Ethical Data Management

Han Yu[1], Jianshu Weng[2], Yew Soon Ong[1,3], and Gao Cong[1]

[1]School of Computer Science and Engineering, Nanyang Technological University, Singapore
[2]AI Singapore, Singapore
[3]A*STAR, Singapore

I. Introduction

In today's world, machine learning (ML) and artificial intelligence (AI) have been adopted and embedded into a wide range of applications. From natural language processing (NLP) and image processing to systematic applications such as healthcare decision support, ML and AI have demonstrated remarkable value and changed how such tasks are performed. In this chapter, we provide an overview of data management as it relates to auditability issues in the ethics of AI.

For AI to advance further, data management is one of the most crucial issues which must be addressed. Most successful AI applications are data-driven. AI applications often depend on collecting, aggregating, processing and managing large volumes of various types of data.

When training AI models, considerable time is spent on preparing and processing the data, including data collection, data labelling, and sometimes, handling legacy data. Data quality is vital for ML, and data acquisition can improve data quality, which includes data discovery, data cleaning, data integration, data labelling and data lineage. Each can take up considerable time and effort and may require data scientists performing these tasks to comply with data handling regulations and guidelines, such as the General Data Protection Regulation (GDPR) and the Personal Data Protection Act (PDPA).

As AI enters new application domains, it may run into problems such as shortage of data during the bootstrapping phase. For example, in an automated car manufacturing plant run mostly by algorithms, new failure cases (with previously unencountered causes) may throw the AI decision support system into disarray due to lack of training data. That can be challenging for complex deep neural network models which require large amounts of labelled training data to achieve a desirable level of performance.

For such challenges, the right approach would be to try to gather and label more training data for any novel AI applications. It may take a while for new data to be generated by the latest applications, as well as incur high cost.

With the surge in crowdsourcing platforms [Doan et al., 2011], data labelling tasks can be outsourced to a large number of "crowd workers". Crowd workers are generally educated and use crowdsourcing to hone their skills or earn extra money. Data scientists must, however, pay attention to quality control as crowd workers may lack proficiency. Moreover, during the data collection and labelling stage, the performance of the AI model might remain low, which can cause deterioration in user experience.

Data scientists dealing with such issues can also leverage transfer learning [Pan & Yang, 2010] methods. That essentially allows an existing AI model previously trained for a different but similar domain to be adapted to operate in the new domain with a small amount of initial training data.

Such an approach can help the new application with a satisfactory level of performance to tide over the bootstrapping phase with an acceptable level of user experience. Transfer learning can also be a useful approach for data scientists to enable legacy data to make an impact on new AI applications.

Nevertheless, this must be done in a way that complies with data privacy protection laws which may limit how existing data can be used. The emerging technique of federated learning [Yang et al., 2019] may need to be leveraged when dealing with legacy data stored across different data silos.

This chapter discusses the three critical facets of data management – data acquisition, data labelling and leveraging legacy data – to set the stage for discussions on auditability issues later. Readers who want to explore more details about data management can refer to survey articles such as [Roh et al., 2019].

II. Facets of Data Management

For an understanding of data management for AI, organisations will need to:

- Review the most vital data acquisition methods with a focus on the discovery, augmentation and generation of data.
- Ensure useful data labelling techniques, including utilising existing labels, using crowdsourcing-based labelling, and weak supervision-based labelling.
- Review methods for leveraging existing data or models, especially when acquiring and labelling new data is not feasible.

A. Data Acquisition

Let's first get a higher-level view of the role of data in a typical ML/AI project lifecycle. CRISP-DM (Cross-industry standard process for data mining) is an open standard process model that describes the common approaches used by data mining experts. It was conceived more than 20 years ago for the data mining community. The fundamental principles are still relevant to the AI community. Data remains the core, and there are four broad stages that projects typically execute:

- Business understanding.
- Data acquisition and understanding.

- Modelling, including data preparation/feature engineering, modelling, and evaluation.
- Deployment.

Note that there are alternative methodologies (such as SEMMA by SAS, and Team Data Science Process by Microsoft). The four-stage view is common across those different methodologies.

The process is often iterative. It usually starts with a business problem statement, which helps steer the data acquisition, such as what data is needed, and how to obtain it. Understanding of data could also help to refine the problem statement. For example, data understanding may lead to a decision that the collected data is insufficient. However, if the acquisition of more data is not a viable option – due to technical or other reasons – the problem statement may need to be revised.

In the modelling stage, data transformation, such as missing value imputation, binning, feature selection, and others, are conducted before the data is ready for model training. Typically, multiple models are built with different training algorithms and hyper-

parameters. After that, model evaluation is conducted to choose the one/s with the optimal performance among the multiple models built.

