Privacy Preserving Autocoders

Rob Chew, RTI
Terrance D. Savitsky, BLS
Matt R. Williams, RTI
Elan Segarra, BLS

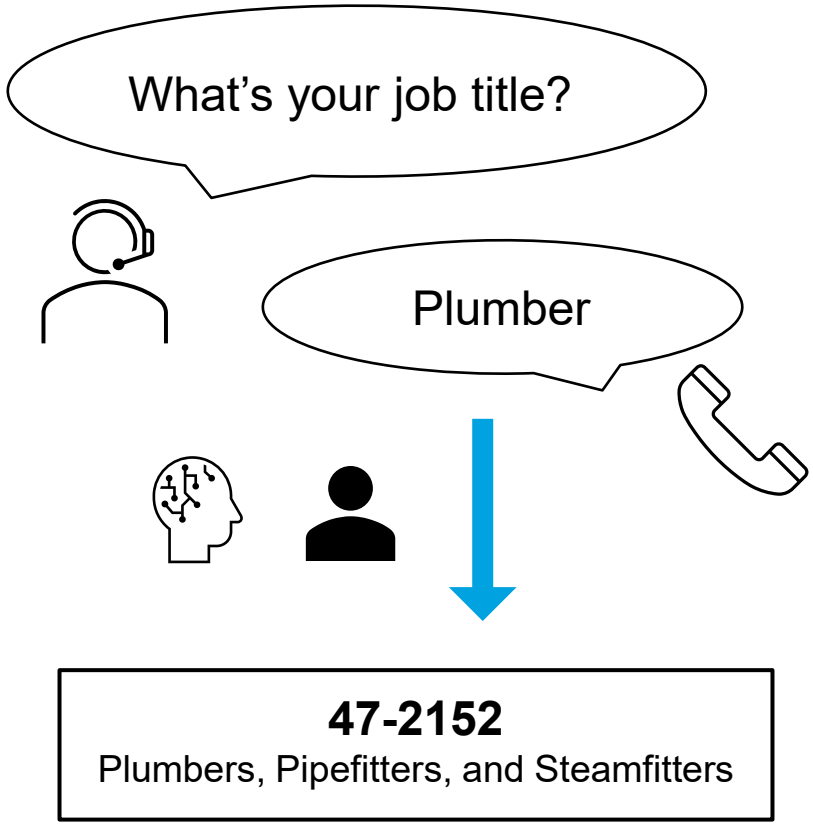
10/23/24

FCSM 2024, Session F-2:
New Bayesian Methods for Statistical Data Privacy

RTI INTERNATIONAL

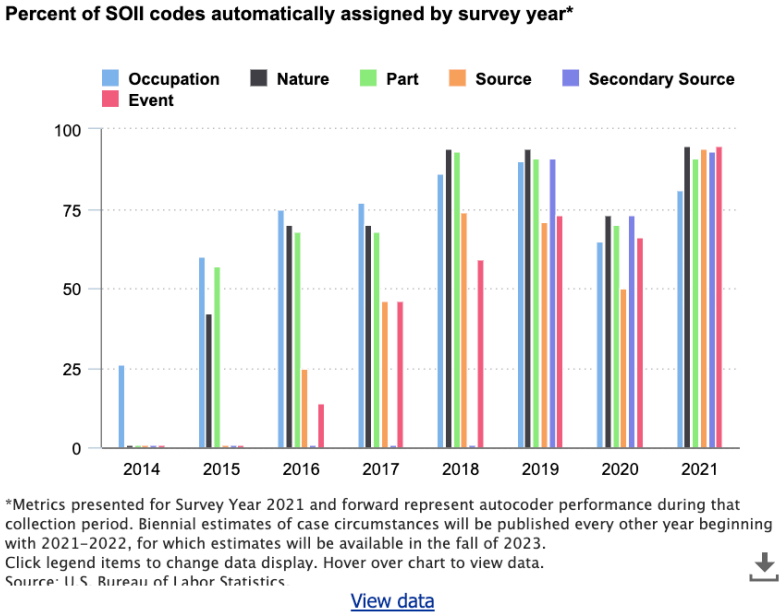
Survey Coding

- Assign numerical codes to free form text responses
  - Standardized categories → statistical analysis
- Traditionally, trained coders learn the classification system and assign text
  - Require substantial training
  - Time consuming and labor intensive
- **Autocoding:** computer program automatically assigns codes
  - Early autocoders used fixed dictionaries
  - ML/NLP models are now more common
  - Often, both human coders and models are used



