Machine Learning for Healthcare

HST.956, 6.S897

Lecture 24: Robustness to dataset shift

David Sontag

CSAIL

imes
INSTITUTE FOR MEDICAL
ENGINEERING & SCIENCE

HST
HEALTH SCIENCES
& TECHNOLOGY

Course announcements

- Please complete the subject evaluation for this class https://registrar.mit.edu/classes-grades-evaluations/subject-evaluation
- Projects
  - Poster session Tuesday, May 14th from 5-7pm in 34-401
  - Send posters to print by **Monday, 9am**!
  - Final report due end of day, Thursday May 16th
- Grading
  - PS5 & PS6 will be graded by early next week
  - Please let us know immediately if you see any mistakes with grading

Machine learning is brittle

- So, you train your ML model and do a prospective evaluation at your institution → all looks good!
- What could go wrong at time of deployment?
  - Adversarial perturbations of inputs
  - Natural changes in the data (e.g. from transferring to a new place, or non-stationarity)

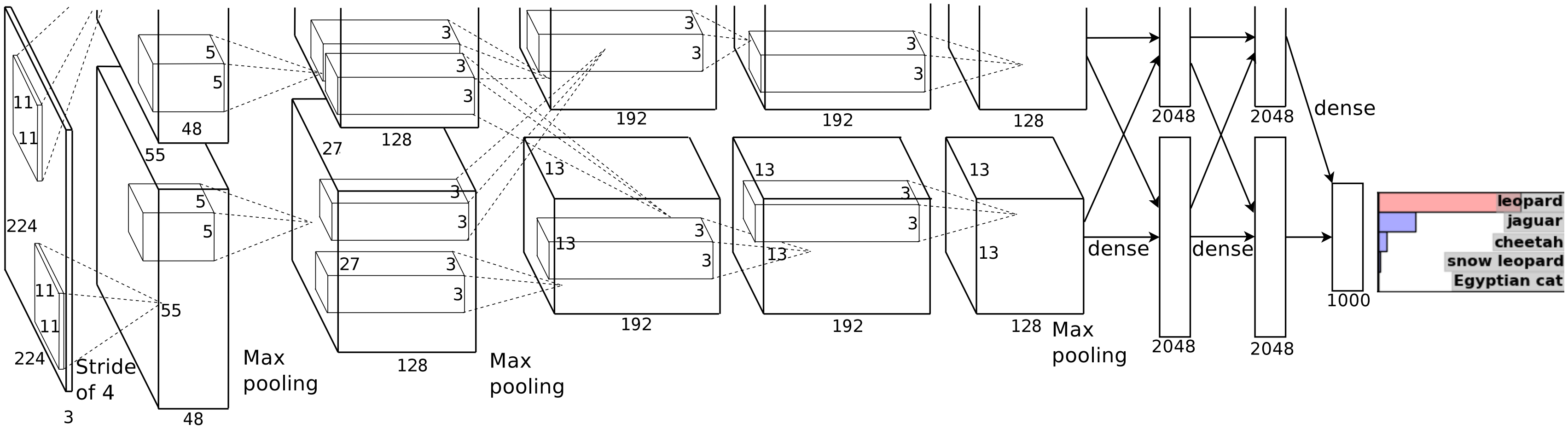
Machine learning breaks when test distribution ≠ train distribution

Machine learning is brittle: adversarial perturbations

Consider a deep neural network used for image classification

Input:

Output:



[Krizhevsky, Sutskever, Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”, NIPS ’12]

Machine learning is brittle: adversarial perturbations

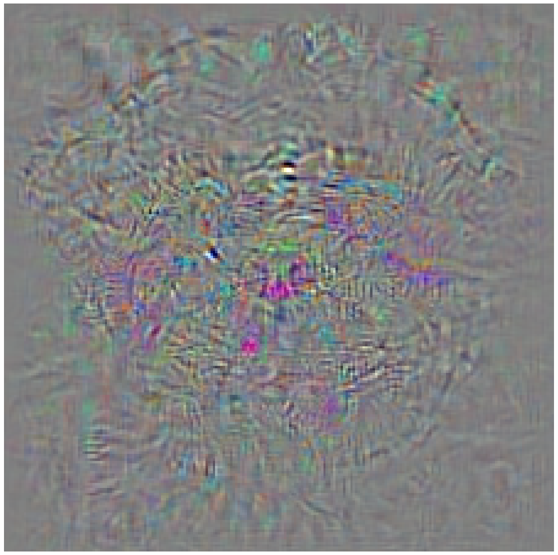
Correctly classified as a Dog

[Szegedy et al., “Intriguing properties of neural networks”, ICLR 2014]

