Lecture 11: Privacy and Anonymity
Privacy and Anonymity

• Being on-line without giving up everything about you
• Ensuring collected data doesn’t reveal its users data
• Privacy in
  • Structured Data: k-anonymity, differential privacy
  • Text: obfuscating authorship
  • Speech: speaker id and de-identification
Companies Getting Your Data

- They actually don’t want your data, they want to upsell
  - They want to be able to do tasks (recommendations)
  - They actually don’t care about the individual you
- Can they process data to never have identifiable content
  - Cumulated statistics
  - Averages, counts, for classes
- How many examples before it is anonymous
k-anonymity

- Latanya Sweeney and Pierangela Samarati 1998
- Given some table for data with features and values
- Release data that guarantees individuals can’t be identified
  - **Suppression**: Delete entries that are too “unique”
  - **Generalization**: relax specificness of fields,
    - e.g. age to age-range or city to region
### k-anonymity

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Gender</th>
<th>State of domicile</th>
<th>Religion</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ramsha</td>
<td>29</td>
<td>Female</td>
<td>Tamil Nadu</td>
<td>Hindu</td>
<td>Cancer</td>
</tr>
<tr>
<td>Yadu</td>
<td>24</td>
<td>Female</td>
<td>Kerala</td>
<td>Hindu</td>
<td>Viral infection</td>
</tr>
<tr>
<td>Salima</td>
<td>28</td>
<td>Female</td>
<td>Tamil Nadu</td>
<td>Muslim</td>
<td>TB</td>
</tr>
<tr>
<td>Sunny</td>
<td>27</td>
<td>Male</td>
<td>Karnataka</td>
<td>Parsi</td>
<td>No illness</td>
</tr>
<tr>
<td>Joan</td>
<td>24</td>
<td>Female</td>
<td>Kerala</td>
<td>Christian</td>
<td>Heart-related</td>
</tr>
<tr>
<td>Bahuksana</td>
<td>23</td>
<td>Male</td>
<td>Karnataka</td>
<td>Buddhist</td>
<td>TB</td>
</tr>
<tr>
<td>Rambha</td>
<td>19</td>
<td>Male</td>
<td>Kerala</td>
<td>Hindu</td>
<td>Cancer</td>
</tr>
<tr>
<td>Kishor</td>
<td>29</td>
<td>Male</td>
<td>Karnataka</td>
<td>Hindu</td>
<td>Heart-related</td>
</tr>
<tr>
<td>Johnson</td>
<td>17</td>
<td>Male</td>
<td>Kerala</td>
<td>Christian</td>
<td>Heart-related</td>
</tr>
<tr>
<td>John</td>
<td>19</td>
<td>Male</td>
<td>Kerala</td>
<td>Christian</td>
<td>Viral infection</td>
</tr>
</tbody>
</table>

• From wikipedia: K-anonymity
## k-anonymity

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Gender</th>
<th>State of domicile</th>
<th>Religion</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>20 &lt; Age ≤ 30</td>
<td>Female</td>
<td>Tamil Nadu</td>
<td>*</td>
<td>Cancer</td>
</tr>
<tr>
<td>*</td>
<td>20 &lt; Age ≤ 30</td>
<td>Female</td>
<td>Kerala</td>
<td>*</td>
<td>Viral infection</td>
</tr>
<tr>
<td>*</td>
<td>20 &lt; Age ≤ 30</td>
<td>Female</td>
<td>Tamil Nadu</td>
<td>*</td>
<td>TB</td>
</tr>
<tr>
<td>*</td>
<td>20 &lt; Age ≤ 30</td>
<td>Male</td>
<td>Karnataka</td>
<td>*</td>
<td>No illness</td>
</tr>
<tr>
<td>*</td>
<td>20 &lt; Age ≤ 30</td>
<td>Female</td>
<td>Kerala</td>
<td>*</td>
<td>Heart-related</td>
</tr>
<tr>
<td>*</td>
<td>20 &lt; Age ≤ 30</td>
<td>Male</td>
<td>Karnataka</td>
<td>*</td>
<td>TB</td>
</tr>
<tr>
<td>*</td>
<td>Age ≤ 20</td>
<td>Male</td>
<td>Kerala</td>
<td>*</td>
<td>Cancer</td>
</tr>
<tr>
<td>*</td>
<td>20 &lt; Age ≤ 30</td>
<td>Male</td>
<td>Karnataka</td>
<td>*</td>
<td>Heart-related</td>
</tr>
<tr>
<td>*</td>
<td>Age ≤ 20</td>
<td>Male</td>
<td>Kerala</td>
<td>*</td>
<td>Heart-related</td>
</tr>
<tr>
<td>*</td>
<td>Age ≤ 20</td>
<td>Male</td>
<td>Kerala</td>
<td>*</td>
<td>Viral infection</td>
</tr>
</tbody>
</table>

