Logistic regression

Morteza H. Chehreghani
morteza.chehreghani@chalmers.se

Chalmers University of Technology

April 27, 2020

Reference

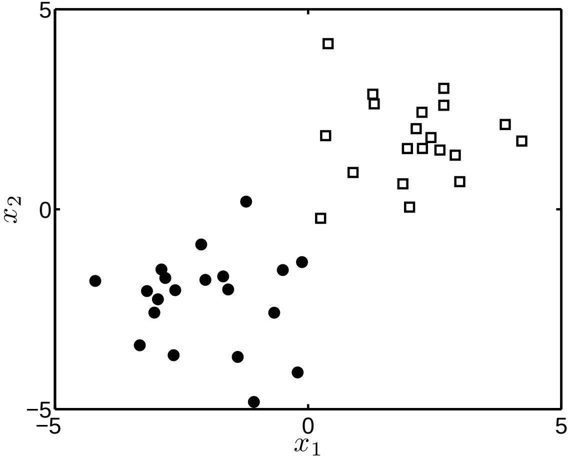
The content and the slides are adapted from

S. Rogers and M. Girolami, A First Course in Machine Learning (FCML), 2nd edition, Chapman & Hall/CRC 2016, ISBN: 9781498738484

Classification syllabus

- 4 classification algorithms.
- Of which:
  - 2 are probabilistic.
    - Bayes classifier.
    - **Logistic regression.**
  - 2 are non-probabilistic.
    - K-nearest neighbours.
    - Support Vector Machines.
- There are many others!

Some data



Logistic regression

- In the Bayes classifier, we built a model of each class and then used Bayes rule:

$$P(t_{\text{new}} = k|\mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t}) = \frac{p(\mathbf{x}_{\text{new}}|t_{\text{new}} = k, \mathbf{X}, \mathbf{t})P(t_{\text{new}} = k)}{\sum_j p(\mathbf{x}_{\text{new}}|t_{\text{new}} = j, \mathbf{X}, \mathbf{t})P(t_{\text{new}} = j)}$$

Logistic regression

- In the Bayes classifier, we built a model of each class and then used Bayes rule:

$$P(t_{\text{new}} = k | \mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t}) = \frac{p(\mathbf{x}_{\text{new}} | t_{\text{new}} = k, \mathbf{X}, \mathbf{t}) P(t_{\text{new}} = k)}{\sum_j p(\mathbf{x}_{\text{new}} | t_{\text{new}} = j, \mathbf{X}, \mathbf{t}) P(t_{\text{new}} = j)}$$

- Alternative is to directly model $P(t_{\text{new}} = k | \mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t}) = f(\mathbf{x}_{\text{new}}; \mathbf{w})$ with some parameters $\mathbf{w}$.

Logistic regression

- In the Bayes classifier, we built a model of each class and then used Bayes rule:

$$P(t_{\text{new}} = k|\mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t}) = \frac{p(\mathbf{x}_{\text{new}}|t_{\text{new}} = k, \mathbf{X}, \mathbf{t})P(t_{\text{new}} = k)}{\sum_j p(\mathbf{x}_{\text{new}}|t_{\text{new}} = j, \mathbf{X}, \mathbf{t})P(t_{\text{new}} = j)}$$

- Alternative is to directly model $P(t_{\text{new}} = k|\mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t}) = f(\mathbf{x}_{\text{new}}; \mathbf{w})$ with some parameters $\mathbf{w}$.

- We've seen $f(\mathbf{x}_{\text{new}}; \mathbf{w}) = \mathbf{w}^{\mathsf{T}}\mathbf{x}_{\text{new}}$ before – can we use it here?
  - No – *output is unbounded and so can't be a probability.*

Logistic regression

- In the Bayes classifier, we built a model of each class and then used Bayes rule:

$$P(t_{\text{new}} = k|\mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t}) = \frac{p(\mathbf{x}_{\text{new}}|t_{\text{new}} = k, \mathbf{X}, \mathbf{t})P(t_{\text{new}} = k)}{\sum_j p(\mathbf{x}_{\text{new}}|t_{\text{new}} = j, \mathbf{X}, \mathbf{t})P(t_{\text{new}} = j)}$$

- Alternative is to directly model $P(t_{\text{new}} = k|\mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t}) = f(\mathbf{x}_{\text{new}}; \mathbf{w})$ with some parameters $\mathbf{w}$.

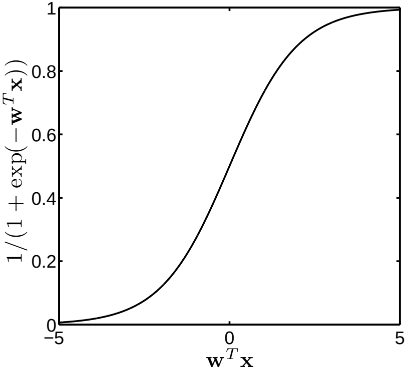
- We've seen $f(\mathbf{x}_{\text{new}}; \mathbf{w}) = \mathbf{w}^\mathsf{T}\mathbf{x}_{\text{new}}$ before – can we use it here?
  - No – *output is unbounded and so can't be a probability.*

- But, can use $P(t_{\text{new}} = k|\mathbf{x}_{\text{new}}, \mathbf{w}) = h(f(\mathbf{x}_{\text{new}}; \mathbf{w}))$ where $h(\cdot)$ *squashes* $f(\mathbf{x}_{\text{new}}; \mathbf{w})$ to lie between 0 and 1 – a probability.

$h(\cdot)$

- For logistic regression (binary), we use the sigmoid function:

$$P(t_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{w}) = h(\mathbf{w}^{\mathsf{T}}\mathbf{x}_{\text{new}}) = \frac{1}{1 + \exp(-\mathbf{w}^{\mathsf{T}}\mathbf{x}_{\text{new}})}$$

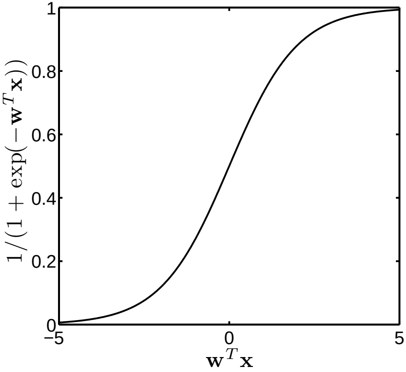


$h(\cdot)$

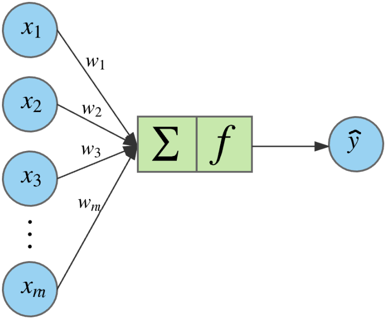
- For logistic regression (binary), we use the sigmoid function:

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

$$P(t = 0|\mathbf{x}, \mathbf{w}) = 1 - h(\mathbf{w}^\mathsf{T}\mathbf{x}) = \frac{\exp(-\mathbf{w}^\mathsf{T}\mathbf{x})}{1 + \exp(-\mathbf{w}^\mathsf{T}\mathbf{x})}$$