Standard techniques include cross-validation and A/B testing, among others. The chosen model/s would then be deployed into a production environment to provide inference on the new data. It is possible that the models built may not achieve the desired performance. In such cases, more data may be necessary, or the problem statement may need to be revised. And a new iteration of the different stages would start.

With this end-to-end view in mind, let's take a closer look at data acquisition. The goal is to obtain data that can be used to train ML/AI models. From a data management perspective, there are broadly three main approaches: data discovery, data augmentation, and data generation

[Roh et al., 2019]. These are not mutually exclusive, and a combination of multiple approaches should be applied.

a. Data Discovery

Data discovery is the process for searching for existing data that is available either internally or externally in an organisation [Roh et al., 2019]. With the rapid pace of digitalisation across most industries, more data gets generated. Typically, data is generated with different systems. Note that the data owners may not publish the generated information through one system. That data would be uploaded and stored in data lakes within an organisation [Terrizzano et al., 2015].

External data is also valuable for ML/AI projects. For example, data.gov.sg is a one-stop portal for accessing publicly-available datasets from 70 public agencies in Singapore. Quandl provides investment professionals with financial, economic, and alternative datasets. Commercial data discovery systems, like Infogather by Microsoft [Yakout et al., 2012], can also be tapped.

However, much of that data is generated without the intention of supporting specific downstream ML/AI project needs. Therefore, it may not be possible to make the data easily discoverable by other teams or individuals in the organisation.

Thus, there is a need for a data catalogue system. Gartner noted in a 2017 report that “data catalogue maintains an inventory of data assets through the discovery, description and organisation of datasets. The catalogue provides context to enable data analysts, data scientists, data stewards and other data consumers to find and understand a relevant dataset to extract business value.”

b. Data Cleaning

Data cleaning techniques can also be used to boost the performance of AI models. Model training can be carried out iteratively in two steps: One, estimating the likelihood that the data is noisy; and two, selecting data samples to be cleaned based on how much the cleaning has improved the model accuracy so far [Krishnan et al., 2016]. These can be used to carry out data transformation and filtering to clean the data samples.

Some techniques have been proposed to clean crowdsourced labels using “oracles”, which are programs or individuals that can make accurate judgements about the labels. The TARS method [Dolatshah et al., 2018], for example, assists data scientists to clean crowdsourced data by providing decision support in two aspects:

- It predicts how well the AI model can perform if the labels are corrected; this will help a data scientist determine if the data cleaning is worthwhile.
- It determines which data samples shall be sent to the oracle for cleaning to achieve maximal improvement on model performance.

Label quality is instrumental to the performance of AI models. When labels are noisy, the accuracy will not increase when more training data is fed to the model. In the case of crowdsourced data labels, one way to improve label quality is through repeated labelling using crowd workers with a level of past competence. A round-robin labelling approach can substantially improve the resulting label quality. Reputation-based crowd worker selection based on past performance can boost data label accuracy [Sheng et al., 2008].

Legacy data may have already been used to train AI models for some ML tasks. Additionally, the existing AI models can also be reused to train new AI models, especially if the new learning task does not yet have enough labelled training data.

c. Data Augmentation

Data augmentation complements data discovery, where existing data is augmented or enhanced. For structured data (such as data that can be stored and found in the relational DBMS), data integration can be considered as one form of data augmentation [Stonebraker & Ilyas, 2018].

Data integration is a well-established discipline. Simply put, data integration involves combining data residing in different sources and providing a unified view of them. This is relevant for ML/AI because most ML/AI toolkits would assume that the training data is a single file, and ignore the fact there are multiple tables in a database due to normalisation.

In its most primitive form, data integration involves key-foreign key (KFK) joins. One noticeable system is Hamlet, and subsequently, Hamlet++ [Kumar et al., 2016; Shah et al., 2017], which tried to address the question whether KFK joins are necessary for

improving the model accuracy for various classifiers (linear, decision trees, SVMs, and neural networks). A key finding is that KFK joins can often be avoided without negatively influencing the model's accuracy. It also proposed rules to predict when it is safe to prevent joins – and significantly reduce the total runtime.

Data cleaning is an essential component of data augmentation, as most datasets are dirty or inconsistent, which could result in inaccurate data analytics results and incorrect business decisions. For instance, poor data across businesses cost the US trillions of dollars a year [Ilyas & Chu, 2019].

Data cleaning is a very active area of research. Different approaches have been proposed to solve the different types of data cleaning tasks, including outlier detection, data transformation, error repair, and data deduplication. Traditional functional dependencies and conditional functional dependencies [Cong et al., 2007] have been utilised for developing data cleaning algorithms. Many ML methods (Mudgal et al., 2018]) have been used for data cleaning.