- From wikipedia: K-anonymity
But if X is in the dataset you do know they have a disease
You can set “k” to something though to be unique enough
Making a dataset “k-anonymous” is NP-Hard
But it is a measure of anonymity for a data set
Is there a better way to hide identification?
Differential Privacy

- Maximize statistical queries, minimize identification
- When asked about feature x for record y
  - Toss a coin: if heads give right answer
  - If tails: throw coin again, answer yes if heads, no if tails
- Still has accuracy at some level of confidence
- Still has privacy at some level of confidence
Authorship Obfuscation

- Remove most identifiable words/n-grams
  - “So” → “Well”, “wee” -> “small”, “If its not too much trouble” → “do it”
- Reddy and Knight 2016
  - Obfuscating Gender in Social Media Writing
  - “omg I’m soooo excited!!!”
  - “dude I’m so stoked”
## Authorship Obfuscation

- Most gender related words (Reddy and Knight 16)

<table>
<thead>
<tr>
<th></th>
<th>Twitter</th>
<th>Yelp</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Male</strong></td>
<td>bro, bruh, game, man, team, steady, drinking, dude, brotha, lol</td>
<td>wifey, wifes, bachelor, girlfriend, proposition, urinal, oem</td>
</tr>
<tr>
<td><strong>Female</strong></td>
<td>my, you, me, love, omg, boyfriend, miss, mom, hair, retail</td>
<td>corvette, wager, fairways, urinals, firearms, diane, barbers</td>
</tr>
<tr>
<td><strong>Male</strong></td>
<td>hubby, boyfriend, hubs, bf, husbands, dh, mani/pedi, boyfriends</td>
<td>bachelorette, leggings, aveda, looooove, yummy, xoxo, pedi, bestie</td>
</tr>
</tbody>
</table>

### Notes:
- The table shows a comparison of gender-related words in Twitter and Yelp, highlighting the differences in language use between males and females.
Authorship Obfuscation

• Learning substitutions
  • Mostly individual words/tokens
  • Spelling corrections “goood” → “good”
  • Slang to standard “buddy” → “friend”
  • Changing punctuation

• But
  • Although it obfuscates, a new classifier might still identify differences
  • It really only does lexical substitutions (authorship is more complex)
Speaker ID

- Your speech is as true as a photograph
- Synthesis can (often) fake your voice
- Court case authentication
  - (usually poor recording conditions)
  - Human experts vs Machines
- Probably records exist for all your voices
Who is speaking?

- Speaker ID, Speaker Recognition
- When do you use it
  - Security, Access
  - Speaker specific modeling
    • Recognize the speaker and use their options
  - Diarization
    • In multi-speaker environments
    • Assign speech to different people
    • Allow questions like did Fred agree or not.
Voice Identity

• What makes a voice identity
  - Lexical Choice:
    • Woo-hoo,
    • I’ll be back ...
  - Phonetic choice
  - Intonation and duration
  - Spectral qualities (vocal tract shape)
  - Excitation
Voice Identity

• What makes a voice identity
  - Lexical Choice:
    • Woo-hoo,
    • I’ll be back …
  - Phonetic choice
  - Intonation and duration
  - Spectral qualities (vocal tract shape)
  - Excitation
• But which is most discriminative?
• GMM Speaker ID

• Just looking at spectral part
  - Which is sort of vocal tract shape

• Build a single Gaussian of MFCCs
  - Means and Standard Deviation of all speech
  - Actually build N-mixture Gaussian (32 or 64)

• Build a model for each speaker

• Use test data and see which model its closest to
• GMM Speaker ID

• How close does it need to be?
  - One or two standard deviations?

• The set of speakers needs to be different
  - If they are closer than one or two stddev
  - You get confusion.

• Should you have a “general” model
  - Not one of the set of training speakers
GMM Speaker ID

- Works well on constrained tasks
  - In similar acoustic conditions
  - (not telephone vs wide-band)
  - Same spoken style as training data
  - Cooperative users

- Doesn’t work well when
  - Different speaking style (conversation/lecture)
  - Shouting whispering
  - Speaker has a cold
  - Different language
Speaker ID Systems

Training
- Example speech from each speaker
- Build models for each speaker
- (maybe an exception model too)

ID phase
- Compare test speech to each model
- Choose “closest” model (or none)
• Basic Speaker ID system
• Accuracy

• Works well on smaller sets
  - 20-50 speakers

• As number of speakers increase
  - Models begin to overlap – confuse speakers

• What can we do to get better distinctions
• What about transitions

• Not just modeling isolated frames
• Look at phone sequences
• But ASR
  - Lots of variation
  - Limited amount of phonetic space
• What about lots of ASR engines
• Phone-based Speaker ID

• Use *lots* of ASR engines
  - But they need to be different ASR engines
• Use ASR engines from lots of different languages
  - It doesn’t matter what language the speech is
  - Use many different ASR engines
  - Gives lots of variation
• Build models of what phones are recognized
  - Actually we use HMM states not phones
Phone-based SID (Jin)
• Phone-based Speaker ID

• Much better distinctions for larger datasets
• Can work with 100 plus voices
• Slightly more robust across styles/channels
• But we need more …

• Combined models
  - GMM models
  - Ph-based models
  - Combine them
  - Slightly better results

• What else …
  - Prosody (duration and F0)
• Can VC beat Speaker-ID

• Can we fake voices?
• Can we fool Speaker ID systems?
• Can we make lots of money out of it?