BLS SOII Autocoders

- SOII autocoders automatically assign ~85% of all codes with expected error rate ≥ manual coding
- BLS receives requests from external parties to use the SOII autocoders.
  - However, BLS is unable to fulfill requests due to concerns of disclosure risk
- ML models can leak information about their training data if attacker has access to deployed model
  - Membership leaks increase as # classes increase (Shokri et al., 2017; Truex et al., 2021)
  - Models with more parameters tend to be more susceptible to high accuracy attacks (Nasr et al., 2019)
- Concerning for SOII autocoders since they:
  - Have large # of classes (e.g., >850 occupation codes)
  - Use deep learning models (i.e., DistilRoBERTa)

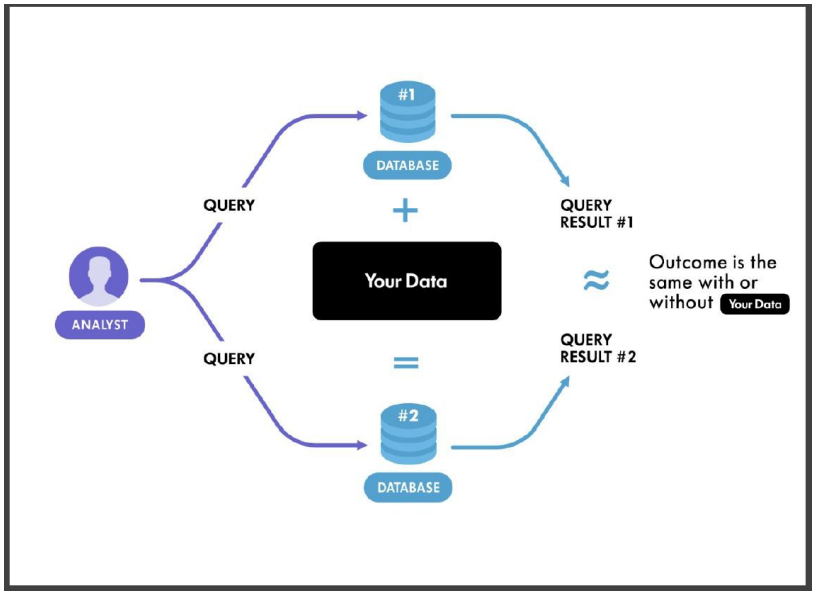


https://www.bls.gov/iif/automated-coding.htm

Privatizing Models with the Posterior Mechanism

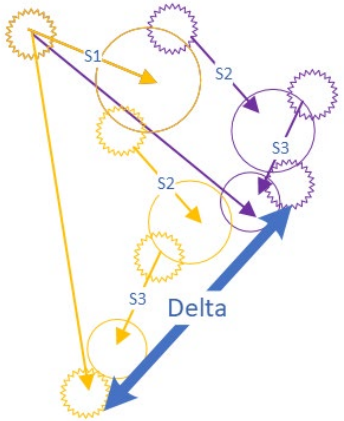
Differential Privacy (DP)

- Ensures that an algorithm applied to two databases *differing by one record* will result in similar output
  - Promises that the chance of an outcome is about the same whether or not you contribute your data
- For supervised ML, the “two databases” are training sets that differ by one training example
- Allows for rigorous quantification of privacy



Random Additive Noise vs. Random Selection

- Many popular DP mechanisms add random noise
- Unfortunately, additive noise mechanisms tend to suffer from a few drawbacks:
  - They can produce values that are not plausible (negative counts of people)
  - Challenging to calculate the sensitivity bound for a complex operation– instead add noise to each intermediate step
- Random selection mechanisms can sometimes be a good alternative