Perceptron



Likelihood

We consider likelihood on train data to infer the parameters $\mathbf{w}$.

$$
\begin{aligned}
p(\mathbf{t}|\mathbf{X}, \mathbf{w}) &= \prod_{n=1}^{N} p(t_n|\mathbf{x}_n, \mathbf{w}) \\
&= \prod_{t_n=1} p(t_n|\mathbf{x}_n, \mathbf{w}) \prod_{t_n=0} p(t_n|\mathbf{x}_n, \mathbf{w}) \\
&= \prod_{t_n=1} h(\mathbf{w}^\mathsf{T}\mathbf{x}_n) \prod_{t_n=0} (1 - h(\mathbf{w}^\mathsf{T}\mathbf{x}_n))
\end{aligned}
$$

Cross Entropy

The negative log-likelihood is written by

$$
\begin{aligned}
\mathbf{J}(\mathbf{w}) &= -\sum_{t_n=1} \log h(\mathbf{w}^\top \mathbf{x}_n) - \sum_{t_n=0} \log(1 - h(\mathbf{w}^\top \mathbf{x}_n)) \\
&= -\sum_{n=1}^{N} t_n \log h(\mathbf{w}^\top \mathbf{x}_n) + (1 - t_n) \log(1 - h(\mathbf{w}^\top \mathbf{x}_n))
\end{aligned}
$$

Minimization of Cross Entropy

We minimize Cross Entropy to infer the model parameters w_j.

$$\frac{\partial \mathbf{J}}{\partial w_j} = -\sum_{n=1}^{N}[t_n - h(\mathbf{w}^T \mathbf{x}_n)]\mathbf{x}_{n,j}$$

We may use Gradient Descent for this purpose:

$$w_j \leftarrow w_j - \eta \frac{\partial \mathbf{J}}{\partial w_j}$$

In logistic regression, Cross Entropy is *convex*.

Multiclass Classification

Data in K classes

$$(\mathbf{x}_1, t_1), \cdots (\mathbf{x}_N, t_N),$$

where each $t_n \in \{1 \cdots K\}$

One hot representation

Each label $t_n \in \{1 \cdots K\}$ can be represented as a 0/1 K-vector, with

$$t_{n,k} = \begin{cases} 1, \text{if } t_n = k \\ 0, \text{otherwise} \end{cases}$$

Softmax Regression

$$P(t_n = k | \mathbf{x}_n, \{\mathbf{w}_\ell\}) = \frac{\exp(-\mathbf{w}_k \mathbf{x}_n)}{\sum_{\ell=1}^{K} \exp(-\mathbf{w}_\ell \mathbf{x}_n)}$$

That is, we have K parameter vectors $\mathbf{w}_1, \cdots, \mathbf{w}_K$ with $\mathbf{w}_k$ used to compute the probability $P(t_{n,k} = 1)$.

Cross Entropy: Multiple Classes

The Cross-Entropy loss is written by

$$\mathbf{J} = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{n,k} \log \frac{\exp(-\mathbf{w}_k \mathbf{x}_n)}{\sum_{\ell=1}^{K} \exp(-\mathbf{w}_\ell \mathbf{x}_n)}$$

Gradient: Multiple Classes

The gradient can be used in Gradient-Descent optimization, or for other purposes.

$$\frac{\partial \mathbf{J}}{\partial w_{k,j}} = -\sum_{n=1}^{N} \left[t_{n,k} - \frac{\exp(-\mathbf{w}_k \mathbf{x}_n)}{\sum_{\ell=1}^{K} \exp(-\mathbf{w}_\ell \mathbf{x}_n)} \right] \mathbf{x}_{n,j}$$

Bayesian logistic regression (back to binary setting)

- Recall the Bayesian ideas from few lectures ago....
- In theory, if we place a *prior* on $\mathbf{w}$ and define a *likelihood* we can obtain a *posterior*:

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

Bayesian logistic regression (back to binary setting)

- Recall the Bayesian ideas from few lectures ago....
- In theory, if we place a *prior* on $\mathbf{w}$ and define a *likelihood* we can obtain a *posterior*:

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

- And we can make predictions by taking expectations (averaging over $\mathbf{w}$):

$$P(t_{\text{new}} = 1|\mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t}) = \mathbf{E}_{p(\mathbf{w}|\mathbf{X},\mathbf{t})}\left\{P(t_{\text{new}} = 1|\mathbf{x}_{\text{new}}, \mathbf{w})\right\}$$

- Sounds good so far....

Defining a prior

- Choose a Gaussian prior:

$$p(\mathbf{w}) = \prod_{d=1}^{D} \mathcal{N}(0, \sigma^2).$$

  - For simplicity, here we assume w_0 is zero.
  - The prior has the parameter σ^2.
  - Prior choice is *always* important from a data analysis point of view.
  - Previously, it was also important 'for the maths'.
  - This isn't the case today – could choose any prior – no prior makes the maths easier!

Defining a likelihood

- First assume independence:

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

Defining a likelihood

- First assume independence:

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

- We have already defined this – it's our squashing function! If $t_n = 1$:

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

- and if $t_n = 0$:

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

Posterior

$$p(\mathbf{w}|\mathbf{X},\mathbf{t},\sigma^2) = \frac{p(\mathbf{t}|\mathbf{X},\mathbf{w})p(\mathbf{w}|\sigma^2)}{p(\mathbf{t}|\mathbf{X},\sigma^2)}$$

- Now things start going wrong.
- We can't compute $p(\mathbf{w}|\mathbf{X},\mathbf{t},\sigma^2)$ analytically.
  - Prior is not conjugate to likelihood. No prior is!
  - This means we don't know the *form* of $p(\mathbf{w}|\mathbf{X},\mathbf{t},\sigma^2)$
  - And we can't compute the marginal likelihood:

$$p(\mathbf{t}|\mathbf{X},\sigma^2) = \int p(\mathbf{t}|\mathbf{X},\mathbf{w})p(\mathbf{w}|\sigma^2)\ d\mathbf{w}$$

What can we compute?

$$p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2) = \frac{p(\mathbf{t}|\mathbf{X}, \mathbf{w})p(\mathbf{w}|\sigma^2)}{p(\mathbf{t}|\mathbf{X}, \sigma^2)}$$

- We may not be able to compute $p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$
  - Define $g(\mathbf{w}; \mathbf{X}, \mathbf{t}, \sigma^2) = p(\mathbf{t}|\mathbf{X}, \mathbf{w})p(\mathbf{w}|\sigma^2)$

What can we compute?

$$p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2) = \frac{p(\mathbf{t}|\mathbf{X}, \mathbf{w})p(\mathbf{w}|\sigma^2)}{p(\mathbf{t}|\mathbf{X}, \sigma^2)}$$

- We may not be able to compute $p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$
  - Define $g(\mathbf{w}; \mathbf{X}, \mathbf{t}, \sigma^2) = p(\mathbf{t}|\mathbf{X}, \mathbf{w})p(\mathbf{w}|\sigma^2)$
- Armed with this, we have three options:
  - Find the most likely value of **w** – a point estimate.

