Bayesian Analysis for Audio Engineers

Part I: Conditional Probability, Bayes' Rule, & Naive Bayes Classification

(with some WebAudio thrown in for fun)

Scott H. Hawley, Ph.D.
Belmont University

Motivation

- Many newer products hitting the audio market are employing 'intelligent' algorithms for things like
  - Source separation (e.g., signal from noise, drum mic leakage suppression, vocal correction, ...sweet Karaoke)
  - Data Compression
  - Characterization / Classification (e.g., Pandora)
- Me: I wanted to write a little toy in WebAudio that would 'slurp' in a song and split off the instrument tracks and let me move 'em around in time & frequency. (= lots of math!)

Demos: WebAudio

- HTML5 'standard.' Runs in the browser
- Google, Mozilla major developers
  - Best to use Chrome or Safari browsers.
- Applications: Gaming, Recording, "Full Platform"
- Features include: multi-tracking, effects chains, convolution, positional audio
- Demo Links:
  - Chromium:
    - GUI Demos: https://chromium.googlecode.com/svn/trunk/samples/audio/samples.html
    - Convolution: https://chromium.googlecode.com/svn/trunk/samples/audio/convolution-effects.html
  - My "DumbDaw": http://hedges.belmont.edu/~shawley/dumbdaw

"A.I."

"Come with me if you want to live."

- 'Intelligent' can imply
  - Based on cognitive / behavioral models, or
  - 'Machine learning' algorithms
  - Prominent class of these are 'Bayesian'. Classic example: Spam Filter

Proverbs 22:6 (Paraphrased):
"Train up a Bayesian filter in the way it should go, and after many iterations it will not depart from it."*

**unless it's running a Markov Chain Monte Carlo scheme, in which case the probability is small but finite that it will 'depart from it.'*

Audio "A.I."

- Notable "pattern matching" application to audio: Blind Source Separation via 'either' Sparse Coding, Bayesian Analysis, or Non-Negative Matrix Factorization (NMF).
- Demo: DrumAtom plugin (uses NMF):
  https://www.youtube.com/watch?v=oSlSqIBAw7U
- Bayesian Approach used with vocals, gravitational waves,…and LOTS of other fields. Decided to force myself to learn it.

"Bayes-ian" Statistics

- Deals with assigning probabilities to relationships between data
- Credited to Thomas Bayes (1701-1761) English statistician, philosopher and Presbyterian minister
- Applications:

  Spam filtering Medical Diagnosis Setting your insurance rates
  Forensics
  Feature extraction from images Findin' them terr'rists
  Predicting "what else you'll like"
  Locating enemy submarines (Alan Turing, WWII - classified!)
  Forecasting elections
  Extracting gravity wave signals (LIGO) ...and much, much more!

- Relies on the concept of "conditional probability"...

Conditional Probability, p.1

- You see someone new and you wonder if he* is an audio engineer. How to decide?
- What are the odds that a 'random' person from the general population will be an audio engineer?...
- Call this P(A), where "A" stands for "is an audio engineer" and "P()" stands for "Probability of" (or "Percentage of")

$$P(A) = (\text{\# of audio engineers}) / (\text{\# of population})$$

*Oh, calm down; we're only choosing among men for grammatical and algorithmic simplicity. Female engineers can rock too.

Conditional Probability, p.2

- What are some identifying characteristics of audio engineers?

Black T-shirt

Geeeear!!

Pony Tail

Headphones

- What If I now told you this 'random person' has a pony tail? Would you think it more or less likely that he's an audio engineer (than you did without this added information)?

- This is the essence of **conditional probability**. Let "T" denote "has a pony tail", then