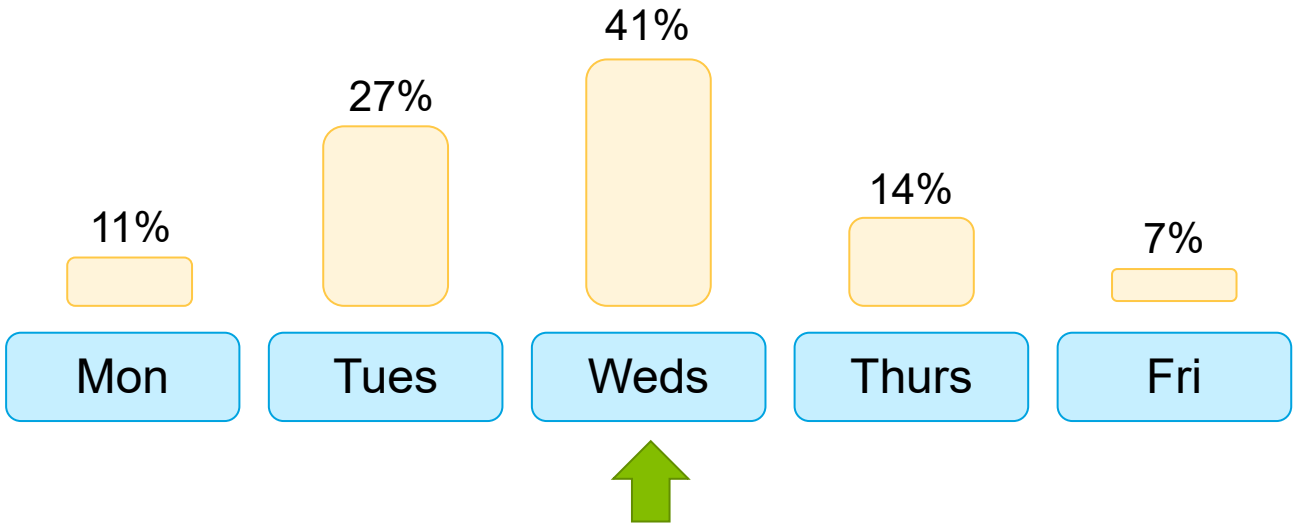
Exponential Mechanism

"Management wants to know **what day of the week is most popular** for staff to come into the office."

- We take a poll of the office and ask each person what day of the week they most consistently come in
- **Wednesday** is the most common answer

Exponential Mechanism

- Maybe we use the survey response distribution
  - Now Weds is selected more often than others (~41%)
  - However, utility is still not great and it's not differentially private (yet)

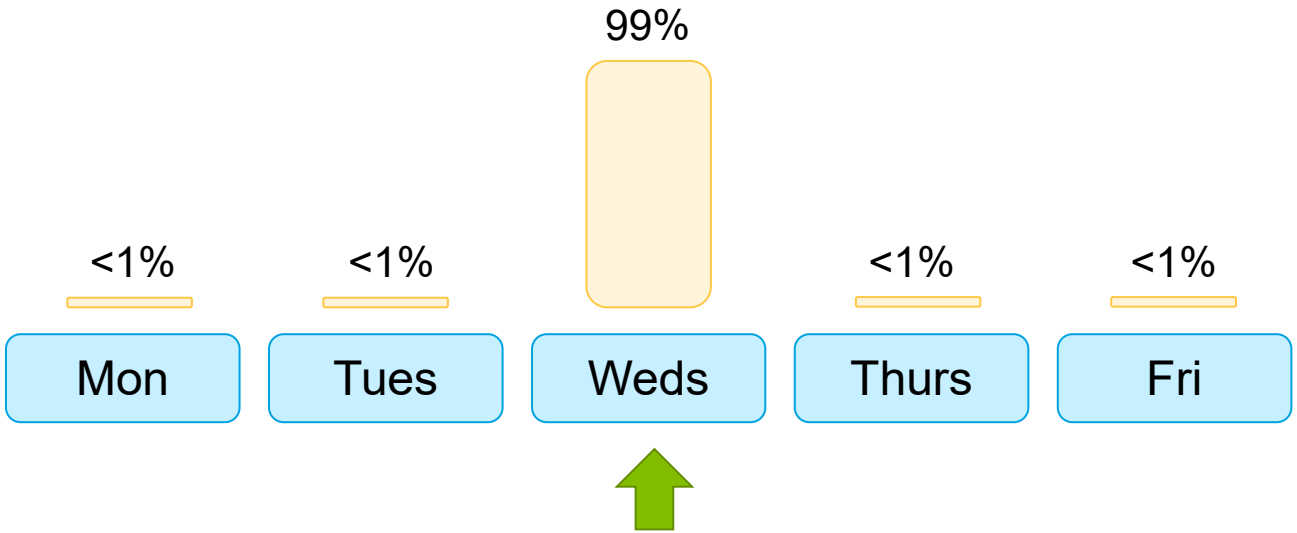


Exponential Mechanism

- The exponential mechanism selects an output with probability proportional to the exponent of a scoring function
- If we use the response distribution from last slide as a scoring function, we're in business
  - Now Weds is almost always selected (~99%)