What can we compute?

$$p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2) = \frac{p(\mathbf{t}|\mathbf{X}, \mathbf{w})p(\mathbf{w}|\sigma^2)}{p(\mathbf{t}|\mathbf{X}, \sigma^2)}$$

- We may not be able to compute $p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$
  - Define $g(\mathbf{w}; \mathbf{X}, \mathbf{t}, \sigma^2) = p(\mathbf{t}|\mathbf{X}, \mathbf{w})p(\mathbf{w}|\sigma^2)$
- Armed with this, we have three options:
  - Find the most likely value of $\mathbf{w}$ – a point estimate.
  - Approximate $p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$ with something easier.

What can we compute?

$$p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2) = \frac{p(\mathbf{t}|\mathbf{X}, \mathbf{w})p(\mathbf{w}|\sigma^2)}{p(\mathbf{t}|\mathbf{X}, \sigma^2)}$$

- We may not be able to compute $p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$
  - Define $g(\mathbf{w}; \mathbf{X}, \mathbf{t}, \sigma^2) = p(\mathbf{t}|\mathbf{X}, \mathbf{w})p(\mathbf{w}|\sigma^2)$
- Armed with this, we have three options:
  - Find the most likely value of $\mathbf{w}$ – a point estimate.
  - Approximate $p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$ with something easier.
  - Sample from $p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$.

What can we compute?

$$p(\mathbf{w}|\mathbf{X},\mathbf{t},\sigma^2) = \frac{p(\mathbf{t}|\mathbf{X},\mathbf{w})p(\mathbf{w}|\sigma^2)}{p(\mathbf{t}|\mathbf{X},\sigma^2)}$$

- We may not be able to compute $p(\mathbf{w}|\mathbf{X},\mathbf{t},\sigma^2)$
  - Define $g(\mathbf{w};\mathbf{X},\mathbf{t},\sigma^2) = p(\mathbf{t}|\mathbf{X},\mathbf{w})p(\mathbf{w}|\sigma^2)$
- Armed with this, we have three options:
  - Find the most likely value of $\mathbf{w}$ – a point estimate.
  - Approximate $p(\mathbf{w}|\mathbf{X},\mathbf{t},\sigma^2)$ with something easier.
  - Sample from $p(\mathbf{w}|\mathbf{X},\mathbf{t},\sigma^2)$.
- We'll cover *examples* of each of these in turn....
- These examples aren't the only ways of approximating/sampling.
- They are also general techniques not unique to logistic regression.

MAP estimate

- Our first method is to find the value of $\mathbf{w}$ that maximises $p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$ (call it $\widehat{\mathbf{w}}$).
  - $g(\mathbf{w}; \mathbf{X}, \mathbf{t}, \sigma^2) \propto p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$
  - $\widehat{\mathbf{w}}$ therefore also maximises $g(\mathbf{w}; \mathbf{X}, \mathbf{t}, \sigma^2)$.
- Very similar to maximum likelihood but additional effect of prior.
- Known as MAP (maximum a posteriori) solution.

MAP estimate

- Our first method is to find the value of $\mathbf{w}$ that maximises $p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$ (call it $\widehat{\mathbf{w}}$).
  - $g(\mathbf{w}; \mathbf{X}, \mathbf{t}, \sigma^2) \propto p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$
  - $\widehat{\mathbf{w}}$ therefore also maximises $g(\mathbf{w}; \mathbf{X}, \mathbf{t}, \sigma^2)$.
- Very similar to maximum likelihood but additional effect of prior.
- Known as MAP (maximum a posteriori) solution.

- Once we have $\widehat{\mathbf{w}}$, make predictions with:

$$P(t_{\text{new}} = 1|\mathbf{x}_{\text{new}}, \widehat{\mathbf{w}}) = \frac{1}{1 + \exp(-\widehat{\mathbf{w}}^{\mathsf{T}}\mathbf{x}_{\text{new}})}$$

MAP

- When we met maximum likelihood, we could find $\widehat{\mathbf{w}}$ exactly with some algebra (in logistic regression, Cross Entropy is *convex*.).
- Can't do that here (can't solve $\frac{\partial g(\mathbf{w};\mathbf{X},\mathbf{t},\sigma^2)}{\partial \mathbf{w}} = \mathbf{0}$)

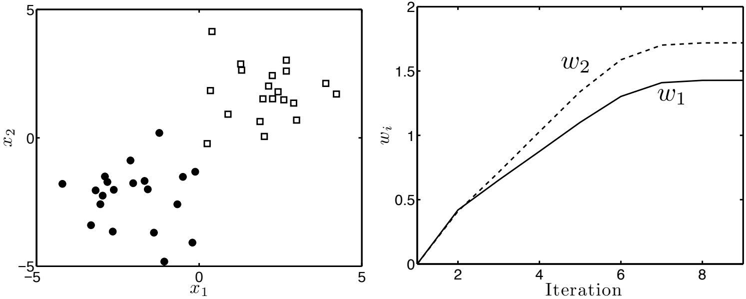
MAP

- When we met maximum likelihood, we could find $\widehat{\mathbf{w}}$ exactly with some algebra (in logistic regression, Cross Entropy is *convex*.).
- Can't do that here (can't solve $\frac{\partial g(\mathbf{w};\mathbf{X},\mathbf{t},\sigma^2)}{\partial \mathbf{w}} = \mathbf{0}$)
- Resort to numerical optimisation:
  1. Guess $\widehat{\mathbf{w}}$
  2. Change it a bit in a way that increases $g(\mathbf{w}; \mathbf{X}, \mathbf{t}, \sigma^2)$
  3. Repeat until no further increase is possible.

MAP

- When we met maximum likelihood, we could find $\widehat{\mathbf{w}}$ exactly with some algebra (in logistic regression, Cross Entropy is *convex*.).
- Can't do that here (can't solve $\frac{\partial g(\mathbf{w};\mathbf{X},\mathbf{t},\sigma^2)}{\partial \mathbf{w}} = \mathbf{0}$)
- Resort to numerical optimisation:
  1. Guess $\widehat{\mathbf{w}}$
  2. Change it a bit in a way that increases $g(\mathbf{w};\mathbf{X},\mathbf{t},\sigma^2)$
  3. Repeat until no further increase is possible.
- Many algorithms exist that differ in how they do step 2.
- e.g. **Gradient Descent** and **Newton-Raphson** (book Chapter 4)
  - You just need to know that sometimes we can't do things analytically and there are methods to help us!

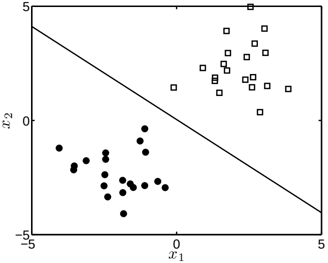
MAP – numerical optimisation for our data



- Left: Data.
- Right: Evolution of $\widehat{\mathbf{w}}$ in numerical optimisation.
- We set $\sigma^2 = 10$.