• Yes, to the first two
  - Jin, Toth, Black and Schultz ICASSP2008
Training/Testing Corpus

- **LDC CSR-I (WSJ0)**
  - US English studio read speech
  - 24 Male speakers
  - 50 sentences training, 5 test
  - Plus 40 additional training sentences
  - Sentence average length is 7s.

- **VT Source speakers**
  - Kal_diphone (synthetic speech)
  - US English male natural speaker (not all sentences)
Experiment I

• VT GMM
  - Kal_diphone source speaker
  - GMM train 50 sentences
  - GMM transform 5 test sentences

• SID GMM
  - Train 50 sentences
  - (Test natural 5 sentences, 100% correct)
GMM-VT vs GMM-SID

- **VT fools GMM-SID 100% of the time**
GMM-VT vs GMM-SID

- Not surprising (others show this)
  - Both optimizing spectral properties
- These used the same training set
  - (different training sets doesn’t change result)
- VT output voices sounds “bad”
  - Poor excitation and voicing decision
- Human can distinguish VT vs Natural
  - Actually GMM-SID can distinguish these too
  - If VT included in training set
VT is always S17, S24 or S20
Kal_diphone is recognized as S17 and S24
Phone-SID seems to recognized **source** speaker
and Synthetic Speech?

• Clustergen: CG
  - Statistical Parametric Synthesizer
  - MLSA filter for resynthesis

• Clunits: CL
  - Unit Selection Synthesizer
  - Waveform concatenation
Smaller is better
Synth vs Phone-SID

- Smaller is better
- Opposite order from GMM-SID
Conclusions

• GMM-VT fools GMM-SID
• Ph-SID can distinguish source speaker
  - Ph-SID cares about dynamics
• Synthesis (pretty much) fools Ph-SID
  - We’ve not tried to distinguish Synth vs Real
Future

• Much larger dataset
  - 250 speakers (male and female)
  - Open set (include background model)
  - WSJ (0+1)
• Use VT with long term dynamics
  - HTS adaptation
  - Articulatory position data
  - Prosodics (F0 and duration)
• Use ph-SID to tune VT model
Future II

• VT that fools Ph-SID
  - Develop X-SID (prosody?)
    • Develop X-VT that fools X-SID
      - Develop X2-SID
        • Develop X2-VT that fools …
        …..
• De-identification

• Using Speaker ID to score de-identification
  - Reverse of voice transformation
    • Masking source, rather than being like target

• Simplest view
  - Full ASR and TTS in new engine (two hard)

• Voice conversion to synthetic voice
  - Natural speech to TTS (kal_diphone)
De-identification

- Morph your voice to something else
- Use voice conversion technology
- Mostly works (for spectral/phonetic information)
  - But what about words?
  - But what about timing/location/source
Future

• Advisory Development
  - ID, counter-ID, better ID, better counter-ID
• Evolution is a very strong function
• De-identification hides your voice
  - But hides the others’ voices too
• We could just end up with the best bot
Always Listening ...

- Google Glass, Amazon Echo
  - Looks for keyword ...
  - So listens all the time
  - (But doesn't upload to the cloud, probably)
- What happens to the data I give up
  - Sentences do get uploaded.
  - (Probably) protected partially
- What about hackers:
  - Malicious, legal and “legal”
So we're doomed!

- Can we have web services and privacy?
So we're doomed!

- Can we have web services and privacy?
- Maybe ...
Homomorphic Encryption

- Doing Arithmetic in the Encrypted domain.
- For example:
  - Electronic voting
  - Summing bank account values
- Pass the encrypted cumulated values
  - Sum them in the encrypted domain
  - $\text{st. unencrypt}(a') + \text{unencrypt}(b') = \text{unencrypt}(a' \text{ “+” } b')$
Homomorphic Encryption

- No unencrypted data is given to the server
- e.g.
  - HIPAA requirements:
    - ASR without revealing the content
  - Can search encrypted calls from Terrorist without (unencrypted) access to non-Terrorist calls
- Can still update general models (ish)
Homomorphic Encryption

- Privacy Preserving Speech Processing (Manas Pathak 2012)
- Keyword spotting and HMM Recognition
- Great, where can I download it ...
Homomorphic Encryption

- Privacy Preserving Speech Processing (Manas Pathak 2012)
- Its computational very expensive
- (300-3000 times slower)
- It requires transfer of much more data
So We’re Saved

Maybe:
- We have to trust the makers for cryptography
- We have to do develop new anticryptography
- We have to be vigilant
  - (don't check your private keys into github)