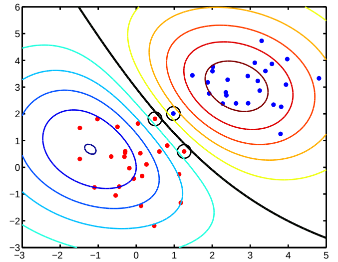
Advanced Statistical Inference
Bayesian Logistic Regression

Maurizio Filippone
Maurizio.Filippone@eurecom.fr

Department of Data Science
EURECOM

Classification



- A set of N objects with attributes (usually vector) $\mathbf{x}_n$.
- Each object has an associated response (or label) y_n.
- Binary classification: $y_n \in \{0, 1\}$ or $y_n \in \{-1, 1\}$,
  - (depends on algorithm).
- Multi-class classification: $y_n \in \{1, 2, \ldots, K\}$.

Probabilistic v non-probabilistic classifiers

Classifier is trained on $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^\top$ and $\mathbf{y} = (y_1, \ldots, y_n)^\top$ and then used to classify $\mathbf{x}_*$.

- Probabilistic classifiers produce a probability of class membership $P(y_* = k|\mathbf{x}_*, \mathbf{X}, \mathbf{y})$
  - e.g. binary classification: $P(y_* = 1|\mathbf{x}_*, \mathbf{X}, \mathbf{y})$ and $P(y_* = 0|\mathbf{x}_*, \mathbf{X}, \mathbf{y})$.
- Non-probabilistic classifiers produce a hard assignment
  - e.g. $y_* = 1$ or $y_* = 0$.
- Which one to choose depends on the application....

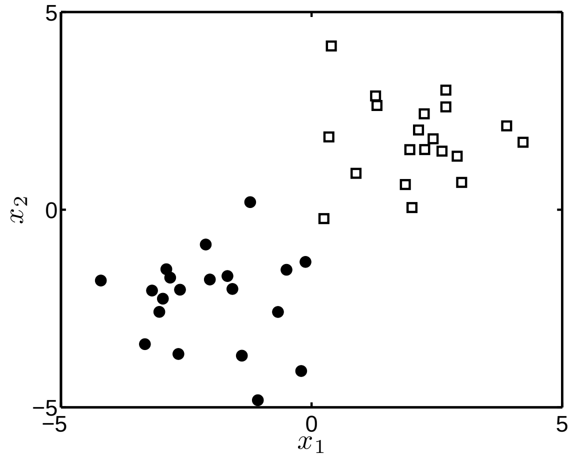
Probabilistic v non-probabilistic classifiers

- Probabilities provide us with more information – $P(y_* = 1) = 0.6$ is more useful than $y_* = 1$.
  - Tells us how **sure** the algorithm is.
- Particularly important where cost of misclassification is high and imbalanced.
  - e.g. Diagnosis: telling a diseased person they are healthy is much worse than telling a healthy person they are diseased.
- Extra information (probability) often comes at a cost.

Classification syllabus

- We will study two probabilistic classifiers:
  - Bayes classifier.
  - **Logistic regression.**

Some data



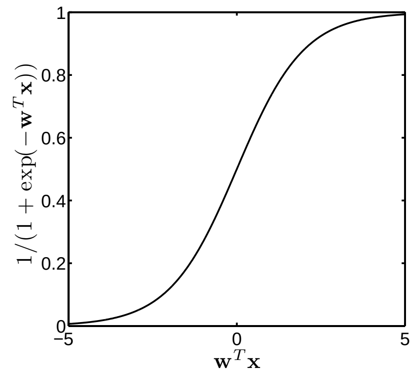
Logistic regression

- Similarly to regression, we could think about modeling $P(y_* = k|\mathbf{x}_*, \mathbf{w})$ through some $f(\mathbf{x}_*; \mathbf{w})$ with parameters $\mathbf{w}$.
- We've seen $f(\mathbf{x}_*; \mathbf{w}) = \mathbf{w}^\top \mathbf{x}_*$ before – can we use it here?
  - No – output is unbounded and so can't be a probability.
- But, can use $P(y_* = k|\mathbf{x}_*, \mathbf{w}) = h(f(\mathbf{x}_*; \mathbf{w}))$ where $h(\cdot)$ squashes $f(\mathbf{x}_*; \mathbf{w})$ to lie between 0 and 1 – a probability.