In many cases, data is incomplete and needs to be augmented with additional information. For example, to build an AI model to predict whether a life insurance policy is likely to lapse, only looking at data within the insurance domain would be insufficient. Data from other industries, such as the financial status of policyholders, which could be obtained from banks, would help to augment the data. There are also privacy concerns related to cross-organisation data sharing.

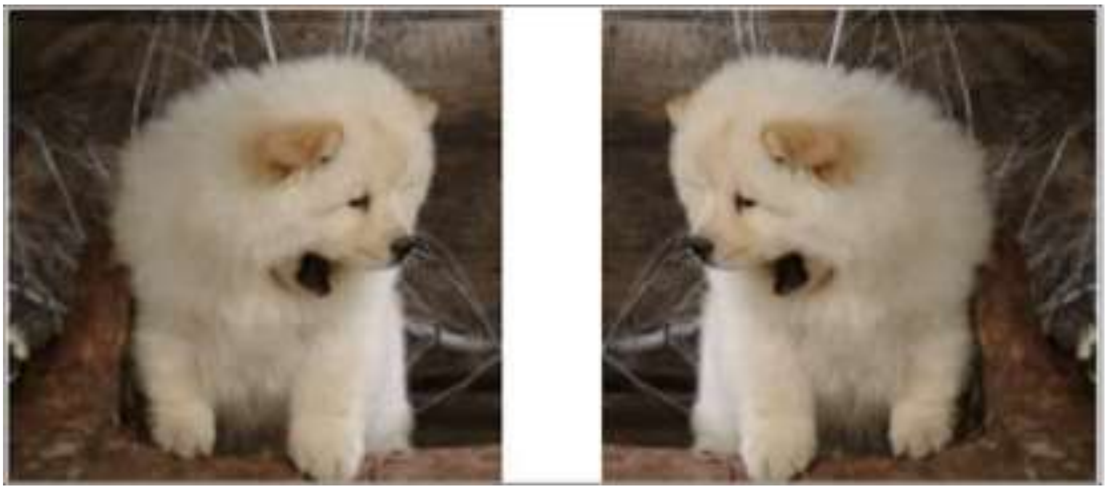


Figure 1: Left: original image, Right: image transformed with a horizontal flip (Source: ImageNet)

For image data, data augmentation can be done by creating new images by different ways of processing or combination of multiple processing, such as random rotation, shifts, shear, flips, contrast change, adding white noise, etc. Figure 1 shows an image and a transformed version.

Image augmentation is usually done manually. However, determining which augmentation will work best for the data at hand is not easy. Google has suggested “AutoAugment” to increase both the amount and diversity of data in an existing image dataset in a more automatic and scalable manner [Cubuk et al., 2019]. It uses reinforcement learning to find the optimal image transformation policies from the data itself. It predicts what image transformations to combine – as well as the per-image probability and magnitude of the conversion used – so that the image is not always manipulated in the same way.

However, using Google’s AutoAugment requires extensive computational resources. DeepAugment [Özmen 2019] applies Bayesian Optimisation [Shahriari et al., 2016] instead of Reinforcement Learning to find the optimal combinations of transformation.

d. Data Generation

Data generation is needed when there is no available data. If there is a business process to generate the data, the process could be run for a sufficient period to generate enough data to train an ML/AI model. For example, a smart factory wants to train an image classification model to decide whether a specific component is defective. Initially, there may be no data available to train the model. The organisation can run the production line to generate images of various product components.

Data generation can also be viewed as data augmentation if there is existing data with some missing parts or fields. In the above example, it could be that the data generated from the number of components is not sufficient to train a reliable defects detection model. In that case, image augmentation techniques like DeepAugment could be applied to increase the amount and variety of product images.

Generating synthetic data along with labels is increasingly being used in ML/AI projects due to its low cost and high flexibility [Patki et al., 2016]. A simple method is to start from a probability distribution and generate a sample from that distribution, using toolkits like Scikit-learn [Pedregosa et al., 2011].

There are also more advanced techniques like Generative Adversarial Network (GAN) [Goodfellow et al., 2014] which has seen great success in synthesising image data. The key idea of GAN is to train two competing networks – a generative network and a discriminative network.

The generative network learns to map from a latent space to data distribution. The discriminative network classifies examples taken from the actual distribution from the candidates produced by the generative network. The goal of the generative network is to fool the discriminative network into thinking that its candidates are from the actual distribution.

GAN is not as successful in generating textual data due to the discrete nature of how text generation is done in neural networks [Subramanian et al., 2018]. GPT-2 has found applications in various text generation systems [Radford et al., 2019].

GAN has recently been applied in synthesising structured data, but those applications are mainly in the research community, such as MEDGAN [Choi et al., 2017]. MEDGAN applies GAN to generate synthetic Electronic Health Records (EHR) with discrete features which are based on real EHR. GAN can only learn to approximate discrete patient records with continuous values.

