MLE, MAP, AND NAIVE BAYES

10-601 RECITATION

MARY MCGLOHON

MLE

- The usual representation we come across is a **probability density function:** $P(X = x|\theta)$
- But what if we know that $x_1, x_2, ..., x_n \sim N(\mu, \sigma^2)$, but we don't know μ?
- We can set up a **likelihood** equation: $P(\mathbf{x}|\mu, \sigma)$, and find the value of μ that **maximizes** it.

MLE OF MU

- Since x's are independent and from the same distribution,

$$p(\mathbf{x}|\mu,\sigma^2) = \prod_{i=1}^{n} p(x_i|\mu,\sigma^2)$$

$$L(\mathbf{x}) = \prod_{i=1}^{n} p(x_i|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \prod_{i=1}^{n} exp(\frac{-(x_i-\mu)^2}{2\sigma^2})$$

- Taking the log likelihood (we get to do this since log is monotonic) and removing some constants:

$$log(L(\mathbf{x})) = l(\mathbf{x}) \propto \sum_{i=1}^{n} -(x_i-\mu)^2$$

CALCULUS!

- We can take the derivative of this value and set it equal to zero, to maximize.

$$\frac{dl(x)}{dx} = \frac{d}{dx}(-\sum_{i=1}^{n} x_i^2 - x_i\mu + \mu^2) = -\sum_{i=1}^{n} x_i - \mu$$

$$-\sum_{i=1}^{n} x_i - \mu = 0 \rightarrow n\mu = \sum_{i=1}^{n} x_i \rightarrow \mu = \frac{\sum_{i=1}^{n} x_i}{n}$$

ABOUT THE MLE

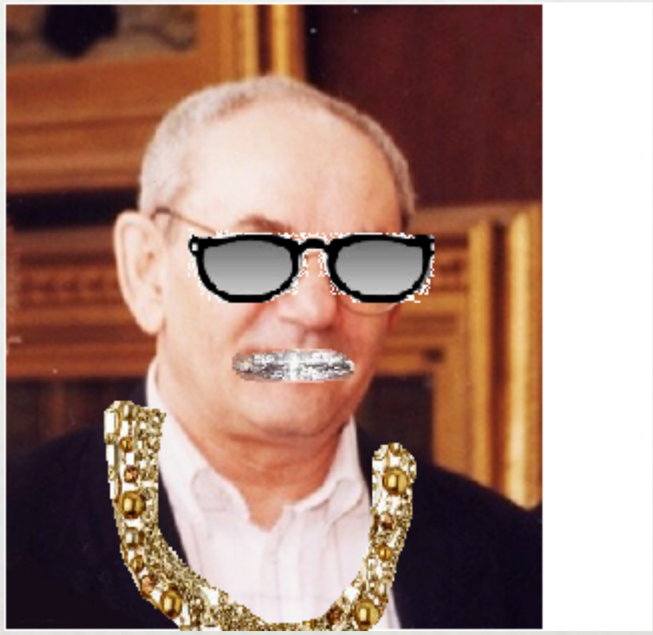
- We can't always do this analytically, but there are all sorts of tricks. (See homework).
- This is the traditional statistical approach to finding parameters.

(The Unbiased M.L.E.)

ABOUT THE MLE

- We can't always do this analytically, but there are all sorts of tricks. (See homework).
- This is the traditional statistical approach to finding parameters.

(The Unbiased M.L.E.)



MAP

- What if you have some ideas about your parameter?
- In the Bayesian school of thought (or “cult”, depending on who you ask)...
- We can use Bayes’ Rule:

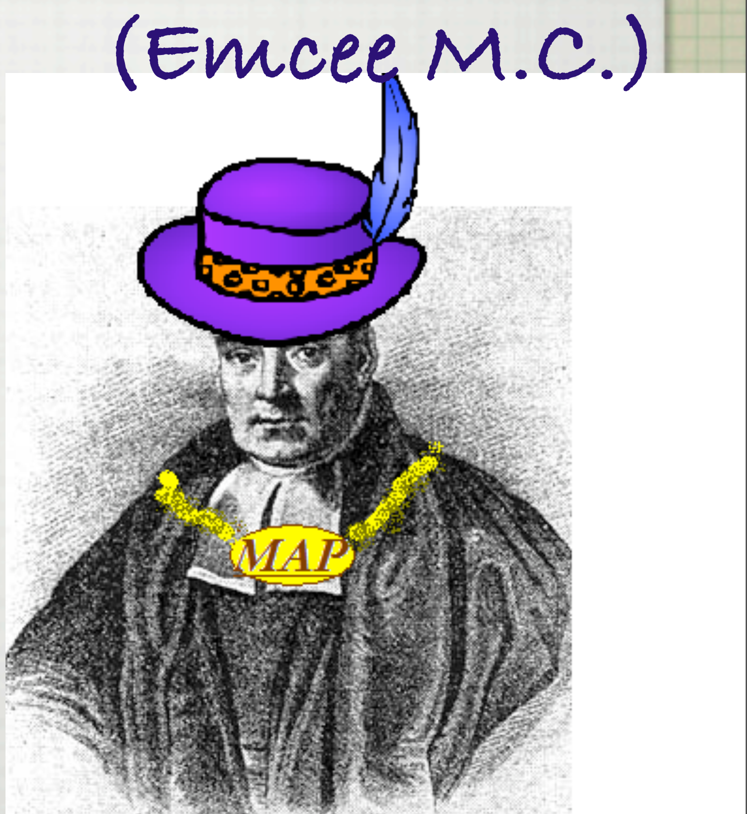
$$P(\theta|\mathbf{x}) = \frac{P(\mathbf{x}|\theta)P(\theta)}{P(\mathbf{x})} = \frac{P(\mathbf{x}|\theta)P(\theta)}{\sum_{\Theta} P(\theta, \mathbf{x})} = \frac{P(\mathbf{x}|\theta)P(\theta)}{\sum_{\Theta} P(\mathbf{x}|\theta)P(\theta)}$$