$h(\cdot)$

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

$$P(y_* = 1|\mathbf{x}_*, \mathbf{w}) = h(\mathbf{w}^\top \mathbf{x}_*) = \frac{1}{1 + \exp(-\mathbf{w}^\top \mathbf{x}_*)}$$



Bayesian logistic regression

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

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

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

$$P(y_* = 1|\mathbf{x}_*,\mathbf{X},\mathbf{y}) = \mathbf{E}_{p(\mathbf{w}|\mathbf{X},\mathbf{y})}\left[P(y_* = 1|\mathbf{x}_*,\mathbf{w})\right]$$

- Sounds good so far....

Defining a prior

- Choose a Gaussian prior:

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

  - 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:

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

- We have already defined this! If $y_n = 1$:

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

- and if $y_n = 0$:

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

Posterior

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

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

$$p(\mathbf{y}|\mathbf{X},s) = \int p(\mathbf{y}|\mathbf{X},\mathbf{w})p(\mathbf{w}|s)\ d\mathbf{w}$$

What can we compute?

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

- For simplicity, let's drop the dependence on s
- We can compute $p(\mathbf{y}|\mathbf{X},\mathbf{w})p(\mathbf{w})$
  - Define $g(\mathbf{w}) = p(\mathbf{y}|\mathbf{X},\mathbf{w})p(\mathbf{w})$
- Armed with this, we have three options:
  - Find the most likely value of $\mathbf{w}$ – a point estimate.
  - Approximate $p(\mathbf{w}|\mathbf{y},\mathbf{X})$ with something easier.
  - Sample from $p(\mathbf{w}|\mathbf{y},\mathbf{X})$.
- We'll cover examples of each of these in turn....
- These are not the only ways of approximating/sampling!
- They are also general - not unique to logistic regression.

MAP estimate

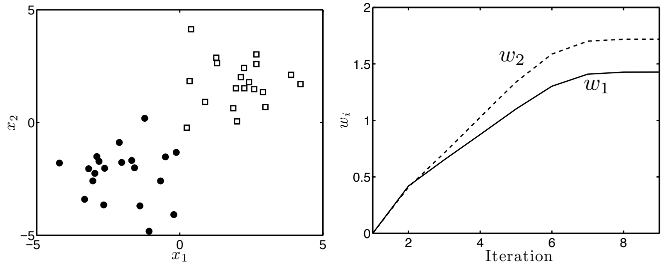
- Out first method is to find the value of $\mathbf{w}$ that maximizes $p(\mathbf{w}|\mathbf{y},\mathbf{X})$ (call it $\hat{\mathbf{w}}$).
  - $g(\mathbf{w}) \propto p(\mathbf{w}|\mathbf{y},\mathbf{X})$
  - $\hat{\mathbf{w}}$ therefore also maximizes $g(\mathbf{w})$.
- Similar to maximum likelihood but additional effect of prior.
- Known as MAP (maximum a posteriori) solution.
- Once we have $\hat{\mathbf{w}}$, make predictions with:

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

MAP

- When we met maximum likelihood, we could find $\hat{\mathbf{w}}$ exactly with some algebra.
- Can't do that here (can't solve $\nabla_{\mathbf{w}} g(\mathbf{w}) = \mathbf{0}$)
- Resort to numerical optimization:
  1. Guess $\hat{\mathbf{w}}$
  2. Change it a bit in a way that increases $g(\mathbf{w})$
  3. Repeat until no further increase is possible.
- Many algorithms exist that differ in how they do step 2.
- e.g. **Newton-Raphson**
  - Not covered in this course. You just need to know that sometimes we can't do things analytically and there are methods to help us!