To address that, an autoencoder is applied to understand the salient features of discrete variables. It is used to project the records into a lower-dimensional space and then project to the original space. With the pre-trained autoencoder, GAN can generate the distributed representation of patient records (which is the output of the encoder), rather than the patient records directly.

Suggestions to improve the quality include designing the right interface to maximise worker productivity, managing workers who may have varying skill levels (or may even be spammers), and decomposing problems into smaller tasks and aggregating them.

A simple interface, such as one with non-verifiable questions, could make it easier for deceptive workers to exploit the system. On the other hand, an unnecessarily complicated interface will discourage honest workers and lead to delays.

For worker selection, allowing workers with requisite skills to contribute to the tasks could help improve the quality. Complex tasks usually need to be broken down into simpler subtasks. Solving a complex task might require more time, cost and expertise, so fewer people will be interested or qualified to perform it.

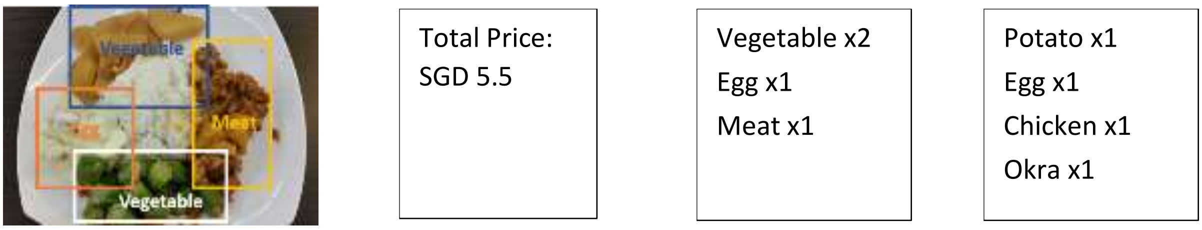
B. Data Labelling

In an ideal situation, the data comes labelled and can be directly used for supervised training. In other instances, you can apply unsupervised learning, which does not require the data to be labelled. In most cases, however, you will need to factor in the additional step of data annotation. That's where the collected data is assigned labels or structured annotations, usually by human annotators. In data labelling, you need to decide on the content to be annotated and the procedures for obtaining the annotations. The goal: high label-quality, broad label-coverage, and low annotation cost.

a. *Designing the Annotation Schema*

The annotation schema refers to the format of the labels. The design of the schema directly determines the cost of the annotation acquisition process. A good annotation schema is one where:

- The labels are as simple as possible.
- The label definitions are unambiguous and easy to understand.
- The annotation of the data requires as little unique expertise as possible.
- There is agreement on the correct labelling among different human annotators.
- The label semantics are unlikely to change over time.
- The labels may differ from the target variable that is predicted by the ML algorithm.
- However, they may correlate strongly with the target variable.
- Existing algorithms can handle the labels relatively well.



Total Price:
SGD 5.5

Vegetable x2
Egg x1
Meat x1

Potato x1
Egg x1
Chicken x1
Okra x1

(a) Annotation Type 1 (b) Annotation Type 2 (c) Annotation Type 3 (d) Annotation Type 4

Figure 3. Different types of data annotations for an automated cashier

In practice, these considerations are rarely achieved simultaneously and may sometimes clash with each other, so some trade-offs are inevitable. Proper trade-off decisions rely on a good understanding of the application domain in addition to ML.

- **Example 1**: A mixed-rice stall at a food court wants to use an ML system that automatically charges a customer based on a photo of the food on their plate. Figure 2 shows a picture and four possible data annotation styles. Let us analyse the pros and cons of each.
  - Figure 2(a) contains structured annotations that include the bounding boxes for each dish and type. These kinds of data labels are used in object detection. Although this type of annotation provides detailed information, the bounding boxes are challenging to draw compared to the text labels in Figures 2(b) to 2(d) and hence result in higher annotation cost per image. The exact locations are not useful in the calculation of the final price; the location information is useless.
  - The schema in Figure 2(b) is the most straightforward schema possible. However, it is also a poor design choice. Inflation may raise the food price over time, rendering existing labels obsolete. When that happens, the data must be re-annotated. The violation of "stability over time" requirement will lead to high cost over the long run.
  - Figures 2(c) and 2(d) count the types of food on the plate. The price is computed by multiplying the item prices by the counts. Figure 2(c) contains just enough information to calculate the price.
  - Figure 2(d) describes the food more precisely. In most mixed-rice stalls in Singapore, food items in the same category (vegetables or meat) have the same price tag; this may continue in the future. The schema in Figure 2(d) offers the right

balance between annotation complexity and semantic stability and reduces the cost of the ML system over its lifetime.