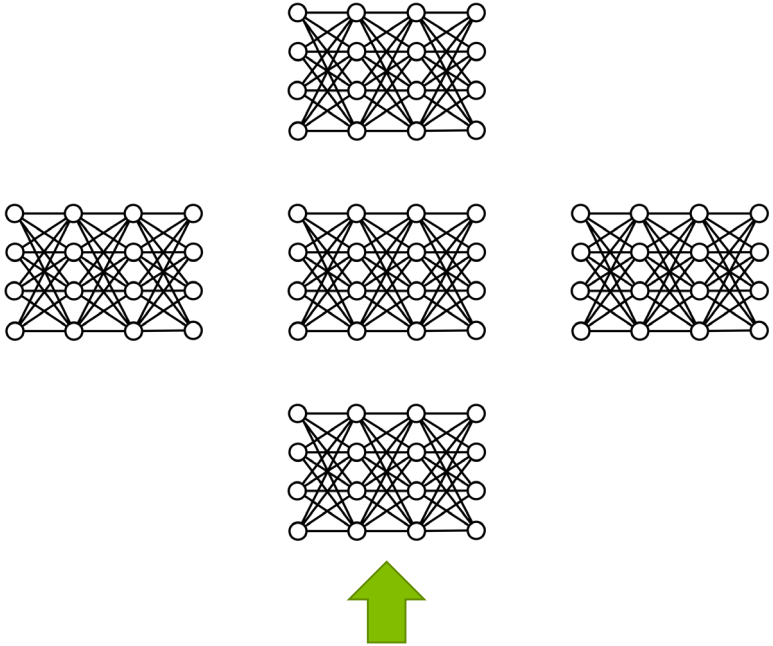
$$\exp\left(\frac{\varepsilon * u(x,r)}{2\Delta}\right)$$

$\varepsilon = 1$
$\Delta = 1$



Privatizing Machine Learning with Random Selection

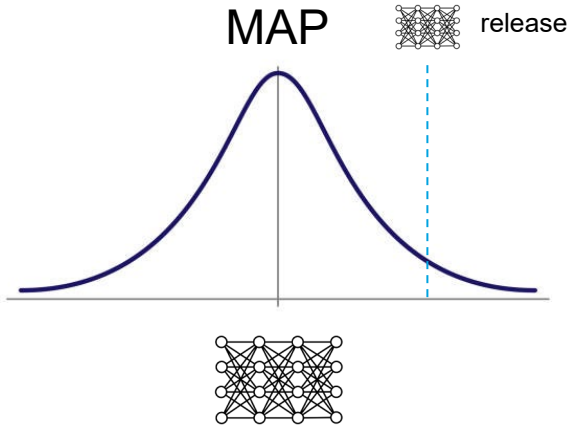
- Now we want to share a machine learning autocoder
  - Instead of a mode, we're sharing fitted model parameters
- Using the previous logic, if we had different models all trained on the private data, maybe we could randomly select one of them to release.
- However, this leaves some questions:
  - What models should be considered?
  - How should we select which model to release?
  - What privacy protection, if any, does this provide?



Icon created by Lucas Rathgeb

Posterior Mechanism

- Bayes to the rescue!
- What models should be considered?
  - The posterior distribution characterizes **all combinations of parameters**, given fixed training data, model specification, and hyperparameters
- How should we select which model to release?
  - A draw from the posterior distribution will release one trained model
  - **Parameter sets closer to the best fit** (i.e., mode of the posterior distribution or MAP) **are more likely to be selected**
- What privacy protection does this provide?
  - The posterior mechanism is a version of the exponential mechanism (Wang et al., 2015; Dimitrakakis et al., 2017)
  - Scoring function is the posterior $P(\theta|D)$
  - $\boldsymbol{\varepsilon}$**-DP** (exact inference)**;** $(\boldsymbol{\varepsilon}, \boldsymbol{\delta})$**-DP** (approximate inference)



SWAG Pseudo Posterior Mechanism

Posterior Mechanism with Deep Learning?