Decision boundary

- Once we have $\widehat{\mathbf{w}}$, we can classify new examples.
- Decision boundary is a useful visualisation:

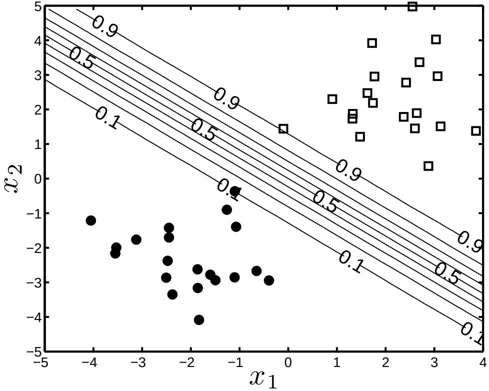


- Line corresponding to $P(t_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \widehat{\mathbf{w}}) = 0.5$.

$$0.5 = \frac{1}{2} = \frac{1}{1 + \exp(-\widehat{\mathbf{w}}^{\mathsf{T}} \mathbf{x}_{\text{new}})}.$$

So: $\exp(-\widehat{\mathbf{w}}^{\mathsf{T}} \mathbf{x}_{\text{new}}) = 1$. Or: $\widehat{\mathbf{w}}^{\mathsf{T}} \mathbf{x}_{\text{new}} = 0$

Predictive probabilities



- Contours of $P(t_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \widehat{\mathbf{w}})$.
- Do they look sensible?

Roadmap

- Find the most likely value of $\mathbf{w}$ – a point estimate.
- **Approximate $p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$ with something easier.**
- Sample from $p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$.

Laplace approximation

- Our second method involves **approximating** $p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$ with another distribution.
- i.e. Find a distribution $q(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$ which is similar.

Laplace approximation

- Our second method involves **approximating** $p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$ with another distribution.
- i.e. Find a distribution $q(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$ which is similar.
- What is 'similar'?
  - Mode (highest point) in same place.
  - Similar shape?
  - Might as well choose something that is easy to manipulate!

Laplace approximation

- Approximate $p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$ with a Gaussian:

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

- Where:

$$\boldsymbol{\mu} = \widehat{\mathbf{w}}, \;\; \boldsymbol{\Sigma}^{-1} = -\left.\frac{\partial^2 \log g(\mathbf{w}; \mathbf{X}, \mathbf{t}, \sigma^2)}{\partial \mathbf{w} \partial \mathbf{w}^\mathsf{T}}\right|_{\widehat{\mathbf{w}}}$$

- And:

$$\widehat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmax}} \; \log g(\mathbf{w}; \mathbf{X}, \mathbf{t}, \sigma^2)$$

- We already know $\widehat{\mathbf{w}}$. $\boldsymbol{\Sigma}$ is the negative of the inverse Hessian.

Laplace approximation

- Justification?
- Not covered in this course.
- Based on Taylor expansion of $\log g(\mathbf{w}; \mathbf{X}, \mathbf{t}, \sigma^2)$ around mode ($\widehat{\mathbf{w}}$).
  - Means approximation will be best at mode.
  - Expansion up to 2nd order terms 'looks' like a Gaussian.
- See book Chapter 4 for details.

Laplace approximation – 1D example

$$p(y|\alpha, \beta) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} y^{\alpha-1} \exp(-\beta y)$$

Laplace approximation – 1D example

$$
\begin{aligned}
p(y|\alpha,\beta) &= \frac{\beta^\alpha}{\Gamma(\alpha)} y^{\alpha-1} \exp(-\beta y) \\
\widehat{y} &= \frac{\alpha-1}{\beta}
\end{aligned}
$$

- Note, I happen to know what the mode is. You're not expected to be able to work this out!

Laplace approximation – 1D example

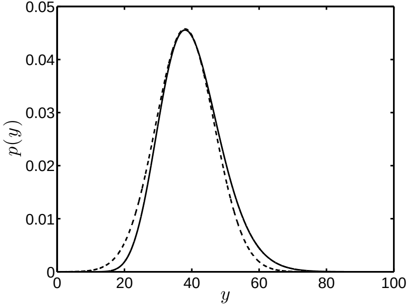
$$
\begin{aligned}
p(y|\alpha, \beta) &= \frac{\beta^\alpha}{\Gamma(\alpha)} y^{\alpha-1} \exp(-\beta y) \\
\widehat{y} &= \frac{\alpha - 1}{\beta} \\
\frac{\partial^2 \log p(.)}{\partial y^2} &= -\frac{\alpha - 1}{y^2} \\
\left.\frac{\partial^2 \log p(.)}{\partial y^2}\right|_{\widehat{y}} &= -\frac{\alpha - 1}{\widehat{y}^2}
\end{aligned}
$$

- Note, I happen to know what the mode is. You're not expected to be able to work this out!

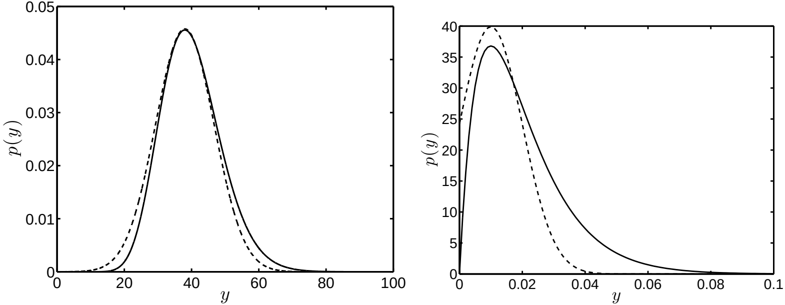
Laplace approximation – 1D example

$$
\begin{aligned}
p(y|\alpha,\beta) &= \frac{\beta^\alpha}{\Gamma(\alpha)} y^{\alpha-1}\exp(-\beta y) \\
\widehat{y} &= \frac{\alpha-1}{\beta} \\
\frac{\partial^2 \log p(.)}{\partial y^2} &= -\frac{\alpha-1}{y^2} \\
\left.\frac{\partial^2 \log p(.)}{\partial y^2}\right|_{\widehat{y}} &= -\frac{\alpha-1}{\widehat{y}^2} \\
q(y|\alpha,\beta) &= \mathcal{N}\left(\frac{\alpha-1}{\beta}, \frac{\widehat{y}^2}{\alpha-1}\right)
\end{aligned}
$$

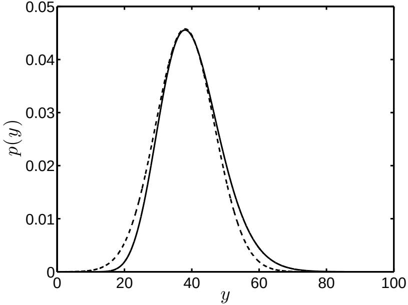
- Note, I happen to know what the mode is. You're not expected to be able to work this out!

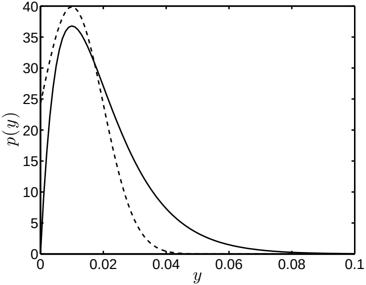


- Solid: true density. Dashed: approximation.
- Left: $\alpha = 20$, $\beta = 0.45$



- Solid: true density. Dashed: approximation.
- Left: $\alpha = 20$, $\beta = 0.45$
- Right: $\alpha = 2$, $\beta = 100$





- Solid: true density. Dashed: approximation.
- Left: $\alpha = 20, \ \beta = 0.45$
- Right: $\alpha = 2, \ \beta = 100$
- Approximation is best when density looks like a Gaussian (left).
- Approximation deteriorates as we move away from the mode (both).