MAP

- $$argmax_\theta P(\theta|\mathbf{x}) = argmax_\theta \frac{P(\mathbf{x}|\theta)P(\theta)}{\sum_\Theta P(\mathbf{x}|\theta)P(\theta)}$$
- This is just maximizing the numerator, since the denominator is a normalizing constant.
- This assumes a **prior** distribution, $P(\theta)$. (Emcee M.C.)
- Old-school statisticians hate that. But if you get a good estimate of the prior, you'll probably be ok.
- (This is why we don't have old-school statisticians writing spam filters.)

MAP

- $$argmax_{\theta} P(\theta|\mathbf{x}) = argmax_{\theta} \frac{P(\mathbf{x}|\theta)P(\theta)}{\sum_{\Theta} P(\mathbf{x}|\theta)P(\theta)}$$
- This is just maximizing the numerator, since the denominator is a normalizing constant.
- This assumes a **prior** distribution, $P(\theta)$.
- Old-school statisticians hate that. But if you get a good estimate of the prior, you'll probably be ok.
- (This is why we don't have old-school statisticians writing spam filters.)



WHAT WE CAN DO NOW

- MAP is the foundation for Naive Bayes classifiers.
- Here, we're assuming our data are drawn from two "classes". We have a bunch of data where we know the class, and want to be able to predict P(class|data-point).
- So, we use empirical probabilities

$$prediction = argmax_C P(C = c|X = x) \propto argmax_C \hat{P}(X = x|C = c)\hat{P}(C = c)$$

- In NB, we also make the assumption that the features are **conditionally independent**.

SPAM FILTERING

- Suppose we wanted to build a spam filter. To use the "bag of words" approach, assuming that *n* words in an email are **conditionally independent,** we'd get:

$$P(spam|\mathbf{w}) \propto \prod_{i=1}^{n} \widehat{P}(w_i|spam)\widehat{P}(spam)$$

$$P(\neg spam|\mathbf{w}) \propto \prod_{i=1}^{n} \widehat{P}(w_i|\neg spam)\widehat{P}(\neg spam)$$

- Whichever one's bigger wins!

THE IMPORTANCE OF THE SAMPLE

- What happens if you train on a set of data that's mostly spam, and test on a set that's mostly good emails?

$$P(spam|\mathbf{w}) \propto \prod_{i=1}^{n} \widehat{P}(w_i|spam)\widehat{P}(spam)$$

- Also, just choosing one test set "wastes data".

- What can we do?

CROSS-VALIDATION

- Cross-validation involves training several times.
- LOOCV (Leave-one-out cross validation):
  - For each data point, train classifier on x_{-i} -- that is, all data points besides x_i, and classify x_i.
  - Error is the average accuracy.
  - What's wrong with this?

K-FOLDS CV

- A cheaper way of doing cross-validation is to divide ("fold") the dataset into *k* pieces.
- For each piece *i*,
  - Train on all data not in set *i*, classify set *i*.
  - Report mean error.
- This is a "happy medium" between straight-up training/testing and LOOCV.

LESS-NAIVE BAYES

- How would we modify NB to use two dependent attributes?
  - Hint: $P(x_i|c) = P(x_{i,1}, x_{i,2}, ..., x_{i,A}|c)$ -- you're estimating the joint conditional distribution of the attributes.

CONTINUOUSLY NAIVE BAYES

- How could we modify NB for continuous attributes?
  - For instance, classify whether you like basketball given your age (real), height (real), whether you like football (binary), and type of shoes you wear (categorical).

TOPOLOGY OF NAIVE BAYES

- What's the decision surface for Naive Bayes?
  - Hint:

$$P(c|word) = P(word|c)I(word) + P(\neg word|c)I(\neg word)$$