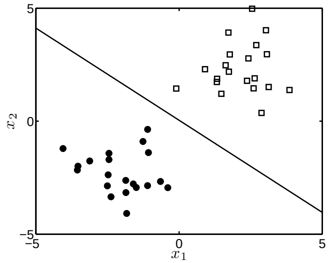
MAP – numerical optimization for our data



- Left: Data.
- Right: Evolution of $\hat{\mathbf{w}}$ in numerical optimization.

Decision boundary

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

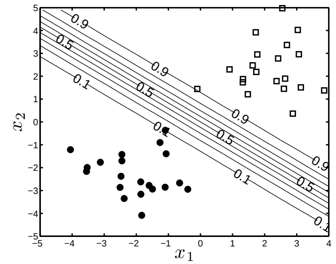


- Line corresponding to $P(y_* = 1|\mathbf{x}_*, \hat{\mathbf{w}}) = 0.5$.

$$0.5 = \frac{1}{2} = \frac{1}{1+\exp(-\hat{\mathbf{w}}^\top \mathbf{x}_*)}.$$

So: $\exp(-\hat{\mathbf{w}}^\top \mathbf{x}_*) = 1$. Or: $\hat{\mathbf{w}}^\top \mathbf{x}_* = 0$

Predictive probabilities



- Contours of $P(y_* = 1|\mathbf{x}_*, \hat{\mathbf{w}})$.
- Do they look sensible?

Roadmap

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

Laplace approximation

- **Approximating** $p(\mathbf{w}|\mathbf{y}, \mathbf{X})$ with another distribution.
- i.e. Find a distribution $q(\mathbf{w}|\mathbf{y}, \mathbf{X})$ which is similar.
- What is 'similar'?
  - Mode (highest point) in same place.
  - Similar shape?
  - Might as well choose something that is easy to manipulate!
- Approximate $p(\mathbf{w}|\mathbf{y}, \mathbf{X}, s)$ with a Gaussian:

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

- Where:

$$\boldsymbol{\mu} = \hat{\mathbf{w}}, \quad \boldsymbol{\Sigma}^{-1} = -\nabla_{\mathbf{w}}\nabla_{\mathbf{w}} \log[g(\mathbf{w})]\Big|_{\hat{\mathbf{w}}}$$

- And:

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmax}} \; \log[g(\mathbf{w})]$$

- We already know $\hat{\mathbf{w}}$.

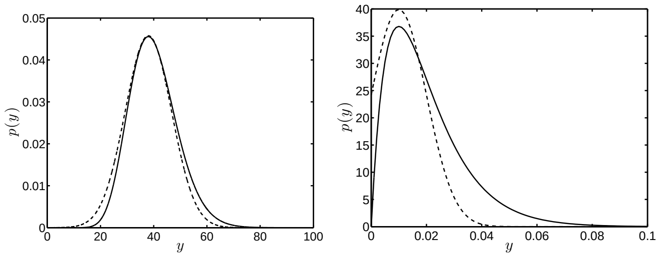
Laplace approximation

- Justification?
- Based on Taylor expansion of $\log[g(\mathbf{w})]$ around mode ($\hat{\mathbf{w}}$).
  - Means approximation will be best at mode.
  - Expansion up to 2nd order terms 'looks' like a Gaussian.

Laplace approximation – 1D example

- Laplace approximation of the Gamma density function:

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

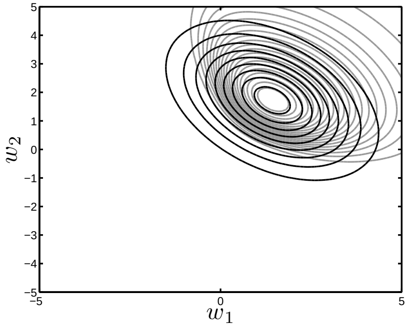


- Solid: true density. Dashed: approximation.
- Left: $\alpha = 20$, $\beta = 0.5$
- Right: $\alpha = 2$, $\beta = 100$
- Approximation is best when density looks like a Gaussian (left).
- Approximation deteriorates as we move away from the mode (both).