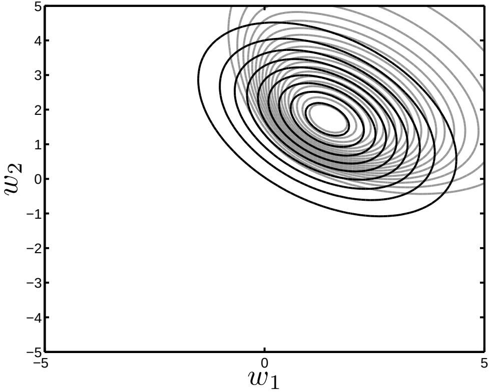
Laplace approximation for logistic regression

- Not going into the details here.
- $p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2) \approx \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.
- Find $\mu = \widehat{\mathbf{w}}$ (that maximises $g(\mathbf{w}; \mathbf{X}, \mathbf{t}, \sigma^2)$) by Gradient-Descent or Newton-Raphson (already done it – MAP).
- Find:

$$\boldsymbol{\Sigma}^{-1} = - \left. \frac{\partial^2 \log g(\mathbf{w}; \mathbf{X}, \mathbf{t}, \sigma^2)}{\partial \mathbf{w} \partial \mathbf{w}^\mathsf{T}} \right|_{\widehat{\mathbf{w}}}$$

- (Details given in book Chapter 4 if you're interested)
- How good an approximation is it?

Laplace approximation for logistic regression



- Dark lines – approximation. Light lines – proportional to $p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$.
- Approximation is OK.
- As expected, it gets worse as we travel away from the mode.

Predictions with the Laplace approximation

- We have $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ as an approximation to $p(\mathbf{w}|\mathbf{X}, \mathbf{t})$.
- Can we use it to make predictions?

Predictions with the Laplace approximation

- We have $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ as an approximation to $p(\mathbf{w}|\mathbf{X}, \mathbf{t})$.
- Can we use it to make predictions?
- Need to evaluate:

$$\begin{aligned} P(t_{\text{new}} = 1|\mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t}) &= \mathbf{E}_{\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})}\left\{P(t_{\text{new}} = 1|\mathbf{x}_{\text{new}}, \mathbf{w})\right\} \\ &= \int \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \frac{1}{1 + \exp(-\mathbf{w}^{\mathsf{T}}\mathbf{x}_{\text{new}})} \, d\mathbf{w} \end{aligned}$$

Predictions with the Laplace approximation

- We have $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ as an approximation to $p(\mathbf{w}|\mathbf{X}, \mathbf{t})$.
- Can we use it to make predictions?
- Need to evaluate:

$$\begin{aligned} P(t_{\text{new}} = 1|\mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t}) &= \mathbf{E}_{\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})} \left\{ P(t_{\text{new}} = 1|\mathbf{x}_{\text{new}}, \mathbf{w}) \right\} \\ &= \int \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \frac{1}{1 + \exp(-\mathbf{w}^{\mathsf{T}} \mathbf{x}_{\text{new}})} \; d\mathbf{w} \end{aligned}$$

- Cannot do this! So, what was the point?

Predictions with the Laplace approximation

- We have $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ as an approximation to $p(\mathbf{w}|\mathbf{X}, \mathbf{t})$.
- Can we use it to make predictions?
- Need to evaluate:

$$
\begin{aligned}
P(t_{\text{new}} = 1|\mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t}) &= \mathbf{E}_{\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})}\left\{P(t_{\text{new}} = 1|\mathbf{x}_{\text{new}}, \mathbf{w})\right\} \\
&= \int \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \frac{1}{1 + \exp(-\mathbf{w}^{\mathsf{T}}\mathbf{x}_{\text{new}})} \, d\mathbf{w}
\end{aligned}
$$

- Cannot do this! So, what was the point?
- Sampling from $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is **easy**
  - And we can approximate an expectation with samples!

Predictions with the Laplace approximation

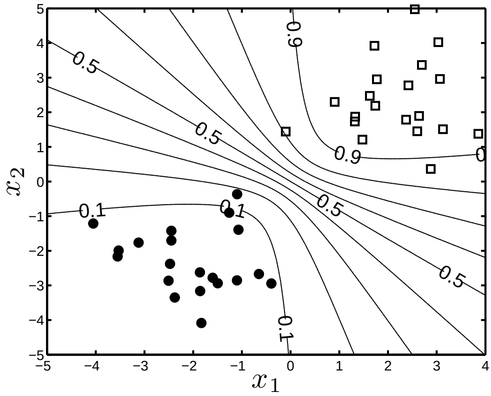
- Draw S samples $\mathbf{w}_1, \ldots, \mathbf{w}_S$ from $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

$$\mathbf{E}_{\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})} \left\{ P(t_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{w}) \right\} \approx \frac{1}{S} \sum_{s=1}^{S} \frac{1}{1 + \exp(-\mathbf{w}_s^{\mathsf{T}} \mathbf{x}_{\text{new}})}$$

Predictions with the Laplace approximation

- Draw S samples $\mathbf{w}_1, \ldots, \mathbf{w}_S$ from $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

$$\mathbf{E}_{\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})}\left\{P(t_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{w})\right\} \approx \frac{1}{S}\sum_{s=1}^{S} \frac{1}{1 + \exp(-\mathbf{w}_s^{\mathsf{T}}\mathbf{x}_{\text{new}})}$$

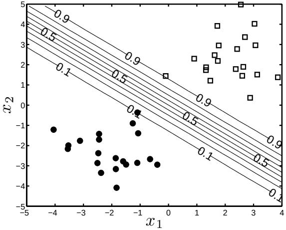


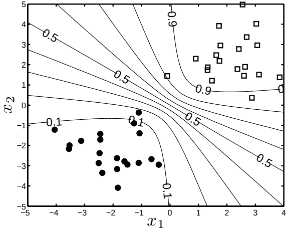
- Contours of $P(t_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t})$.
- Better than those from the point prediction?

Point prediction v Laplace approximation

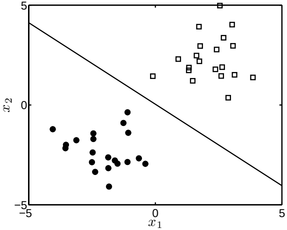
Why the difference?

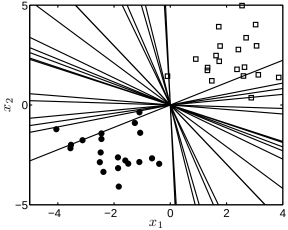
Point prediction v Laplace approximation





Why the difference?





Laplace uses a distribution ($\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$) over $\mathbf{w}$ (and therefore a distribution over decision boundaries) and hence has less certainty.

Summary – roadmap

- Defined a squashing function that meant we could model $P(t_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{w}) = h(\mathbf{w}^{\mathsf{T}} \mathbf{x}_{\text{new}})$
- Wanted to make 'Bayesian predictions': average over all posterior values of $\mathbf{w}$.
- Couldn't do it exactly.
- Tried a point estimate (MAP) and an approximate distribution (via Laplace).
- Laplace probability contours looked more sensible (to me at least!)