- Unfortunately, applying the posterior mechanism to modern autocoders (using deep learning) poses some challenges
- The sensitivity can be unbounded
  - The likelihood / loss values for these models can be arbitrarily large
  - This makes it hard to define the sensitivity (max θ can change when an obs is added or removed)
  - We address this with the **Pseudo Posterior Mechanism**
- Bayesian deep learning is both computationally and inferentially challenging
  - Deep learning models have millions to billions of parameters and non-convex loss landscapes.
  - Exact Bayesian inference (computing the “true” posterior distribution) isn’t possible with deep learning models
  - Most popular approximate Bayesian inference methods (MCMC, Metropolis-Hastings, HMC) aren’t feasible with deep learning
  - We address this with **Gaussian Stochastic Weight Averaging (SWAG)**
- We combine these two ideas to propose the **SWAG Pseudo Posterior Mechanism**

Pseudo Posterior Mechanism

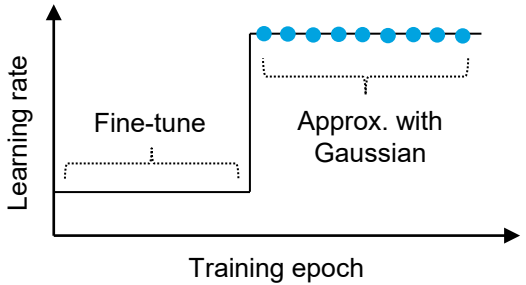
- The pseudo posterior mechanism (Savitisky et al., 2022) extends the posterior mechanism with a weighted likelihood function and risk-based case weights
- This offers two main benefits:
  - It bounds the loss values, ensuring a finite sensitivity
  - It downweights riskier records, improving utility
- The case weights (red) are generated by finding the max loss values across posterior draws for each record
  - Records with higher max loss values are “riskier”, since they diverge more from the model predictions and are therefore easier to identify

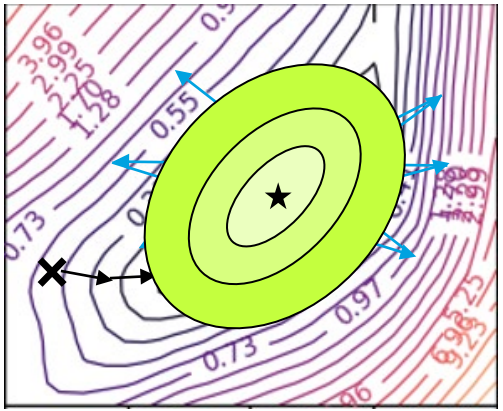
$$\xi^{\boldsymbol{\alpha}}(\theta \mid \mathbf{x}, \gamma) \propto \left[\prod_{i=1}^{n} \pi(x_i \mid \theta)^{\alpha_i}\right] \pi(\theta \mid \gamma)$$

$$\alpha_i \propto 1/\sup_{\theta \in \Theta} \mid f_\theta(x_i) \mid$$
$$\text{with } f_\theta(x_i) = \log(\pi(x_i \mid \theta))$$

Gaussian Stochastic Weight Averaging (SWAG)

- SWAG (Maddox et al., 2019) is an approximate Bayesian inference method for deep learning
  - Builds off work by (Mandt et al., 2017) showing that stochastic gradient descent (SGD) with a constant learning rate can capture the shape of the posterior
- SWAG extends this for deep learning by proposing a three-step approach:
  - Initially fine-tune model with a small learning rate until you get to a local minima
  - Continue training with a larger constant learning rate to explore the space around the local minima
  - Use the iterations from the higher LR to estimate a multivariate Gaussian as approximate posterior distribution

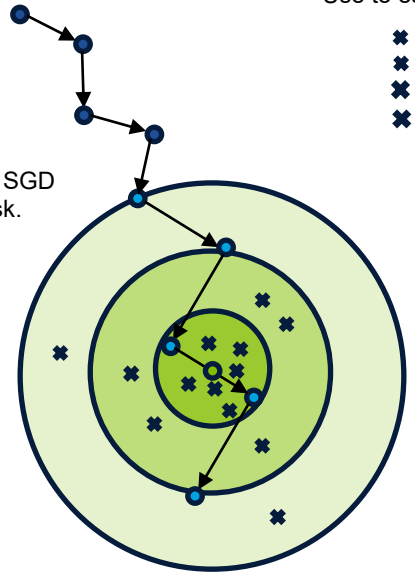




SWAG Pseudo Posterior Mechanism

Step 1:
Start with pre-trained model parameters.