Laplace approximation for logistic regression

- Not going into the details here.
- $p(\mathbf{w}|\mathbf{y},\mathbf{X}) \approx q(\mathbf{w}|\mathbf{y},\mathbf{X}) = \mathcal{N}(\mathbf{w}|\boldsymbol{\mu},\boldsymbol{\Sigma})$.
- Find $\mu = \hat{\mathbf{w}}$ (that maximizes $g(\mathbf{w})$) by Newton-Raphson (already done it – MAP).
- Find:
$$\boldsymbol{\Sigma}^{-1} = -\nabla_{\mathbf{w}}\nabla_{\mathbf{w}}\log[g(\mathbf{w})]\Big|_{\hat{\mathbf{w}}}$$
- How good an approximation is it?

Laplace approximation for logistic regression



- Black – approximation. Grey – $p(\mathbf{w}|\mathbf{y}, \mathbf{X})$.
- Approximation is OK.
- As expected, it gets worse as we move away from the mode.

Predictions with the Laplace approximation

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

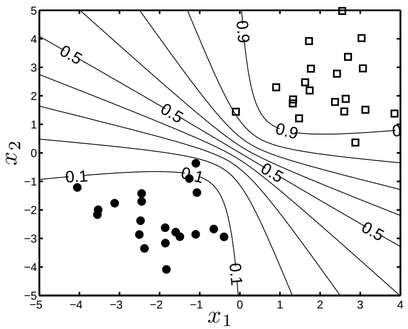
$$\begin{aligned} P(y_* = 1|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) &= \mathrm{E}_{\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})}\left[P(y_* = 1|\mathbf{x}_*, \mathbf{w})\right] \\ &= \int \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \frac{1}{1 + \exp(-\mathbf{w}^\top \mathbf{x}_*)} \, d\mathbf{w} \end{aligned}$$

- 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

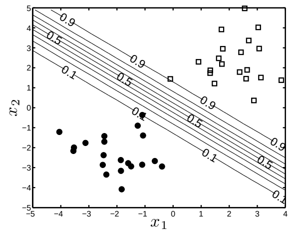
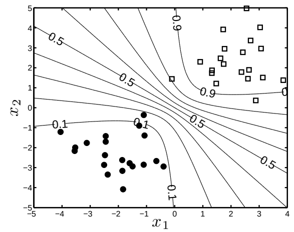
- Draw S samples $\mathbf{w}_1, \ldots, \mathbf{w}_S$ from $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

$$\mathrm{E}_{\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})}\left[P(y_* = 1|\mathbf{x}_*, \mathbf{w})\right] \approx \frac{1}{S} \sum_{s=1}^{S} \frac{1}{1 + \exp(-\mathbf{w}_s^\top \mathbf{x}_*)}$$

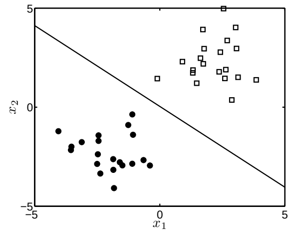
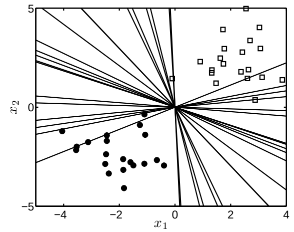


- Contours of $P(y_* = 1|\mathbf{x}_*, \mathbf{y}, \mathbf{X})$.
- Better than those from the point prediction?