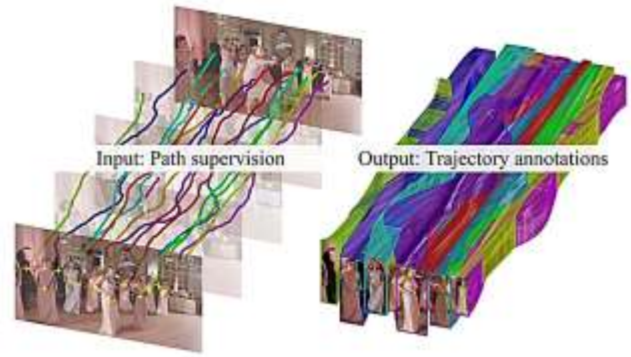


Figure 4. PathTrack [Manen et al., 2017] converts weak annotations in the form of cursor trajectory into bounding box trajectories, which are need for the training of tracking algorithms.

- **Example 2**: Annotation complexity and cost can be reduced by resorting to weak labels. “Weak labels” are correlated with, but differ from, the desired target variable for prediction, due to ambiguity, incompleteness or imprecision. However, in some applications, collecting weak labels is significantly less expensive than annotating the target variable directly. You could then devise computational methods to utilise the correlation between the weak labels and the target variable – to compensate for weak and incomplete labels – and reduce the overall cost.
- PathTrack [Manen et al., 2017] is an example of the use of weak labels. The ML task tracks multiple humans in the video, which involves placing bounding boxes around all humans in every frame and linking them across frames. Instead of putting bounding boxes around every person in every frame, the annotators place the cursor on the same individual and follow them through the video. After that, an integer programming problem assigns the detected bounding boxes to mouse cursor trajectories (Figure 3). Though the recovered trajectories may have small inaccuracies, the significant cost saving provides a compelling reason for this annotation schema.

b. *The Annotation Procedure*

Careful design of the annotation procedure can lead to cost reduction and improvement in label quality. Here's an example with two possibilities, gamification and active learning:

- **Example 3**: The ESP Game [von Anh & Dabbish, 2004] is a classic example of a gamified process for collecting image labels. The game is designed to be played online with human players in pairs. A pair of players receive an image at the same time – and type in a string that describes the picture. A player can type in multiple descriptions. If any pair's descriptions match precisely, they have "won" and can move to the next image.
    - The complexity is that the players are randomly paired online; they have no communication channel. Their best strategy is to describe the images using familiar words. Thus, when the two players agree on a description, there is a high chance that the definition fits the image.
    - To improve labelling accuracy, the same image can be shown to multiple pairs of players and pick the most frequent labels. The ESP Game has taboo words that the players are not allowed to input; this boosts diversity. That forces players to think of different descriptions.
    - The idea of collecting human knowledge in a game has been applied to other problems. Notably, the Fold.it game [Cooper et al., 2010] taps human creativity in a game where players predict how the primary structure of a protein folds in 3D space. Similarly, EteRNA [Lee et al., 2014] is a game where players predict how RNA structures fold.

There are potential ethical concerns with crowdsourcing labels: While the ESP Game is difficult to cheat, a well-coordinated attack by a large number of players – or programs pretending to be players – may insert incorrect labels with the goal to bias the ML algorithm. Therefore, a mechanism that filters vulgarities and other potentially offensive content is a must. The crowd annotation method is also suitable for situations where verifying an answer could be easy, but creating the answer could be complicated.

- **Example 4**: Active learning [Settles 2010] refers to the practice of selecting data points to be annotated based on the needs of the ML algorithm. It starts by annotating a small set of data points and training an ML model on this dataset. Based

  on the analysis of the learned model, you can select additional data points to be annotated, which are also called queries. The chosen data points will thus contain more information for decision-making compared to ones not selected. As a result, you can maintain high predictive accuracy while lowering the annotation cost.
- Active learning can be incorporated into applications such as recommendation engines, where the system can show the user a combination of recommended items and query items. The user's selection would serve as an annotation.
- How to determine the data points that are sent to human annotators? There are several standard methods for this, including selecting the data that the model is the least certain about, the data that a collection of models disagree the most on, the data that cause the most changes in the trained model, or the data that cause the most error reduction on the validation set.

c. Ethical Issue

What could be some ethical issues in active learning? It may involve the asymmetric cost of error and the balance of the dataset. Some ML/AI problems may incur asymmetric costs for false positives/false negatives. For example, an early disease screening procedure may be sensitive to false negatives, because missing a patient with the disease could have severe health consequences. By comparison, classifying a healthy individual as having the illness could result in subsequent medical tests can still catch the mistake.

