Logistic Regression lab

Simon Rogers, Feb 2017

Introduction and aims

Your task in this lab is to implement a Metropolis-Hastings sampler for logistic regression. To make it feasible in an hour, I have provided a skeleton notebook that generates data and makes plots of the decision boundaries etc. You just need to add some Metropolis code.

Task

- Recall that the two steps of Metropolis are to *propose* a new value and then either accept or reject
- To propose, generate a new $\mathbf{w}$ vector from a Gaussian with mean equal to the old value and diagonal covariance (use the `jump_sig` value in the notebook as the proposal standard deviation). To do this generate two Gaussian random variables (`np.random.randn()*jump_sig`) and add one to each dimension of the current $\mathbf{w}$.
- If we use $\mathbf{w}^*$ to denote the new value, compute the Metropolis ratio:

$$r = \frac{p(\mathbf{w}^*)}{p(\mathbf{w})} \times \frac{L(\mathbf{w}^*)}{L(\mathbf{w})}$$

- where p() and L() are the likelihood and prior (functions to compute the log of these provided in the notebook).
- I recommend you compute the ratio in log space and, once you have all the terms in, exponentiate. This is more numerically stable.
- Now generate a random value between 0 and 1 (`np.random.rand()`) and, if the value if less than r, accept. Otherwise reject.
- Remember that if you reject you should store the old value again in your list of samples. I.e. 1000 iterations should result in 1000 w values, although some will be identical to the previous one.
- To store the samples I recommend: `w_samples.append(w.flatten())`. Then, once you have them all use `w_samples = np.array(w_samples)` to make something that will work with my plot code.