Summary – roadmap

- Defined a squashing function that meant we could model $P(t_{\text{new}} = 1|\mathbf{x}_{\text{new}}, \mathbf{w}) = h(\mathbf{w}^{\mathsf{T}}\mathbf{x}_{\text{new}})$
- Wanted to make 'Bayesian predictions': average over all posterior values of $\mathbf{w}$.
- Couldn't do it exactly.
- Tried a point estimate (MAP) and an approximate distribution (via Laplace).
- Laplace probability contours looked more sensible (to me at least!)
- Next:
  - Find the most likely value of $\mathbf{w}$ – a point estimate.
  - Approximate $p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$ with something easier.
  - **Sample from $p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$.**

MCMC sampling

- Laplace approximation still didn't let us exactly evaluate the expectation we need for predictions.
- But....we could easily sample from it and approximate our approximation.

MCMC sampling

- Laplace approximation still didn't let us exactly evaluate the expectation we need for predictions.
- But....we could easily sample from it and approximate our approximation.

- Good news! If we're happy to sample, we can sample directly from $p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \sigma^2)$ even though we can't compute it!
- i.e. don't need to use an approximation like Laplace.
- Various algorithms exist – we'll use *Metropolis-Hastings*

Aside – sampling from things we can't compute

- At first glance it seems strange – we can roll the die but we can't make it!
- But – it's pretty common in the world!
- Darts.....

Darts

- I want to know the probability that I hit treble 20 when I aim for treble 20.
- The distribution over where the dart lands when I aim treble 20:
$$p(\mathbf{x}|\text{stuff})$$

Darts

- I want to know the probability that I hit treble 20 when I aim for treble 20.
- The distribution over where the dart lands when I aim treble 20:

$$p(\mathbf{x}|\text{stuff})$$

- Define function $f(\mathbf{x}) = 1$ if $\mathbf{x}$ in treble 20 and 0 otherwise.

Darts

- I want to know the probability that I hit treble 20 when I aim for treble 20.
- The distribution over where the dart lands when I aim treble 20:
$$p(\mathbf{x}|\text{stuff})$$
- Define function $f(\mathbf{x}) = 1$ if $\mathbf{x}$ in treble 20 and 0 otherwise.
- Probability I hit treble twenty is therefore:

$$\int f(\mathbf{x})p(\mathbf{x}|\text{stuff})\ d\mathbf{x}$$

Darts

- I want to know the probability that I hit treble 20 when I aim for treble 20.
- The distribution over where the dart lands when I aim treble 20:

$$p(\mathbf{x}|\text{stuff})$$

- Define function $f(\mathbf{x}) = 1$ if $\mathbf{x}$ in treble 20 and 0 otherwise.
- Probability I hit treble twenty is therefore:

$$\int f(\mathbf{x})p(\mathbf{x}|\text{stuff})\ d\mathbf{x}$$

- Can't even begin to work out how to write down $p(\mathbf{x}|\text{stuff})$.

Darts

- I want to know the probability that I hit treble 20 when I aim for treble 20.
- The distribution over where the dart lands when I aim treble 20:

$$p(\mathbf{x}|\text{stuff})$$

- Define function $f(\mathbf{x}) = 1$ if $\mathbf{x}$ in treble 20 and 0 otherwise.
- Probability I hit treble twenty is therefore:

$$\int f(\mathbf{x})p(\mathbf{x}|\text{stuff})\ d\mathbf{x}$$

- Can't even begin to work out how to write down $p(\mathbf{x}|\text{stuff})$.
- But can sample – throw S darts, $\mathbf{x}_1, \ldots, \mathbf{x}_S$!
- Compute:

$$\frac{1}{S}\sum_{s=1}^{S} f(\mathbf{x}_s)$$

Back to the script: Metropolis-Hastings

- Produces a sequence of samples – $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_s, \ldots$
- Imagine we've just produced $\mathbf{w}_{s-1}$

Back to the script: Metropolis-Hastings

- Produces a sequence of samples – $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_s, \ldots$
- Imagine we've just produced $\mathbf{w}_{s-1}$
- MH first *proposes* a possible $\mathbf{w}_s$ (call it $\widetilde{\mathbf{w}_s}$) based on $\mathbf{w}_{s-1}$.

Back to the script: Metropolis-Hastings

- Produces a sequence of samples – $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_s, \ldots$
- Imagine we've just produced $\mathbf{w}_{s-1}$
- MH first *proposes* a possible $\mathbf{w}_s$ (call it $\widetilde{\mathbf{w}_s}$) based on $\mathbf{w}_{s-1}$.

- MH then decides whether or not to *accept* $\widetilde{\mathbf{w}_s}$
  - If accepted, $\mathbf{w}_s = \widetilde{\mathbf{w}_s}$
  - If not, $\mathbf{w}_s = \mathbf{w}_{s-1}$

Back to the script: Metropolis-Hastings

- Produces a sequence of samples – $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_s, \ldots$
- Imagine we've just produced $\mathbf{w}_{s-1}$
- MH first *proposes* a possible $\mathbf{w}_s$ (call it $\widetilde{\mathbf{w}_s}$) based on $\mathbf{w}_{s-1}$.
- MH then decides whether or not to *accept* $\widetilde{\mathbf{w}_s}$
  - If accepted, $\mathbf{w}_s = \widetilde{\mathbf{w}_s}$
  - If not, $\mathbf{w}_s = \mathbf{w}_{s-1}$
- Two distinct steps – proposal and acceptance.

MH – proposal

- Treat $\widetilde{\mathbf{w}_s}$ as a random variable conditioned on $\mathbf{w}_{s-1}$
- i.e. need to define $p(\widetilde{\mathbf{w}_s}|\mathbf{w}_{s-1})$
  - Note that this does not necessarily have to be similar to posterior we're trying to sample from.
- Can choose *whatever we like*!

MH – proposal

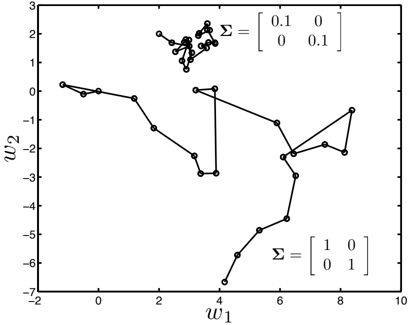
- Treat $\widetilde{\mathbf{w}_s}$ as a random variable conditioned on $\mathbf{w}_{s-1}$
- i.e. need to define $p(\widetilde{\mathbf{w}_s}|\mathbf{w}_{s-1})$
  - Note that this does not necessarily have to be similar to posterior we're trying to sample from.
- Can choose *whatever we like*!
- e.g. use a Gaussian centered on $\mathbf{w}_{s-1}$ with some covariance:

$$p(\widetilde{\mathbf{w}_s}|\mathbf{w}_{s-1}, \boldsymbol{\Sigma}_p) = \mathcal{N}(\mathbf{w}_{s-1}, \boldsymbol{\Sigma}_p)$$