Step 2:
Fine-tune with SGD for new task.



Step 4:
Take draws from SWAG.
Use to estimate risk-based weights.

$$a_i \propto \frac{1}{max_\theta \, l(y_i|\theta)}$$

Step 3:
Continue with high constant learning rate to estimate SWAG.

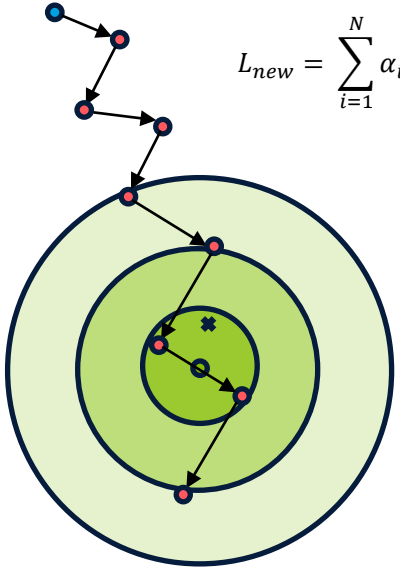
SWAG Posterior

$$\theta|y \sim \mathrm{N}(\hat{\theta}, \tfrac{1}{2}(\Sigma_{diag} + \Sigma_{lowrank}))$$

$$\Sigma_{lowrank} = \frac{1}{K-1}\sum(\theta_k - \bar{\theta}_k)\,(\theta_k - \bar{\theta}_k)^T$$

Step 5:
Return to Step 2 parameters.
Fine-tune using risk-based weights.

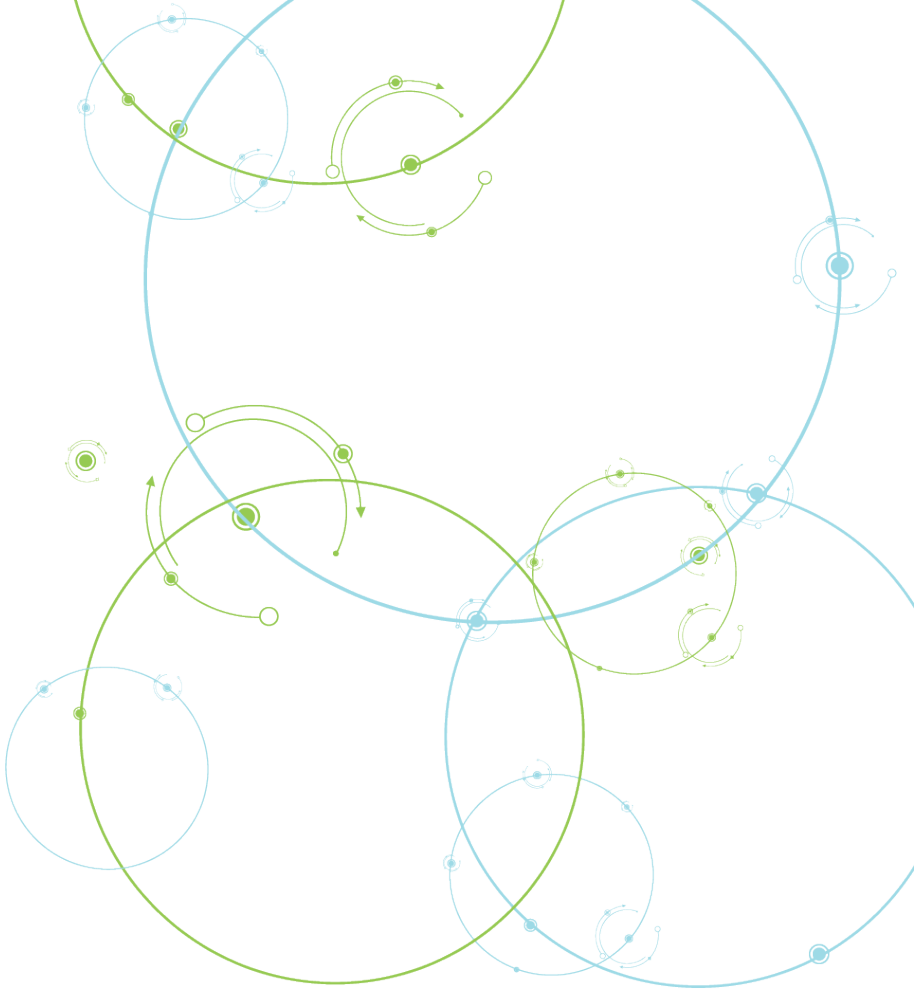
$$L_{new} = \sum_{i=1}^{N} \alpha_i * l(y_i, \theta_i)$$