Machine learning is brittle: adversarial perturbations

Original image

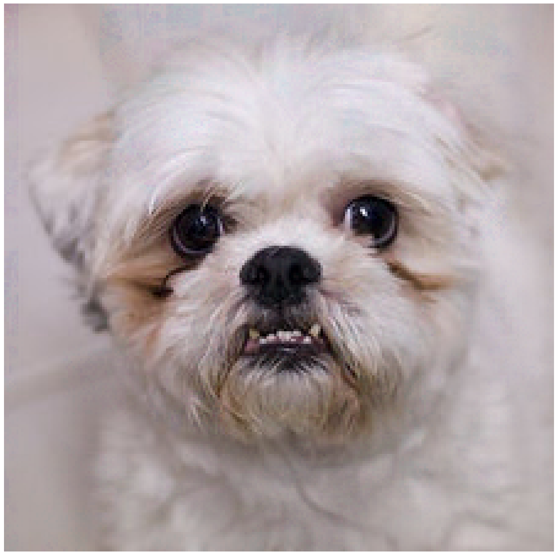
+



Noise (not random)

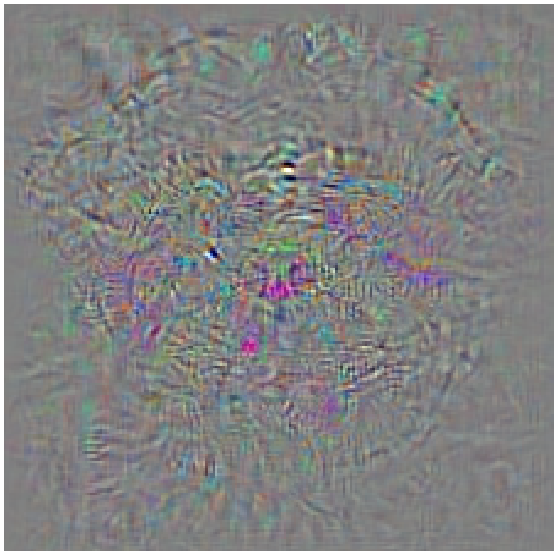
[Szegedy et al., “Intriguing properties of neural networks”, ICLR 2014]

Machine learning is brittle: adversarial perturbations



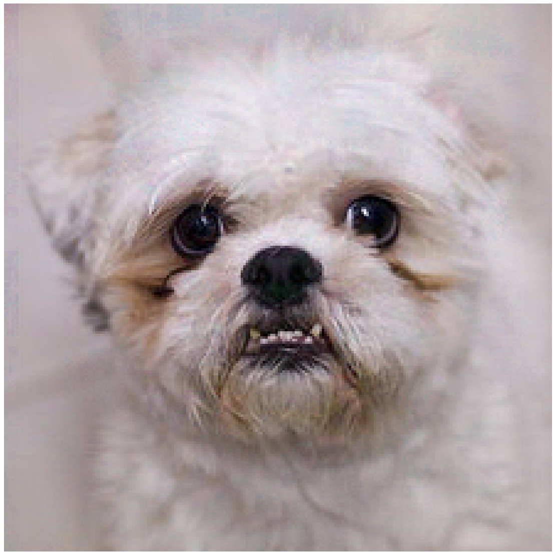
Original image

+



Noise (not random)