MH – proposal

- Treat $\widetilde{\mathbf{w}_s}$ as a random variable conditioned on $\mathbf{w}_{s-1}$
- i.e. need to define $p(\widetilde{\mathbf{w}_s}|\mathbf{w}_{s-1})$
  - Note that this does not necessarily have to be similar to posterior we're trying to sample from.
- Can choose *whatever we like*!
- e.g. use a Gaussian centered on $\mathbf{w}_{s-1}$ with some covariance:

$$p(\widetilde{\mathbf{w}_s}|\mathbf{w}_{s-1}, \mathbf{\Sigma}_p) = \mathcal{N}(\mathbf{w}_{s-1}, \mathbf{\Sigma}_p)$$



MH – acceptance

- Choice of acceptance based on the following ratio:

$$r = \frac{p(\widetilde{\mathbf{w}_s}|\mathbf{X}, \mathbf{t}, \sigma^2)}{p(\mathbf{w}_{s-1}|\mathbf{X}, \mathbf{t}, \sigma^2)} \frac{p(\mathbf{w}_{s-1}|\widetilde{\mathbf{w}_s}, \mathbf{\Sigma}_p)}{p(\widetilde{\mathbf{w}_s}|\mathbf{w}_{s-1}, \mathbf{\Sigma}_p)}.$$

MH – acceptance

- Choice of acceptance based on the following ratio:

$$r = \frac{p(\widetilde{\mathbf{w}_s}|\mathbf{X}, \mathbf{t}, \sigma^2)}{p(\mathbf{w}_{s-1}|\mathbf{X}, \mathbf{t}, \sigma^2)} \frac{p(\mathbf{w}_{s-1}|\widetilde{\mathbf{w}_s}, \mathbf{\Sigma}_p)}{p(\widetilde{\mathbf{w}_s}|\mathbf{w}_{s-1}, \mathbf{\Sigma}_p)}.$$

- Which simplifies to (all of which we can compute):

$$r = \frac{g(\widetilde{\mathbf{w}_s}; \mathbf{X}, \mathbf{t}, \sigma^2)}{g(\mathbf{w}_{s-1}; \mathbf{X}, \mathbf{t}, \sigma^2)} \frac{p(\mathbf{w}_{s-1}|\widetilde{\mathbf{w}_s}, \mathbf{\Sigma}_p)}{p(\widetilde{\mathbf{w}_s}|\mathbf{w}_{s-1}, \mathbf{\Sigma}_p)}.$$

MH – acceptance

- Choice of acceptance based on the following ratio:

$$r = \frac{p(\widetilde{\mathbf{w}_s}|\mathbf{X}, \mathbf{t}, \sigma^2)}{p(\mathbf{w}_{s-1}|\mathbf{X}, \mathbf{t}, \sigma^2)} \frac{p(\mathbf{w}_{s-1}|\widetilde{\mathbf{w}_s}, \mathbf{\Sigma}_p)}{p(\widetilde{\mathbf{w}_s}|\mathbf{w}_{s-1}, \mathbf{\Sigma}_p)}.$$

- Which simplifies to (all of which we can compute):

$$r = \frac{g(\widetilde{\mathbf{w}_s}; \mathbf{X}, \mathbf{t}, \sigma^2)}{g(\mathbf{w}_{s-1}; \mathbf{X}, \mathbf{t}, \sigma^2)} \frac{p(\mathbf{w}_{s-1}|\widetilde{\mathbf{w}_s}, \mathbf{\Sigma}_p)}{p(\widetilde{\mathbf{w}_s}|\mathbf{w}_{s-1}, \mathbf{\Sigma}_p)}.$$

- We now use the following rules:
  - If $r \geq 1$, accept: $\mathbf{w}_s = \widetilde{\mathbf{w}_s}$.
  - If $r < 1$, accept with probability r.

MH – acceptance

- Choice of acceptance based on the following ratio:

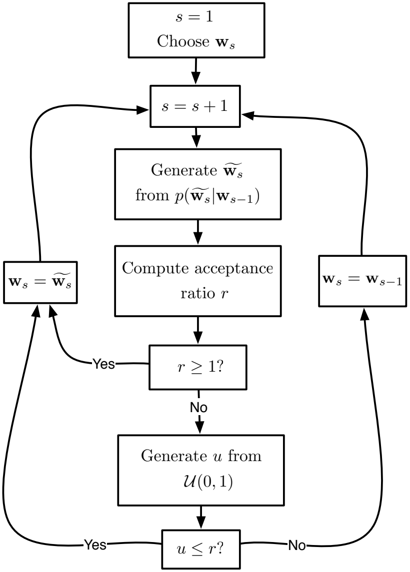
$$r = \frac{p(\widetilde{\mathbf{w}_s}|\mathbf{X}, \mathbf{t}, \sigma^2)}{p(\mathbf{w}_{s-1}|\mathbf{X}, \mathbf{t}, \sigma^2)} \frac{p(\mathbf{w}_{s-1}|\widetilde{\mathbf{w}_s}, \boldsymbol{\Sigma}_p)}{p(\widetilde{\mathbf{w}_s}|\mathbf{w}_{s-1}, \boldsymbol{\Sigma}_p)}.$$

- Which simplifies to (all of which we can compute):

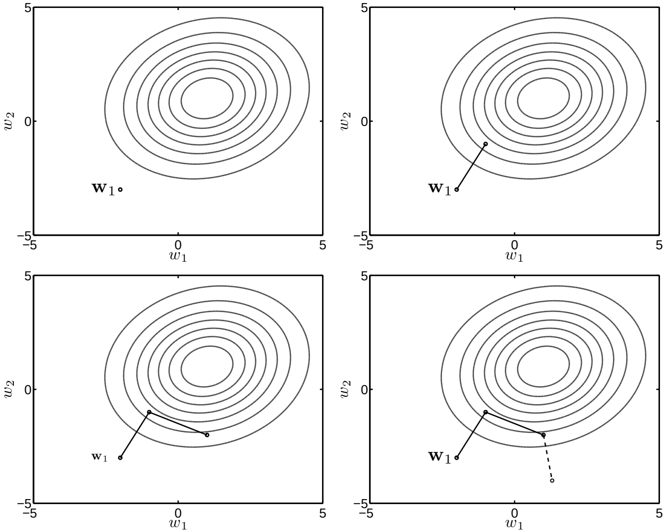
$$r = \frac{g(\widetilde{\mathbf{w}_s}; \mathbf{X}, \mathbf{t}, \sigma^2)}{g(\mathbf{w}_{s-1}; \mathbf{X}, \mathbf{t}, \sigma^2)} \frac{p(\mathbf{w}_{s-1}|\widetilde{\mathbf{w}_s}, \boldsymbol{\Sigma}_p)}{p(\widetilde{\mathbf{w}_s}|\mathbf{w}_{s-1}, \boldsymbol{\Sigma}_p)}.$$

- We now use the following rules:
  - If $r \geq 1$, accept: $\mathbf{w}_s = \widetilde{\mathbf{w}_s}$.
  - If $r < 1$, accept with probability r.
- If we do this enough, we'll eventually be sampling from $p(\mathbf{w}|\mathbf{X}, \mathbf{t})$, no matter where we started!
  - i.e. for any $\mathbf{w}_1$