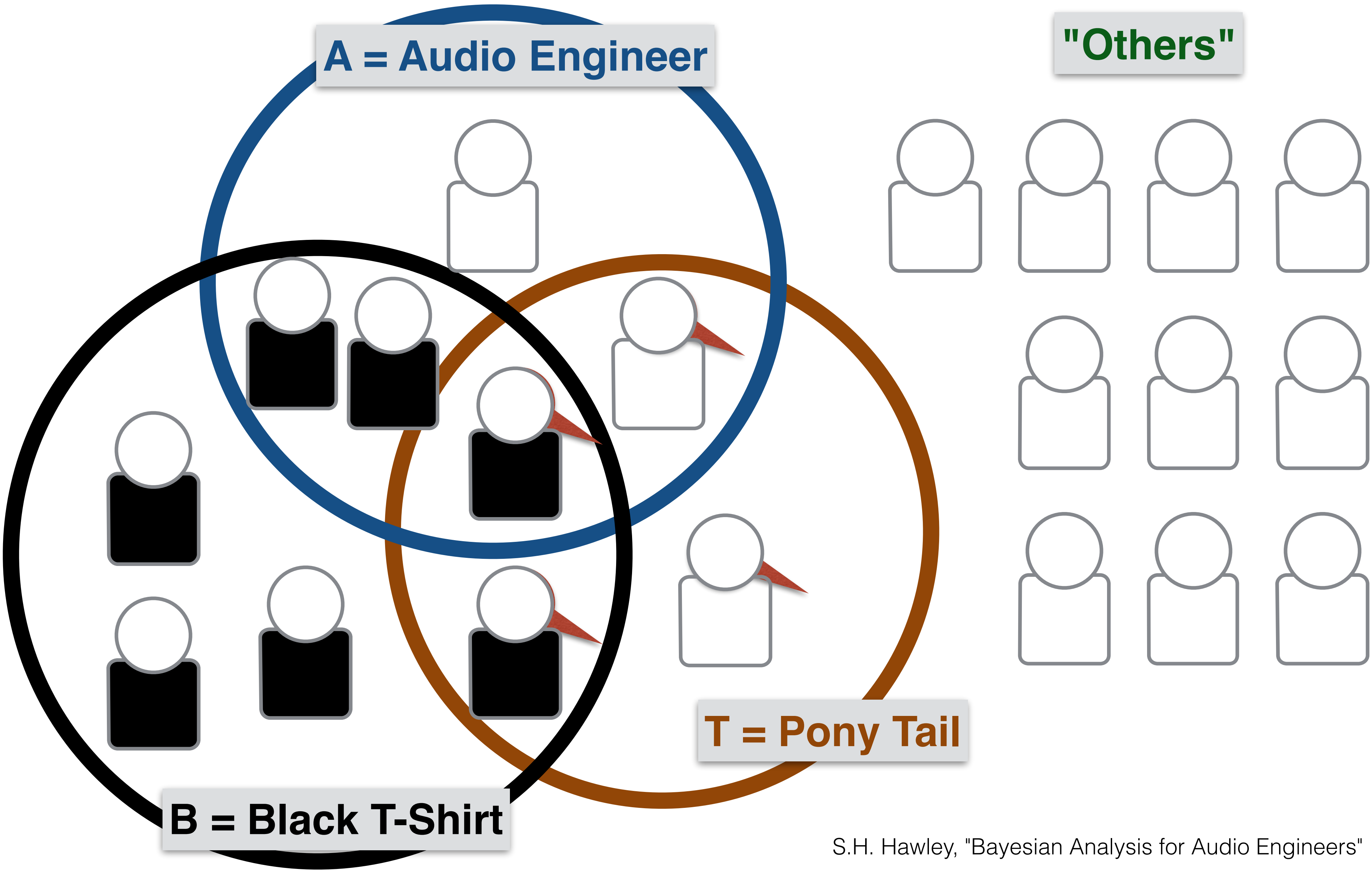
$$P(A \mid T)$$

is the probably of A, given T; so the "|" means "given".
(You can also read it as "Percentage of T's that are A," as in "Percentage of men who are audio engineers among the population of men with pony tails.")