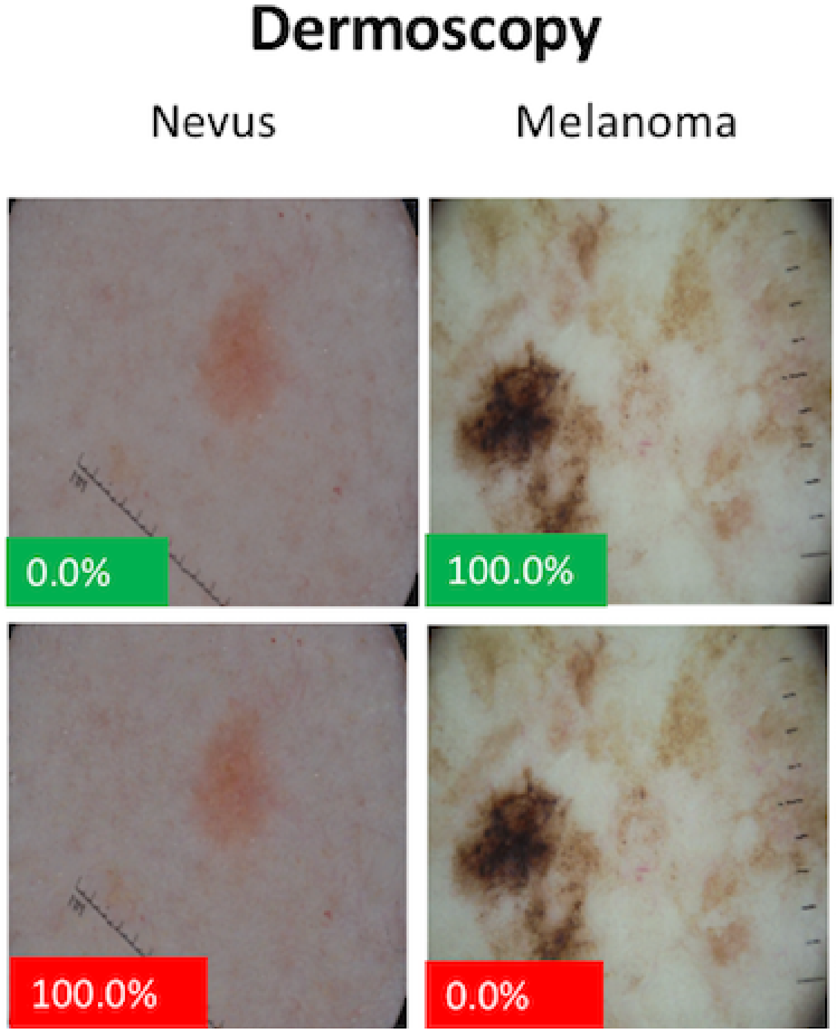
=



Classified as Ostrich!

[Szegedy et al., “Intriguing properties of neural networks”, ICLR 2014]

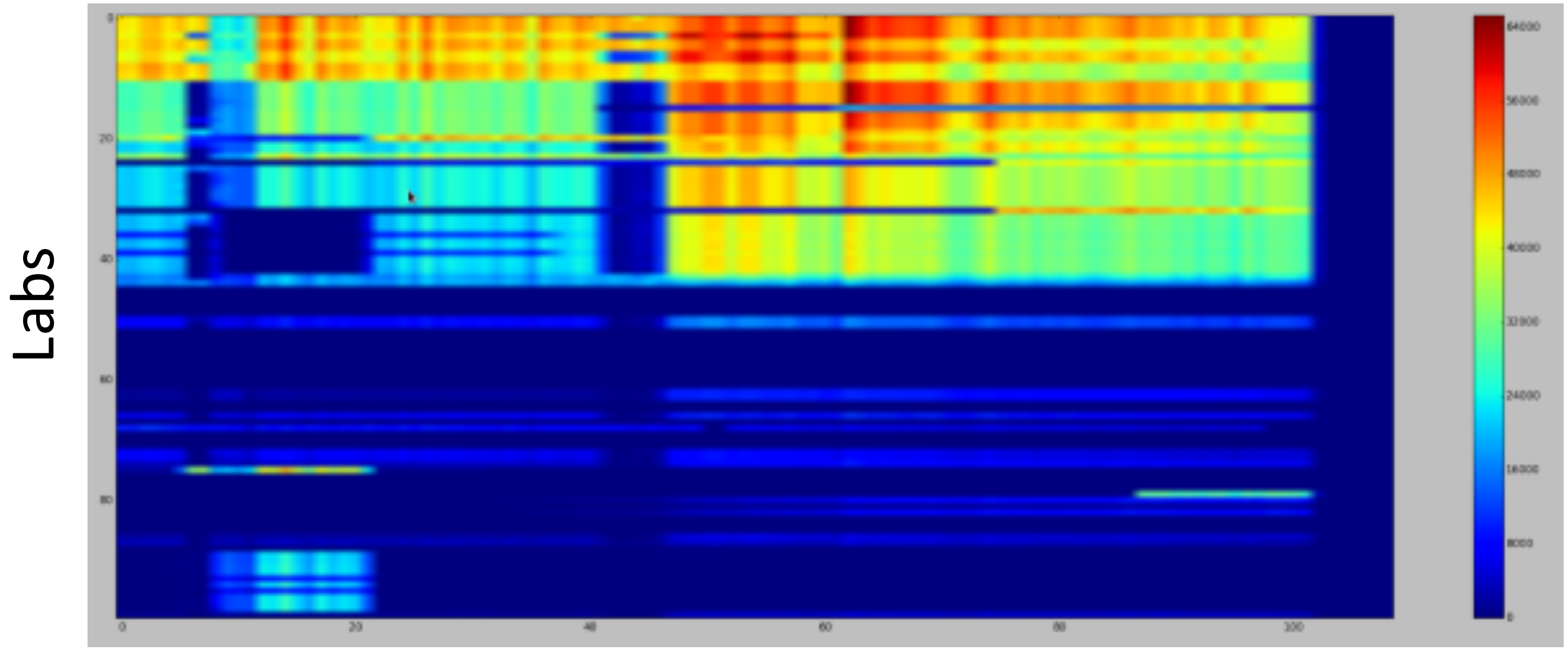
Machine learning is brittle: adversarial perturbations