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 $P(y_* = 1|\mathbf{x}_*, \mathbf{w}) = h(\mathbf{w}^\top \mathbf{x}_*)$
- 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 $p(\mathbf{w}|\mathbf{y}, \mathbf{X})$ with something easier.
  - **Sample from $p(\mathbf{w}|\mathbf{y}, \mathbf{X})$.**

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 $p(\mathbf{w}|\mathbf{y}, \mathbf{X})$ 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

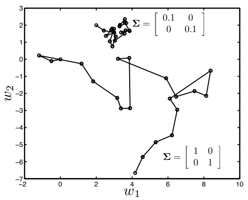
Back to the script: Metropolis-Hastings

- 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 firsts 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!
- 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 – acceptance

- Choice of acceptance based on the following ratio:

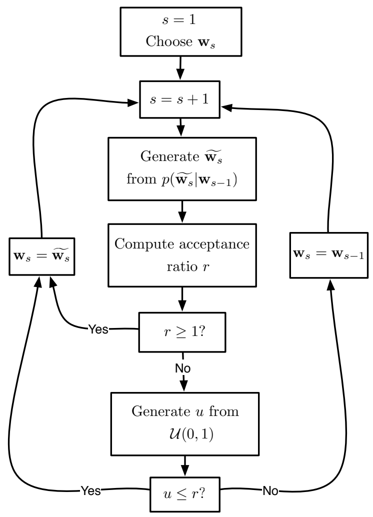
$$r = \frac{p(\widetilde{\mathbf{w}}_s|\mathbf{y}, \mathbf{X})}{p(\mathbf{w}_{s-1}|\mathbf{y}, \mathbf{X})} \frac{p(\mathbf{w}_{s-1}|\widetilde{\mathbf{w}}_s, \boldsymbol{\Sigma}_p)}{p(\widetilde{\mathbf{w}}_s|\mathbf{w}_{s-1}, \boldsymbol{\Sigma}_p)}.$$

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

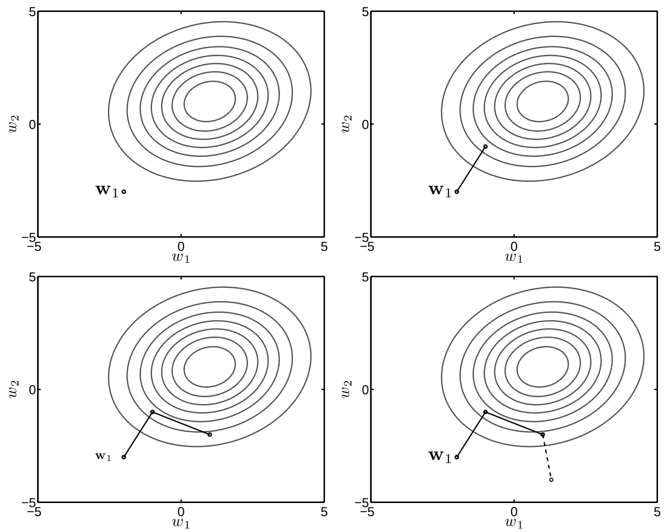
$$r = \frac{g(\widetilde{\mathbf{w}}_s; \mathbf{y}, \mathbf{X})}{g(\mathbf{w}_{s-1}; \mathbf{y}, \mathbf{X})} \frac{p(\mathbf{w}_{s-1}|\widetilde{\mathbf{w}}_s, \boldsymbol{\Sigma}_p)}{p(\widetilde{\mathbf{w}}_s|\mathbf{w}_{s-1}, \boldsymbol{\Sigma}_p)}.$$

- 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 $p(\mathbf{w}|\mathbf{y}, \mathbf{X})$, no matter where we started!
  - i.e. for any $\mathbf{w}_1$