Conditional Probability, p.3

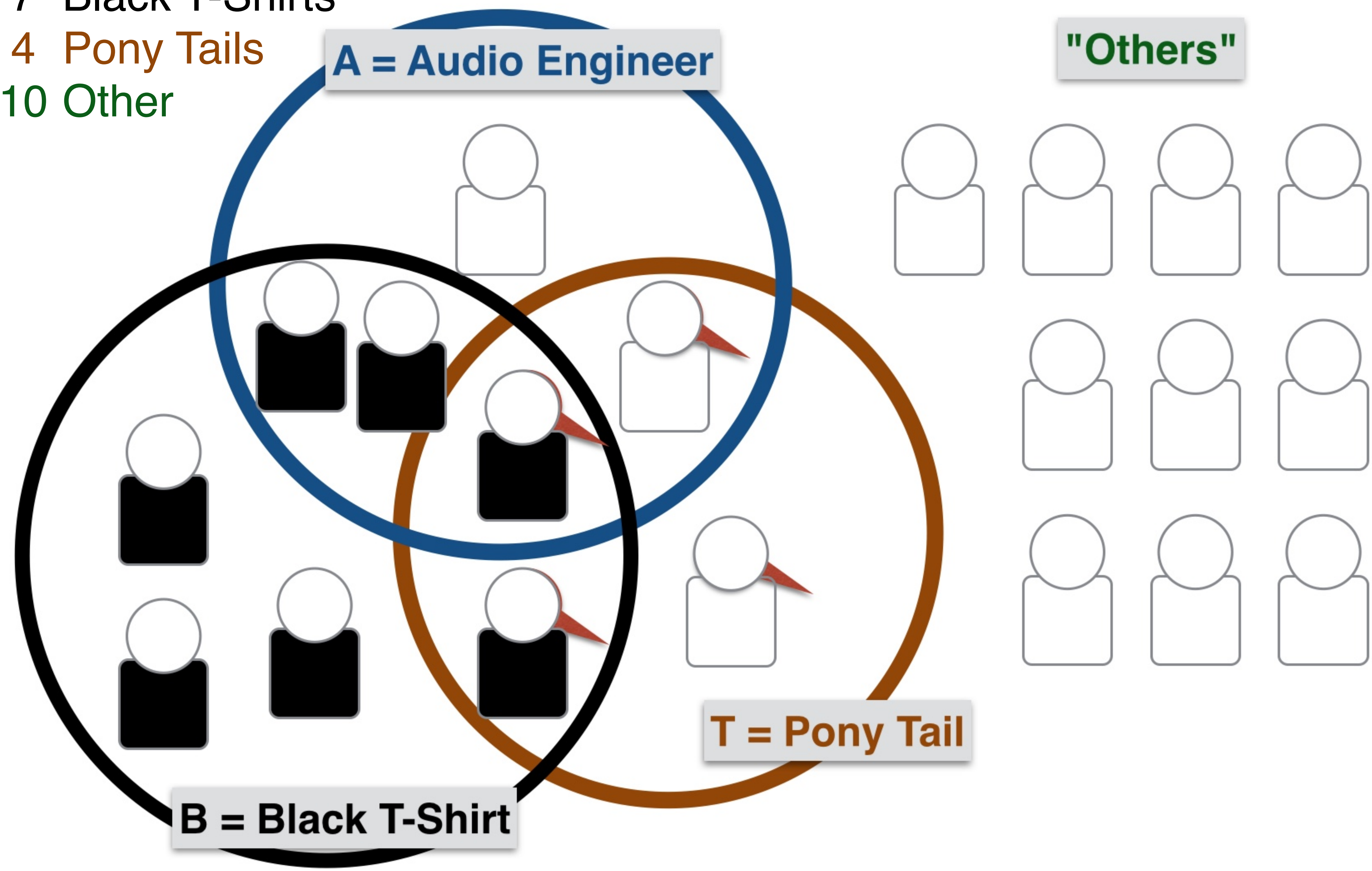
- What if the person has a pony tail *and* a black T-shirt?
- ...pony tail, black T-shirt, headphones, and you're at a conference for audio engineers? ;-)
- To illustrate this estimation process, let's do an example...

"Is that dude an audio engineer?"