[Finlayson et al., “Adversarial Attacks Against Medical Deep Learning Systems”, Arxiv 1804.05296, 2018]

Machine learning is brittle: natural changes in the data

Top 100 lab measurements over time

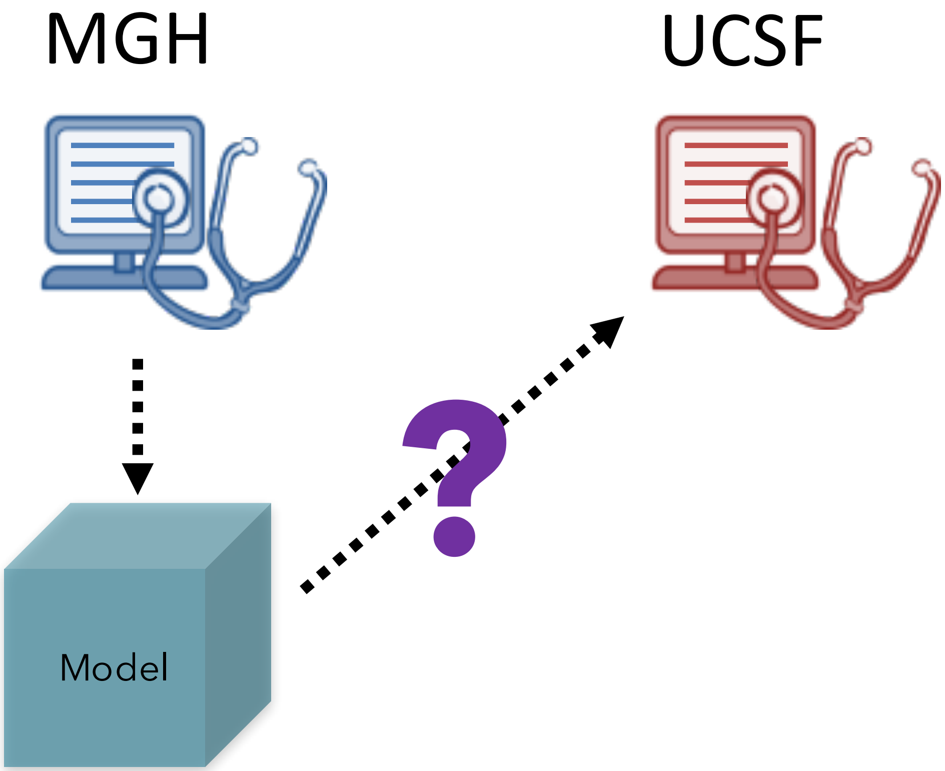


Time (in months, from 1/2005 up to 1/2014)

→ Significance of features may change over time (Figure from Lecture 5)

[Figure credit: Narges Razavian]

Machine learning is brittle: natural changes in the data



[Figure adopted from Jen Gong and Tristan Naumann]

Outline for lecture

1. **Building population-level checks into deployment/transfer**

2. Machine learning in anticipation of dataset shift
   - *Transfer learning*
   - *Defenses against adversarial attacks*

“Table 1”

