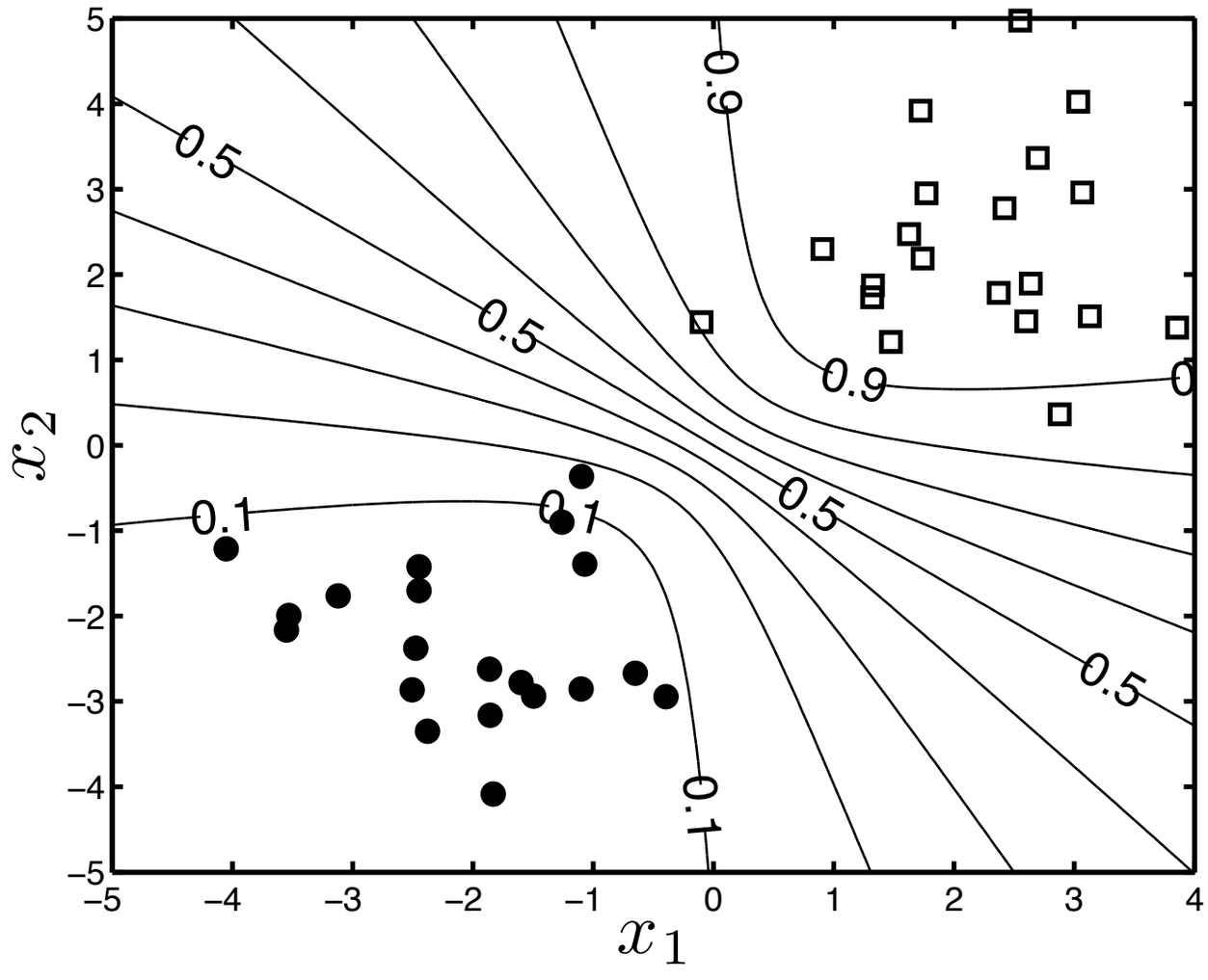
$\boldsymbol{\mu} = \hat{\mathbf{w}}_{\mathrm{MAP}}$ Set the mean to the mode

$$\boldsymbol{\Sigma}^{-1} = -\left.\frac{\partial^2 \log P(\mathbf{t}|\mathbf{w},\mathbf{X})p(\mathbf{w})}{\partial \mathbf{w}^{\mathrm{T}} \partial \mathbf{w}}\right|_{\hat{\mathbf{w}}_{\mathrm{MAP}}}$$

Set the covariance based on Taylor expansion “curvature info around mode”

Laplace approximation





Laplace approximation to the true normalised posterior.
Contours look better? because we average over multiple solutions instead of a single point estimate (like MAP)

Fully Bayesian?

More Bayesian

- What can we do in general (beyond LogReg)?
  1. Find the most likely value of **w** (MAP point estimator)
  2. Approximate the posterior with something easier
     - e.g. Assume it is Gaussian (Laplace approximation)
  3. **Sample from posterior density (Markov Chain Monte Ca**

We can draw samples from the unknown density and approximate it

Not covered as well (unless you insist!)

High-level idea (not that accurate really): Approximating the density of a cloud by only watching the rain-drops on the pavement

The Stats dept at Warwick excels in this area if you are interested