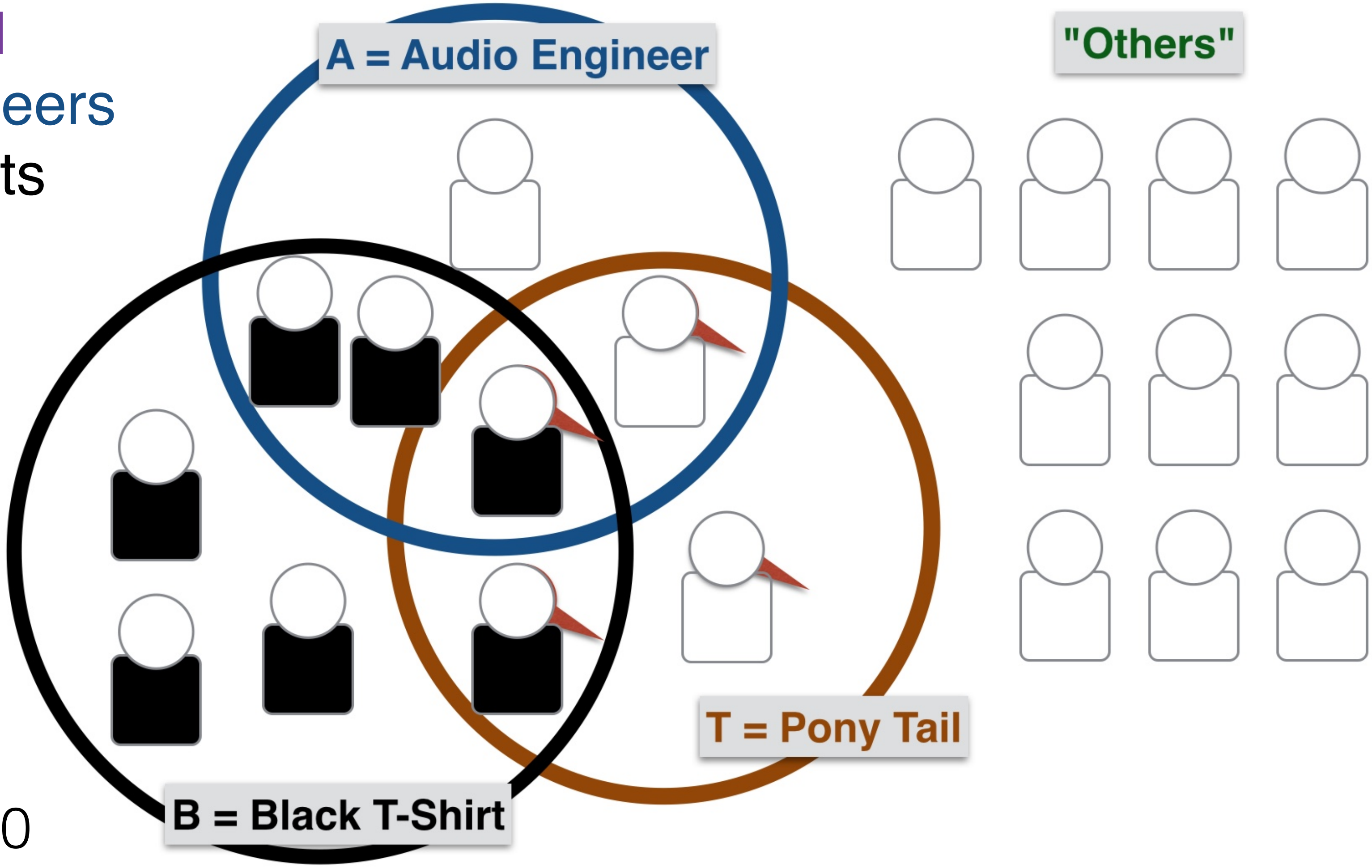


S.H. Hawley, "Bayesian Analysis for Audio Engineers"

20 People Total
5 Audio Engineers
7 Black T-Shirts
4 Pony Tails
10 Other



20 People Total
5 Audio Engineers
7 Black T-Shirts
4 Pony Tails
10 Other



General Stats:

P(A) = 5/20

P(B) = 7/20

P(T) = 4/20

P(A&B) = 3/20

P(A&T) = 2/20

P(A&B&T) = 1/20

Some Conditional Stats:

P(A|T) = 2/4 = 10/20, P(A|B) = 3/7 = 8.6/20

P(T|A) = 2/5 = 8/20, P(B|A) = 3/5 = 12/20

P(A| B&T) = 1/2 = 10/20 = P(A|T) => Including B given T has no effect on P!

Some of these results (and more) can be derived from others.
This is where Bayes' Rule comes in!...