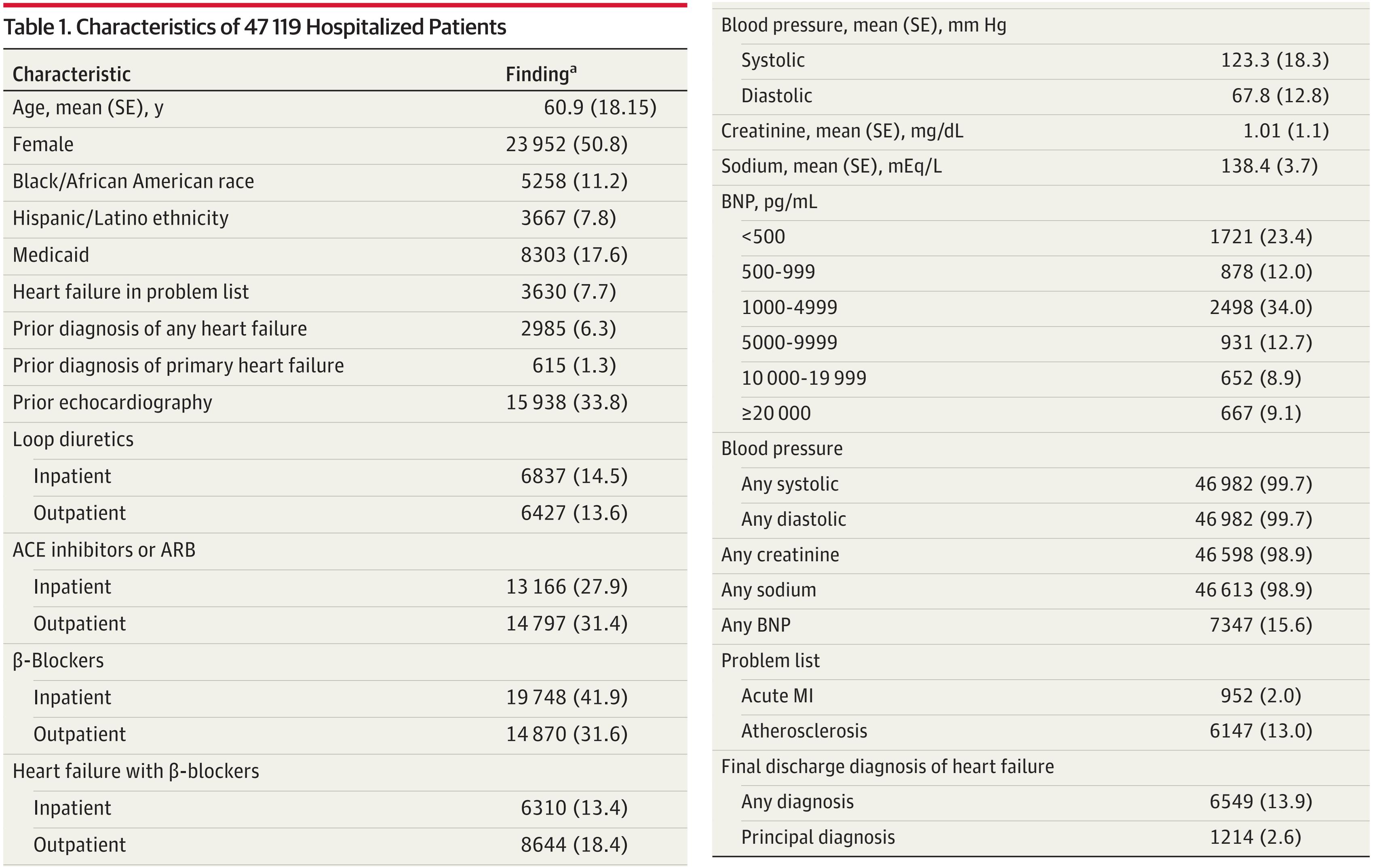
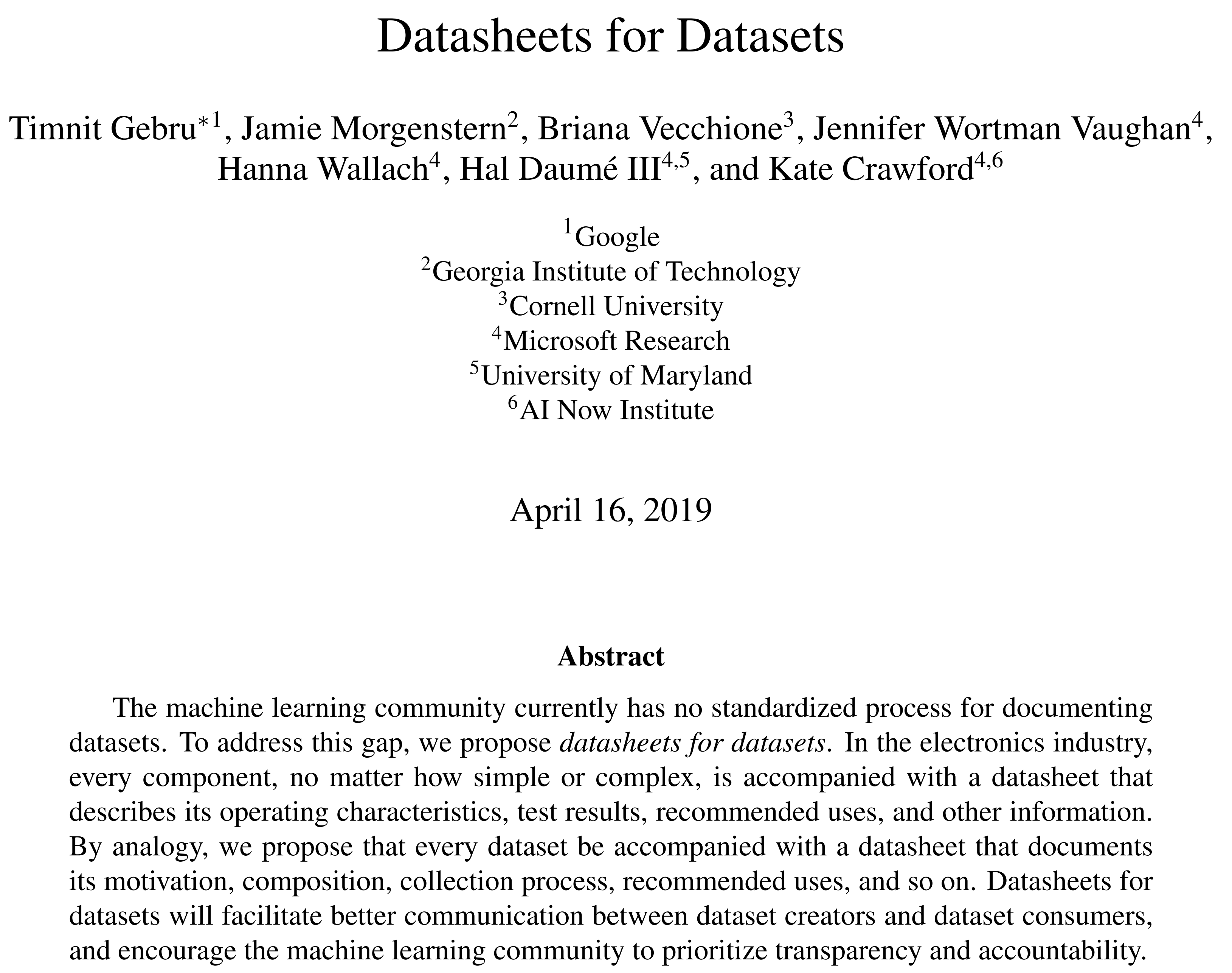