In such scenarios, the error reduction criterion for active learning should incorporate the asymmetric nature of the error. You may also choose to increase the proportion of positive data points in the annotated data so that you get a slightly heightened prior for positive examples. That may reduce false negatives, but then, increase false positives. Likewise, if the goal is for the model to be sensitive to people of all ethnicities, you may wish to oversample data points from the ethnical minorities.

C. Leveraging Legacy Data

In some application scenarios, acquiring and labelling new data may not be the best course of action to pursue. For example, it may be challenging to find useful data because the application is too new. In some other cases, simply adding more data may not improve the model's accuracy.

Relabelling or data cleaning might be required to allow additional data to make an impact on the model's performance. Alternatively, the AI model for a novel application can be trained from an existing model built for a different application using transfer learning techniques [Pan & Yang, 2010].

A critical issue with legacy data is the problem of noisy or incorrect labels; for this, data cleaning or relabelling will help. Noisy data can result during collecting or lack of proper data validation by some applications. Some examples of noisy data are:

- Out-of-range data (such as day of month mistakenly entered beyond 31).
- Data with different units (such as representing time intervals inconsistently)

○ Crowdsourcing can also pose significant problems based on the crowd worker's proficiency.

Data validation techniques can improve data quality by incorporating proper integrity constraints. The data cleaning system can use quality rules, value correlations and reference data to analyse how data was generated in a probabilistic fashion [Rekatsinas et al., 2017], which can be used to repair the data. There are also interactive data cleaning tools to convert data into a form suitable for ML tasks [Raman & Hellerstein, 2001; Kandel et al., 2011].

Let's discuss some methods for improving existing labels and utilising legacy data.

a. *Transfer Learning*

Transfer learning [Pan & Yang, 2010] is a popular approach. One commonly-adopted approach is to start from a well-trained existing model from the source domain to incrementally train a new model in the target domain, which performs well. Transfer learning can be divided based on what is being transferred from the source-domain model to the target-domain model:

- Instance-based transfer learning: This reuses data samples from the source domain by re-weighting them.
- Feature-representation transfer learning: In this, the features that represent the data from the source domain are used to represent the data in the target domain.
- Parameter transfer learning: This is designed for cases in which the source and target domains share some model parameters or prior distributions which can be used as bridges of the transfer.
- Relational knowledge transfer learning: Relationships within the source domain data can also be reused in the target domain.

b. *Data Augmentation*

For unstructured data (such as text and images), data augmentation can take different strategies. One such is to derive latent semantics. For textual data, a popular technique is to generate embeddings that represent words, phrases, or sentences.

One of the seminal works here is Word2vec [Mikolov et al., 2013]. For example, a word is represented by a vector of real numbers that captures the linguistic context of the word in the corpus. Word2vec does not address polysemy effectively. For example, “mouse” could mean a rodent or a pointing device attached to a computer.

Later models such as BERT [Devlin et al., 2018] and GPT-2 [Radford et al., 2019] have managed to handle this better by encoding the context of given words, which is achieved by including information about the preceding and succeeding words in the vector. Such a strategy may lead to significant improvements in NLP tasks.

Google’s TensorFlow Hub [Ruder et al., 2019] and the Google Cloud AutoML allow users to adopt transfer learning with a small dataset. These tools reduce the level of complexity in transfer learning techniques so that the data scientists can leverage models trained on legacy data for new ML tasks.

c. Data Privacy

As AI becomes increasingly ubiquitous, governments are becoming increasingly concerned about AI governance and privacy protection. New legislation such as GDPR and PDPA have therefore emerged. The new regulations restrict how legacy data can be reused, such as data collected for a specific purpose shall not be used for other purposes without explicit user approval.

One option to overcome the restrictions: Federated learning or FL [Yang et al., 2019] in which training happens where data is stored, and only the model parameters can be accessed. FL can help AI thrive in privacy-sensitive crowdsourced environments. FL involves owners of local datasets to train ML models collaboratively. In this way, end- users can become co-creators of future AI solutions.

Google’s TensorFlow Federated toolkit is useful for individual users to get familiar with the basics of FL. The Federated AI Technology Enabler (FATE) toolkit developed by WeBank includes a range of modules for handling FL scenarios and is designed for industry-grade FL app development.

Today’s federated model training [McMahan et al., 2016] is based on a critical assumption: After federated model training, all participants receive the same final model regardless of their contribution. This may pose challenges to the adoption of FL, especially under B2B settings.

- **Example**: Banks A, B, and C want to collaboratively train an AI model to predict the creditworthiness of SMEs in a privacy-preserving manner. Bank A is significantly larger than B and C. If all of them receive the same final model under the current FL paradigm. Bank A – which can contribute more to building a high-quality model – may hesitate for fear of benefiting other smaller banks and eroding its own market share.