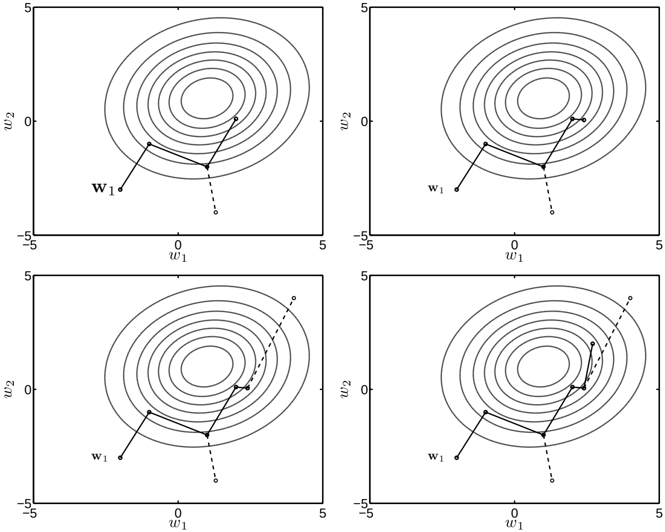
MH – flowchart



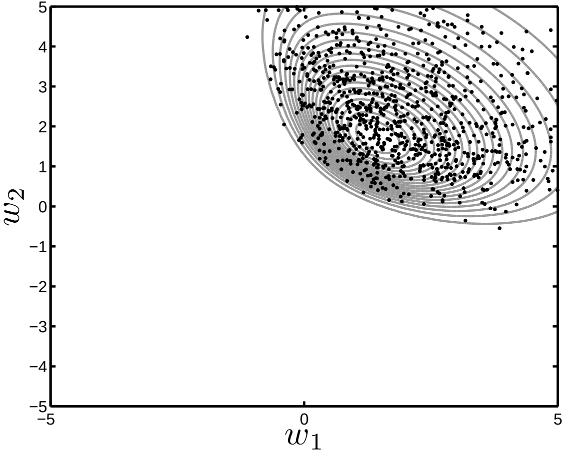
MH – walkthrough 1



MH – walkthrough 2



What do the samples look like?



- 1000 samples from the posterior using MH.

Predictions with MH

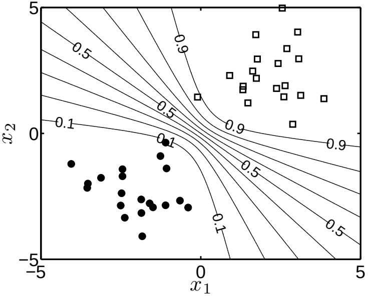
- MH provides us with a set of samples – $\mathbf{w}_1, \ldots, \mathbf{w}_S$.
- These can be used like the samples from the Laplace approximation:

$$
\begin{aligned}
P(t_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t}, \sigma^2) &= \mathbf{E}_{p(\mathbf{w}|\mathbf{X},\mathbf{t},\sigma^2)} \left\{ P(t_{\text{new}} | \mathbf{x}_{\text{new}}, \mathbf{w}) \right\} \\
&\approx \frac{1}{S} \sum_{s=1}^{S} \frac{1}{1 + \exp(-\mathbf{w}_s^{\mathsf{T}} \mathbf{x}_{\text{new}})}
\end{aligned}
$$

Predictions with MH

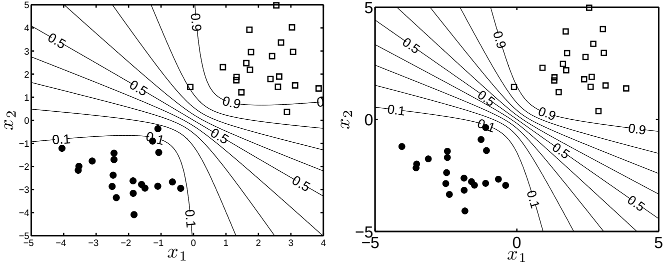
- MH provides us with a set of samples – $\mathbf{w}_1, \ldots, \mathbf{w}_S$.
- These can be used like the samples from the Laplace approximation:

$$\begin{aligned} P(t_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t}, \sigma^2) &= \mathbf{E}_{p(\mathbf{w}|\mathbf{X},\mathbf{t},\sigma^2)} \left\{ P(t_{\text{new}} | \mathbf{x}_{\text{new}}, \mathbf{w}) \right\} \\ &\approx \frac{1}{S} \sum_{s=1}^{S} \frac{1}{1 + \exp(-\mathbf{w}_s^\mathsf{T} \mathbf{x}_{\text{new}})} \end{aligned}$$



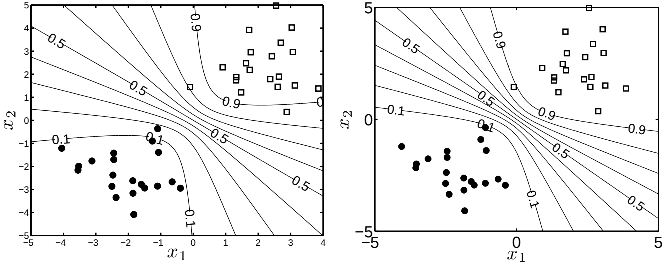
- Contours of $P(t_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t}, \sigma^2)$

Laplace vs. MH

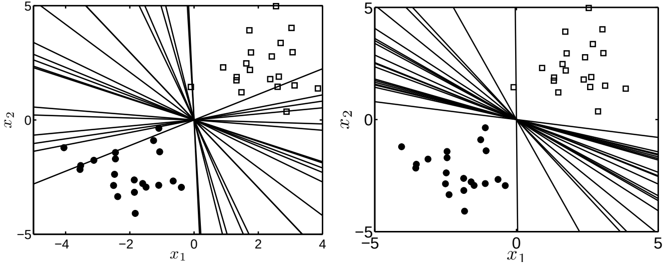


Why?

Laplace vs. MH

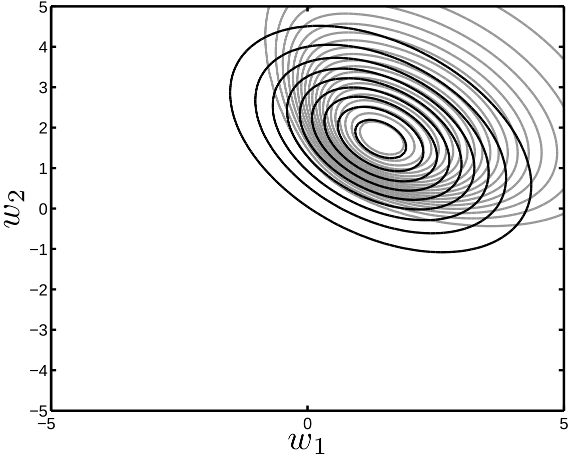


Why?



Laplace approximation (left) allows some *bad* boundaries

Laplace vs. MH



Approximate posterior allows some values of w_1 and w_2 that are very unlikely in true posterior.

Summary

- Introduced logistic regression – a probabilistic binary classifier.
- Saw that we couldn't compute the posterior.
- Introduced *examples of* three alternatives:
    - Point estimate – MAP solution.
    - Approximate the density – Laplace.
    - Sample – Metropolis-Hastings.
- Each is better than the last (in terms of predictions)....
- ...but each has greater complexity!
- To think about:
    - What if posterior is multi-modal?