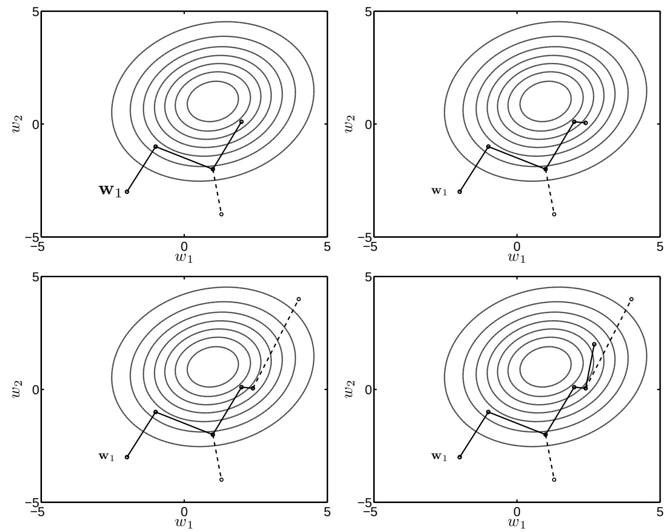
MH – flowchart



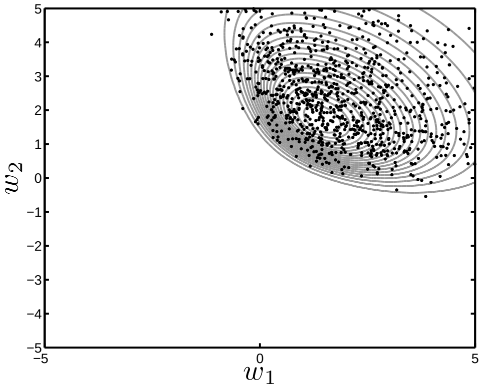
MH – walkthrough 1



MH – walkthrough 2



What do the samples look like?

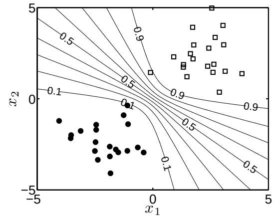


- 1000 samples from the posterior using MH.

Predictions with MH

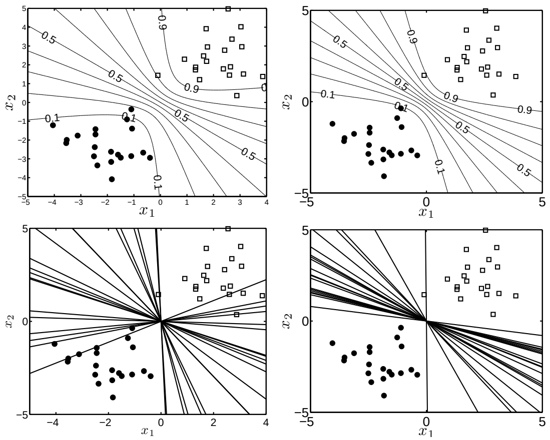
- MH provides us with a set of samples – $\mathbf{w}_1, \ldots, \mathbf{w}_S$.
- These can be used like the samples from the Laplace approximation:

$$
\begin{aligned}
P(y_* = 1|\mathbf{x}_*, \mathbf{y}, \mathbf{X}) &= \mathrm{E}_{p(\mathbf{w}|\mathbf{y},\mathbf{X})}\left[P(y_*|\mathbf{x}_*, \mathbf{w})\right] \\
&\approx \frac{1}{S}\sum_{s=1}^{S} \frac{1}{1+\exp(-\mathbf{w}_s^\top \mathbf{x}_*)}
\end{aligned}
$$



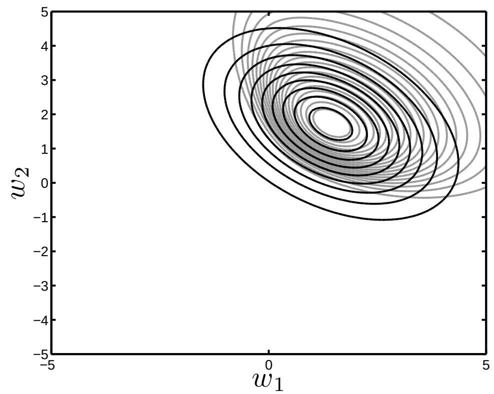
- Contours of $P(y_* = 1|\mathbf{x}_*, \mathbf{y}, \mathbf{X})$

Laplace v MH



Laplace approximation (left) allows some <u>bad</u> boundaries

Laplace v 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?