Table 1. Characteristics of 47 119 Hospitalized Patients

Characteristic	Finding[a]
Age, mean (SE), y	60.9 (18.15)
Female	23 952 (50.8)
Black/African American race	5258 (11.2)
Hispanic/Latino ethnicity	3667 (7.8)
Medicaid	8303 (17.6)
Heart failure in problem list	3630 (7.7)
Prior diagnosis of any heart failure	2985 (6.3)
Prior diagnosis of primary heart failure	615 (1.3)
Prior echocardiography	15 938 (33.8)
Loop diuretics	
Inpatient	6837 (14.5)
Outpatient	6427 (13.6)
ACE inhibitors or ARB	
Inpatient	13 166 (27.9)
Outpatient	14 797 (31.4)
β-Blockers	
Inpatient	19 748 (41.9)
Outpatient	14 870 (31.6)
Heart failure with β-blockers	
Inpatient	6310 (13.4)
Outpatient	8644 (18.4)
Blood pressure, mean (SE), mm Hg	
Systolic	123.3 (18.3)
Diastolic	67.8 (12.8)
Creatinine, mean (SE), mg/dL	1.01 (1.1)
Sodium, mean (SE), mEq/L	138.4 (3.7)
BNP, pg/mL	
<500	1721 (23.4)
500-999	878 (12.0)
1000-4999	2498 (34.0)
5000-9999	931 (12.7)
10 000-19 999	652 (8.9)
≥20 000	667 (9.1)
Blood pressure	
Any systolic	46 982 (99.7)
Any diastolic	46 982 (99.7)
Any creatinine	46 598 (98.9)
Any sodium	46 613 (98.9)
Any BNP	7347 (15.6)
Problem list	
Acute MI	952 (2.0)
Atherosclerosis	6147 (13.0)
Final discharge diagnosis of heart failure	
Any diagnosis	6549 (13.9)
Principal diagnosis	1214 (2.6)