Bayes' Rule

- ...which is really just common sense, says that

$$P(X|Y) = P(X\&Y) / P(Y),$$

or put differently,

$$P(X\&Y) = P(X|Y)\,P(Y)$$
$$= P(Y|X)\,P(X)$$

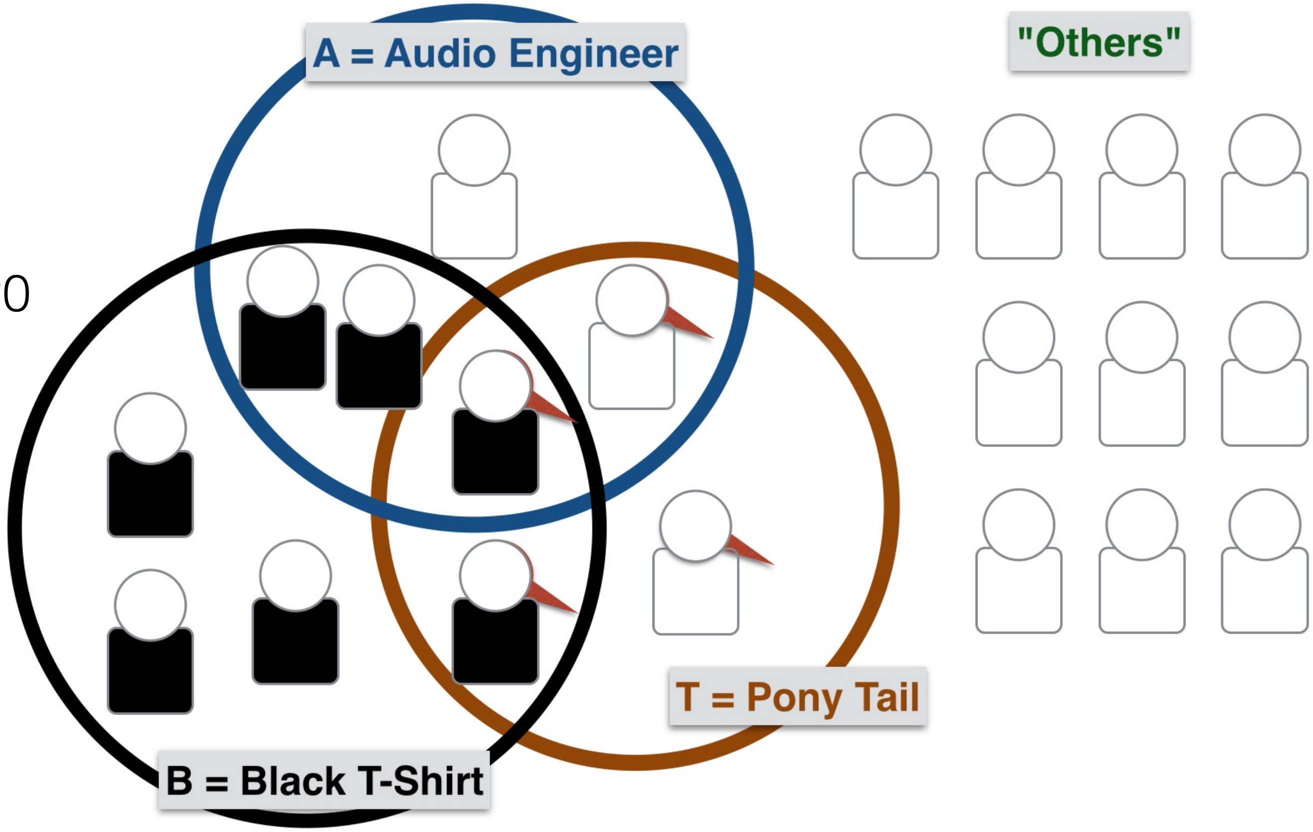
(because $P(X\&Y) = P(Y\&X)$).

- Before we explore the surprising implications of this, let's check it against our example...

From Earlier:

General Stats:
P(A) = 5/20, P(B) = 7/20
P(A&B) = 3/20

Conditional Stats:
P(A|B) = 3/7
P(B|A) = ???



Now, as an example, use Bayes' Rule to calculate P(A|B):

$P(A|B) = P(A\&B) / P(B) = (3/20) / (7/20) = 3/7$ -- yep, it works!

What about P(B|A)? Let's get a little fancier with Bayes' Rule:

$P(B|A) = P(A\&B) / P(A) = [P(A|B) P(B)] / P(A) = (3/7)(7/20) / (5/20) = 3/5$, which we can verify by looking at the picture above.