Recent advances in FL have removed the need of a centrally-trusted entity to coordinate federated model training [Lyu et al., 2020]. Each participant decides the level it wishes to contribute to a given federation. Then it assesses the quality of the local training data of other participants in the collaboration network via mutual evaluation, without looking at the raw data.

Each participant builds a record of the individual local credibility scores of others, which represents how much it can trust the others in each collaborative training round. Eventually, each participant can use the local credibility scores to help it decide with which other participant/s it should form federated model training pairs. Over different rounds of interactions based on mutual trust, each entity will gradually build up its own federated model. These developments provide the data scientist with more technical tools to manage legacy data under different application scenarios.

Parameter exchange alone under FL is not enough to protect data privacy. It has been shown that exchange model parameters in plain text are vulnerable to honest-but- curious participants [Yang et al., 2019]. Thus, additional precautions, such as homomorphic encryption-based protocols, are required to enhance data privacy protection capabilities of federated learning.

III. Concluding Remarks

The bottom line: How can a data scientist leverage existing techniques to deal with various scenarios? Imagine that a data scientist, Bob, is in charge of building a predictive maintenance system powered by AI for a manufacturing plant. In the beginning, when there is little or no data, Bob needs to collect data. He can search for relevant datasets online and look for legacy datasets within the company.

If there are regulatory restrictions for moving data outside the data silos for model training, Bob can adopt FL techniques to circumvent the restrictions. He can also generate an initial dataset by going through the workflow in the manufacturing plant over a period. Once the dataset is procured, Bob can then label the data. If there are enough existing labels, then self- labelling via semi-supervised learning can be used. Else, Bob can engage crowdsourcing workers to label the datasets if he has the budget to do so.

If he does not have the funds, Bob can adopt weak supervision techniques. However, this will mean that the resulting model performance may be reduced due to weak labels. If there is an existing AI model related to Bob's ML task, he can adopt transfer learning to adapt this model to the new task.

References

Allen, M. & Cervo, D. Chapter 9 - Data Quality Management. Editor(s): Allen, M. & Cervo, D. *Multi-Domain Master Data Management, Morgan Kaufmann*, pp. 131-160 (2015).

Choi, E., Biswal, S., Malin, B., Duke, J., Stewart, W. F. & Sun, J. Generating multi-label discrete patient records using generative adversarial networks. In *MLHC*, pp. 286–305 (2017).

Cong, G., Fan, W., Geerts, F., Jia, X. & Ma, S. Improving data quality: Consistency and accuracy. In *VLDB*, pp. 315–326 (2007).

Cooper, S. et al. Predicting protein structures with a multiplayer online game. *Nature*, vol. 466, no. 7307, pp. 756–760 (2010).

Cubuk, E. D., Zoph, B., Mané, D., Vasudevan, V. & Le, Q. V. AutoAugment: Learning Augmentation Strategies from Data. In *CVPR*, pp. 113-123 (2019).

Daniel, F., Kucherbaev, P., Cappiello, C., Benatallah, B. & Allahbakhsh, M. Quality control in crowdsourcing: A survey of quality attributes, assessment techniques, and assurance actions. *ACM Computing Survey*, vol. 51, no. 1, pp. 7:1–7:40 (2018).

Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *CoRR*, arXiv 1810.04805 (2018).

Doan, A., Halevy, A. Y. & Ives, Z. G. *Principles of Data Integration*. Morgan Kaufmann (2012).

Doan, A., Ramakrishnan, R. & Halevy, A. Y. Crowdsourcing systems on the world wide web. *Communcations of the ACM*, vol. 54, no. 4, pp. 86–96 (2011).

Dolatshah, M., Teoh, M., Wang, J. & Pei, J. Cleaning crowdsourced labels using Oracles for statistical classification. *School of Computer Science, Simon Fraser University, Technical Report* (2018).

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. Generative adversarial nets. In *NeurIPS*, pp. 2672–2680 (2014).

Ilyas, I. F. & Chu, X. (2019) *Data Cleaning*. Association for Computing Machinery, New York, NY, USA, p. 285.

Kandel, S., Paepcke, A., Hellerstein, J. M. & Heer, J. Wrangler: interactive visual specification of data transformation scripts. In *CHI*, pp. 3363–3372 (2011).

Krishnan, S., Wang, J., Wu, E., Franklin, M. J. & Goldberg, K. ActiveClean: Interactive data cleaning for statistical modelling. *Proceedings of the VLDB Endowment*, vol. 9, no. 12, pp. 948–959 (2016).

Kumar, A., Naughton, J., Patel, J. M. & Zhu, X. To join or not to join?: Thinking twice about joins before feature selection. In *SIGMOD*, pp. 19–34 (2016).