[Blecker et al., Comparison of Approaches for Heart Failure Case Identification From Electronic Health Record Data, JAMA Cardiology 2016]



Datasheets for Datasets

Timnit Gebru*[1], Jamie Morgenstern[2], Briana Vecchione[3], Jennifer Wortman Vaughan[4],
Hanna Wallach[4], Hal Daumé III[4,5], and Kate Crawford[4,6]

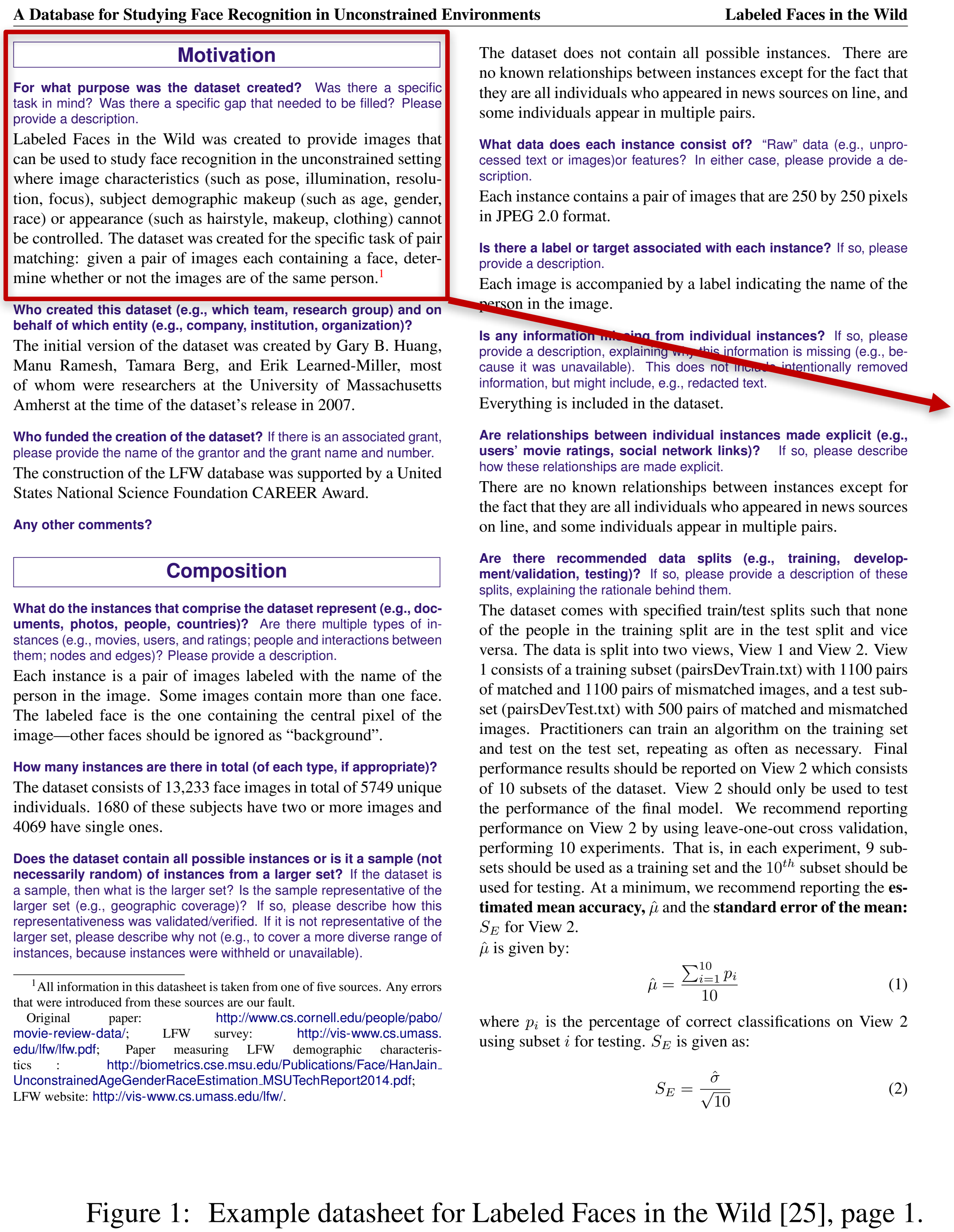
[1]Google
[2]Georgia Institute of Technology
[3]Cornell University
[4]Microsoft Research
[5]University of Maryland
[6]AI Now Institute