Step 6:
Estimate final SWAG.
Take single draw to create final private model.

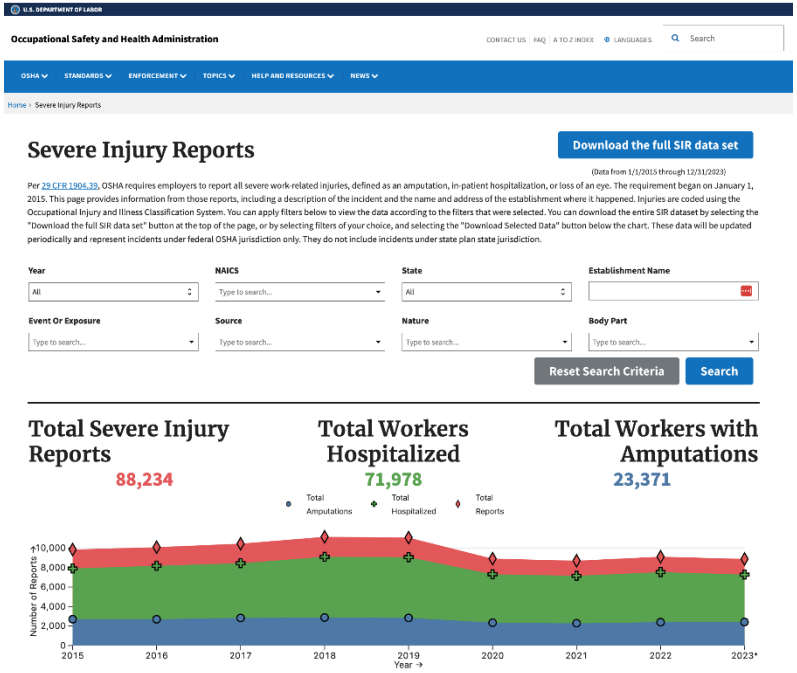
$$\Delta^{\alpha_i} = max_\theta \; \alpha_i * l(y_i, \theta)$$
$$\Delta^{\alpha} = max_i \; \Delta^{\alpha_i}$$
$$\varepsilon = 2\Delta^{\alpha}$$

Results



Data

- **OSHA's Severe Injury Reports**
  - Jan 2015 – Sept 2023
  - Similar structure and same outcomes as BLS SOII
- **Input:**
  - Final Narrative
  - *"A truck driver fell approximately 4 feet while descending a tanker trailer ladder."*
- **Outcome:**
  - Nature of Injury (OIICS code)
  - *"Traumatic injuries and disorders, unspecified"* (10)
- **Summary Stats:**
  - N = 10,692
  - Classes = 153
  - 50/50 Training / Test split
  - Outcome distribution highly skewed



https://www.osha.gov/severe-injury-reports

Model Benchmarks

- **Non-Private**
  - Provides no privacy guarantees but will give us an upper bound on utility
- **SWAG Pseudo Posterior**
  - Uses same hyperparameters as non-private during fine tuning, but higher constant learning rate during SWAG estimation
- **DP-SGD (Abadi et al., 2016)**
  - Most popular and well studied differentially private mechanism for machine learning
  - Additive noise model (adds noise to batch gradients)
  - Use recommended hyperparameters for private fine-tuning of transformer models (Li et al., 2022; Yu et al., 2022)