Lee, J. et al. RNA design rules from a massive open laboratory. *Proc. Natl. Acad. Sci.*, vol. 111, no. 6, pp. 2122–2127 (2014).

Lyu, L., Yu, J., Nandakumar, K., Li, Y., Ma, X., Jin, J., Yu, H. & Ng, K. S. Towards fair and privacy-preserving federated deep models. *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 11, pp. 2524–2541 (2020).

Manen, S., Gygli, M., Dai, D. & Gool, L. V. PathTrack: Fast trajectory annotation with path supervision. *CoRR*, arXiv170302437 Cs (2017).

McMahan, H. B., Moore, E., Ramage, D. & Arcas, B. A. y. Federated learning of deep networks using model averaging. *CoRR*, arXiv:1602.05629 (2016).

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. & Dean, J. Distributed representations of words and phrases and their compositionality. In *NeurIPS*, pp. 3111–3119 (2013).

Mudgal, S., Li, H., Rekatsinas, T., Doan, A., Park, Y., Krishnan, G., Deep, R., Arcaute, E. & Raghavendra, V. Deep learning for entity matching: A design space exploration. In *SIGMOD*, pp. 19–34 (2018).

Özmen, B. DeepAugment: AutoML for Data Augmentation. https://blog.insightdatascience.com/automl-for-data-augmentation-e87cf692c366 (2019).

Pan, S. J. & Yang, Q. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359 (2010).

Patki, N., Wedge, R. & Veeramachaneni, K. The synthetic data vault. In *DSAA*, pp. 399–410 (2016).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, E. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830 (2011).

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. & Sutskever, I. Language Models are Unsupervised Multitask Learners. Technical report, OpenAI, Apr. (2019).

Raman, V. & Hellerstein, J. M. Potter's wheel: An interactive data cleaning system. In *VLDB*, pp. 381–390 (2001).

Rekatsinas, T., Chu, X., Ilyas, I. F. & Re, C. Holoclean: Holistic data repairs with probabilistic inference. *Proceedings of the VLDB Endowment*, vol. 10, no. 11, pp. 1190–1201 (2017).

Rich, A. S., Rudin, C., Jacoby, D. M. P., Freeman, R., Wearn, O. R., Shevlin, H., Dihal, K., ÓhÉigeartaigh, S. S., Butcher, J., Lippi, M., Palka, P., Torroni, P., Wongvibulsin, S., Begoli, E., Schneider, G., Cave, S., Sloane, M., Moss, E., Rahwan, I., Goldberg, K., Howard, D., Floridi, L. & Stilgoe, J. AI reflections in 2019. *Nature Machine Intelligence*, vol. 2, pp. 2–9 (2020).

Roh, Y., Heo, G. & Whang, S. E. A Survey on data collection for ML: A big aata - AI integration perspective. *IEEE Transactions on Knowledge and Data Engineering*, vol. PP, no. 99, doi: 10.1109/TKDE.2019.2946162 (2019).

Ruder, S., Peters, M. E., Swayamdipta, S. & Wolf, T. Transfer learning in natural language processing. In *NAACL-HLT*, pp. 15–18 (2019).

Settles, B. Active Learning Literature Survey. *Department of Computer Sciences, University of Wisconsin-Madison* (2010).

Shah, V., Kumar, A. & Zhu, X. Are key-foreign key joins safe to avoid when learning high-capacity classifiers? *Proceedings of the VLDB Endowment*, vol. 11, no. 3, pp. 366–379 (2017).

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P. & de Freitas, N. Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148-175 (2016).

Sheng, V. S., Provost, F. & Ipeirotis, P. G. Get another label? Improving data quality and data mining using multiple, noisy labellers. In *KDD*, pp. 614–622 (2008).

Stonebraker, M., & Ilyas, I. F. Data integration: The current status and the way forward. *IEEE Data Engineering Bulletin*, vol. 41, no. 2, pp. 3–9 (2018).

Subramanian, S., Rajeswar, S., Sordoni, A., Trischler, A., Courville, A. & Pal, C. Towards text generation with adversarially learned neural outlines. In *NeurIPS*, pp. 7562–7574 (2018).

von Ahn, L & Dabbish, L. Labeling images with a computer game. In *CHI*, pp. 319–326 (2004).

Yakout, M., Ganjam, K., Chakrabarti, K. & Chaudhuri, S. Infogather: entity augmentation and attribute discovery by holistic matching with web tables. In *SIGMOD*, pp. 97–108 (2012).

Yang, Q., Liu, Y., Cheng, Y., Kang, Y., Chen, T. & Yu, H. (2019) *Federated Learning*. Morgan & Claypool Publishers, p. 207.