# MLE, MAP, and Naive Bayes

Suppose we are given a dataset Xof outcomes from some distribution parameterized by \Theta. How do we estimate \Theta?

For example, given a bent coin and a series of heads and tails outcomes from that coin, how can we estimate the probability of the coin landing heads?

So, we have:

X= our dataset

\Theta= unknown parameter (or parameter vector) =Y

p(\Theta)= our prior

p(\Theta|X)= our posterior distribution

p(X|\Theta)= our likelihood

A quick note on terminology: the term ‘probability’ is usually used to talk about the possible outcomes of data given some parameter, while the term ‘likelihood’ is used to talk about the possible values of a parameter given some data. In both cases we’re working with a function of two variables: holding the data constant gives you the likelihood function while holding the parameter constant gives you the probability function.

Probability is used before data are available to describe possible future outcomes given a fixed value for the parameter (or parameter vector). *Likelihood* is used after data are available to describe a function of a parameter (or parameter vector) for a given *outcome.*

**Maximum Likelihood Estimation (MLE)**

Maximum Likelihood asks: for a given dataset, what is the value of \Theta that maximizes the probability of the observed dataset? Trivially, in our bent coin example if we observe 30 H outcomes and 70 T outcomes, the value of \Theta that maximizes the probability of seeing our observed dataset is p(H)= .3, i.e. our bent coin has a 30% chance of coming up heads. But how do we get there?

We would like to find the value of \Thetafor which the observed dataset Xis most probable under some distribution p:

\arg\max_\theta  p(X|\Theta) = \arg\max_\theta  p(x_1,x_2...,x_n|\Theta)

Assuming independence between observations in our dataset, we get:

\displaystyle \prod_{i=1}^n p(x_{i}|\Theta)

And after taking the log (for numerical underflow and ease of computation; also, taking derivative of sums is easier than derivative of products) we get the likelihood function \mathcal{L}:

\mathcal{L}(X|\Theta) =  \displaystyle \sum_{i=1}^n log p(x_{i}|\Theta)

Finally, since we want maximize p(X|\Theta)we set the derivative wrt \Thetato zero:

\frac{\partial \mathcal{L}}{\partial \Theta}  = 0

**Maximum A Posteriori Estimation (MAP)**

MAP asks the same question as MLE, but lets us inject a prior distribution (as well as our “confidence” in that distribution) into our estimation, allowing us to essentially create a regularized a posteriori distribution. For example, taking up another coin example, we may take a coin out of our pocket that looks fair, but over 4 trials produces 1 heads and 3 tails. According to MLE, the best value of \Thetafor this coin is p(H)=.25. And yet, this is not a good estimate for the coin that looks fair: we know that it is entirely possible for a fair coin to give a series of outcomes that looks like this. MAP lets us regularize the estimation of \Thetaso that we may say something like this: we believed the coin to be around 50/50 and it may well be a fair coin, but given our new dataset of 4 flips, let’s combine the prior distribution with the observed data to come up with some new estimation of \Thetathat lies somewhere inbetween the prior and MLE of the new dataset.

While in MLE we looked at \arg\max_\theta p(X|\Theta), in MAP we look at \arg\max_\theta p(\Theta|X).

Remember Bayes’ Theorem?

p(\Theta|X) = \displaystyle \frac {p(X|\Theta) p(\Theta)}{p(X)}

Notice that p(X)in the denominator is a constant value (used in Baye’s Rule to normalize results), so we don’t need it when estimating \Theta.  This gives us:

\arg\max_\theta p(X|\Theta) p(\Theta)

As mentioned above, the difference between this and MLE is the presence of the prior. If we expand this equation as we did with MLE and take the log, we get our likelihood function \mathcal{L}:

\mathcal{L}(X|\Theta) =  \displaystyle \sum_{i=1}^n log p(x_{i}|\Theta) + log p(\Theta)

The prior in this case will essentially pull our estimation towards the prior value of \Theta. Note that when p(\Theta)is uniform, MLE and MAP are equivalent, and as the number of observations in Xincreases, MLE and MAP converge. The prior term in MAP has less and less of an effect on the estimation \Thetaas the amount of data in Xincreases, and intuitively this makes sense – the more data we have, the more reason we have to adjust our prior beliefs to fit the new data.

**Naive Bayes Classifier**

MAP serves as the basis of a Naive Bayes Classifier. Let’s assume that we now have not just one parameter determining the outcome of our random variable, but a multitude. Extending our coin flip example, instead of outcomes just depending on the bendiness of the coin, we now model the outcome of H and T as a function of coin bendiness, wind, surface angle of the floor, elasticity of floor material, height of coin toss, gravitational force…you name it. Now, what’s the probability of our coin landing H given our vector of n predictive features \Theta_{1},...\Theta_{n}?

Now we make the **naive** assumption that these features are all independent from one another, e.g. the conditional relationship between any given feature and the coin landing H is not influenced by any of the other features:

p(\Theta_{i}|H, \Theta_{1},...,\Theta_{i-1}, \Theta_{i+1},...\Theta_{n}) = p(\Theta_{i}|H)

And now our equation simplifies to:

Given that p(\Theta_{1},...\Theta_{n})is a constant, we’re back to MAP!

\arg\max_H p(H) \displaystyle \prod_{i=1}^n p(\Theta_{i}|H)

where the prior p(H) is just the relative class frequency in our dataset or, in our example, the number of H outcomes.