Findings

- For non-private model, weighted F1 is higher than macro F1
  - Macro F1 weights each class equally
  - Weighted F1 weights more populace classes more
  - Performs better on classes with more obs
- SWAG has same weighted F1 as non-private, but lower macro F1
  - Though there is a drop in model performance, it's modest
  - The privacy protection is also good ($\varepsilon = 4$)
  - However, delta is unknown – asymptotically, approaches 0 but additional analyses needed for finite samples
- DP-SGD performance is far worse for the same approximate level of privacy
  - Difference more pronounced with our skewed outcome distribution and sample size
  - Performance gap should shrink as the training size grows

Model	Privacy		Utility	
	Epsilon	Delta	Weighted F1	Macro F1
Non-Private	-	-	0.77	0.50
SWAG Pseudo	4.35	Unknown	0.77	0.46
DP-SGD	4	10^-4	0.09	0.03

Why is SWAG Pseudo Posterior utility so high?

- We believe it's an interaction of the case weights and skewed outcome distribution
  - The method downweights obs with high loss (i.e., those poorly predicted by the model)
  - Normally, heavier downweighing reduces utility since it forces the model to "gives up" on trying to learn certain types of obs
  - However, obs with high loss are more highly concentrated in the numerous rare classes (49% classes $\leq$ 10 obs in training)
- Top vs. Bottom Class Size Quartiles
  - We aggregated the 25% largest and 25% smallest classes together and compared performance between non-private and SWAG models
  - **Top quartile is roughly the same**. This is because observations in these popular classes are barely downweighed.
  - **Bottom quartile is lower for SWAG than non-private**. We now see the impact of heavier downweighing. However, since the non-private performance was already poor, the difference isn't huge.

Model	Class Size Quartile	F1 Macro	F1 Weighted
Non-Private	Top	0.83	0.83
SWAG Pseudo	Top	0.82	0.83
Non-Private	Bottom	0.18	0.2
SWAG Pseudo	Bottom	0.06	0.08

Take-aways

- Trained ML models can leak information about their training data, with disclosure risk implications for survey operations
- DP random additive noise mechanisms (e.g., DP-SGD) can work, but may struggle with smaller training sets and/or highly skewed outcome class distributions
- DP random selection mechanisms (e.g., SWAG Pseudo Posterior Mechanism) may provide better utility / privacy trade-offs under these conditions

Thank you

Contact: Matt Williams| email: mrwilliams@rti.org

References

- Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L. (2016, October). Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security* (pp. 308-318).
- Dimitrakakis, C., Nelson, B., Zhang, Z., Mitrokotsa, A., & Rubinstein, B. I. (2017). Differential privacy for Bayesian inference through posterior sampling. Journal of machine learning research, 18(11), 1-39.
- Li, X., Tramer, F., Liang, P., & Hashimoto, T. (2021). Large language models can be strong differentially private learners. *arXiv preprint arXiv:2110.05679*.
- Maddox, W. J., Izmailov, P., Garipov, T., Vetrov, D. P., & Wilson, A. G. (2019). A simple baseline for bayesian uncertainty in deep learning. *Advances in neural information processing systems*, *32*.
- Mandt. S., Hoffman, M. D., & Blei, D. M. (2017). Stochastic gradient descent as approximate bayesian inference. *Journal of Machine Learning Research*, *18*(134), 1-35.
- Nasr, M., Shokri, R., & Houmansadr, A. (2019, May). Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In 2019 IEEE symposium on security and privacy (SP) (pp. 739-753). IEEE.
- Savitsky, T. D., Williams, M. R., & Hu, J. (2022). Bayesian pseudo posterior mechanism under asymptotic differential privacy. Journal of Machine Learning Research, 23(55), 1-37.
- Shokri, R., Stronati, M., Song, C., & Shmatikov, V. (2017, May). Membership inference attacks against machine learning models. In 2017 IEEE symposium on security and privacy (SP) (pp. 3-18). IEEE.
- Truex, S., Liu, L., Gursoy, M. E., Yu, L., & Wei, W. (2019). Demystifying membership inference attacks in machine learning as a service. IEEE transactions on services computing, 14(6), 2073-2089.
- Wang, Y. X., Fienberg, S., & Smola, A. (2015, June). Privacy for free: Posterior sampling and stochastic gradient monte carlo. In International Conference on Machine Learning (pp. 2493-2502). PMLR.
- Yu, D., Naik, S., Backurs, A., Gopi, S., Inan, H. A., Kamath, G., ... & Zhang, H. (2021). Differentially private fine-tuning of language models. *arXiv preprint arXiv:2110.06500*.