The real power of Bayes' Rule shines in cases where we lack certain pieces of information and need to infer them from others...

Bayesian Inference Example

"How many audio engineers are there in Nashville?"

- The total population of Nashville is N=600,000. If you knew what P(A) (=% of audio engineers) is, then you could use P(A)*N to get the number of engineers. But say you don't have P(A); instead the information you have is (making this up)…
  - A survey from the AES reporting that 3/5 of audio engineers wear black T-shirts, i.e. P(B|A) = 0.6
  - A census of black-T-shirt-wearers (BTSWs) reporting that BTSWs make up 2% of the Nashville population: P(B) = 0.02,
  - ...and that 1/8 of BTSWs are audio engineers: P(A|B) = 0.125
- Use Bayes' Rule ('inverted') to estimate P(A):
  Since P(B|A)=P(A&B) / P(A), then

$$P(A) = P(A\&B) / P(B|A) = [P(A|B)\, P(B)] / P(B|A)$$

i.e., $P(A) = [\, 0.125 * 0.02 \,] / 0.6 = 0.004166\ldots$

=> P(A)*N = (0.004166..)(600,000) = 2500 Audio Engineers in Nashville

Jargon

- A probability function is called a **distribution**, which can be over discrete or continuous variable(s)
  - Discrete: individual people, as in our example with audio engineers
  - Continuous: e.g., probability that the Comcast guy will show up, vs. how long you wait around the house (variable=time).

P(class | observations) = P(obs. | class) P(class) / P(obs.)

"Posterior" "Likelihood" "Priors"

- Much of Bayesian analysis is about using what you already know to estimate probabilities of things you don't (yet) know, i.e., in *finding posterior distributions*.*
- In this way, Bayesian analysis is a suitable mathematical model for decision-making in "real life" -- e.g., "Should I go to this party or will it be lame?"

*Insert "Pick-up Lines for Bayesian Mathematicians"

Inference ->'Learning'?

- Is this inference really 'learning'? Barely; not in the sense of most 'machine learning' algorithms which make use of Bayesian concepts. We're not there yet! Next slide…
- Pause: Summing up so far:
  - Conditional Probability: More info can produce different estimates (by changing the sample you're "pulling from")
  - Bayes' Rule: $P(B|A) = P(A|B)\, P(B) / P(A)$
    - Handy 'mnemonic': $P(A\&B) = P(A|B)\, P(B) = P(B|A)\, P(A)$
  - Bayes' Rule can be used to infer info you may not have direct access to.

“Naive Bayes Classifier”

*Following YouTube videos by Srinath Sridhar: https://www.youtube.com/watch?v=mW9CHXNhwng

- Methodology:
  - Scape a bunch of urls from web-search results using “class” names: “audio engineer” “hippie” “goth” “producer” “musician” “metalhead”

    …supplies P(class). Use urls, scrape words from pages.
  - Tabulate histogram of each word across web pages, given each class. This is $P(\text{word}_i \mid \text{class})$. Also get probability of each word across across all classes: $P(\text{word}_i)$.
  - Apply Bayes’ Rule to, given a new web page, estimate which “class” the page is in: $P(\text{class} \mid \text{word}_1 \,\&\, \text{word}_2 \ldots)$
    - “Naive”: Assume words are “independent”, i.e.

      $P(\text{word}_1 \,\&\, \text{word}_2 \,\&\, \text{word}_3 \ldots) = P(\text{word}_1) * P(\text{word}_2) * P(\text{word}_3) \ldots$

- Live Demo!

  Code here: http://hedges.belmont.edu/~shawley/Bayesian_Slides_Code.tar.gz

Naive Bayes->Machine Learning

- “Semi-Supervised Learning”: When class distributions are “well separated”, can use build training set incrementally from testing set
- Markov-Chain Monte Carlo: Too much data to crunch, so you “sample” the system cleverly
  - Build a “Bayesian Network” to enumerate priors & dependencies, regard this as a “Markov Chain”, sample via “Gibbs Sampling”, traverse via “Monte Carlo”
  - Monte Carlo methods popularized via particle physicists.
  - e.g. Audio source-separation works by Fevotte: http://www.unice.fr/cfevotte/publications/chapters/bass.pdf
- …and we’re out of time!

Questions?