April 16, 2019

Abstract

The machine learning community currently has no standardized process for documenting datasets. To address this gap, we propose *datasheets for datasets*. In the electronics industry, every component, no matter how simple or complex, is accompanied with a datasheet that describes its operating characteristics, test results, recommended uses, and other information. By analogy, we propose that every dataset be accompanied with a datasheet that documents its motivation, composition, collection process, recommended uses, and so on. Datasheets for datasets will facilitate better communication between dataset creators and dataset consumers, and encourage the machine learning community to prioritize transparency and accountability.

[Gebru et al., arXiv:1803.09010, 2019]



Motivation

For what purpose was the dataset created? Was there a specific task in mind? Was there a specific gap that needed to be filled? Please provide a description.

Labeled Faces in the Wild was created to provide images that can be used to study face recognition in the unconstrained setting where image characteristics (such as pose, illumination, resolution, focus), subject demographic makeup (such as age, gender, race) or appearance (such as hairstyle, makeup, clothing) cannot be controlled. The dataset was created for the specific task of pair matching: given a pair of images each containing a face, determine whether or not the images are of the same person.[1]

Who created this dataset (e.g., which team, research group) and on behalf of which entity (e.g., company, institution, organization)?

The initial version of the dataset was created by Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller, most of whom were researchers at the University of Massachusetts Amherst at the time of the dataset's release in 2007.

Who funded the creation of the dataset? If there is an associated grant, please provide the name of the grantor and the grant name and number.

The construction of the LFW database was supported by a United States National Science Foundation CAREER Award.

Any other comments?

Composition

What do the instances that comprise the dataset represent (e.g., documents, photos, people, countries)? Are there multiple types of instances (e.g., movies, users, and ratings; people and interactions between them; nodes and edges)? Please provide a description.

Each instance is a pair of images labeled with the name of the person in the image. Some images contain more than one face. The labeled face is the one containing the central pixel of the image—other faces should be ignored as "background".

How many instances are there in total (of each type, if appropriate)?

The dataset consists of 13,233 face images in total of 5749 unique individuals. 1680 of these subjects have two or more images and 4069 have single ones.

Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set? If the dataset is a sample, then what is the larger set? Is the sample representative of the larger set (e.g., geographic coverage)? If so, please describe how this representativeness was validated/verified. If it is not representative of the larger set, please describe why not (e.g., to cover a more diverse range of instances, because instances were withheld or unavailable).

The dataset does not contain all possible instances. There are no known relationships between instances except for the fact that they are all individuals who appeared in news sources on line, and some individuals appear in multiple pairs.

What data does each instance consist of? "Raw" data (e.g., unprocessed text or images)or features? In either case, please provide a description.

Each instance contains a pair of images that are 250 by 250 pixels in JPEG 2.0 format.

Is there a label or target associated with each instance? If so, please provide a description.

Each image is accompanied by a label indicating the name of the person in the image.

Is any information missing from individual instances? If so, please provide a description, explaining why this information is missing (e.g., because it was unavailable). This does not include intentionally removed information, but might include, e.g., redacted text.

Everything is included in the dataset.

Are relationships between individual instances made explicit (e.g., users' movie ratings, social network links)? If so, please describe how these relationships are made explicit.

There are no known relationships between instances except for the fact that they are all individuals who appeared in news sources on line, and some individuals appear in multiple pairs.

Are there recommended data splits (e.g., training, development/validation, testing)? If so, please provide a description of these splits, explaining the rationale behind them.

The dataset comes with specified train/test splits such that none of the people in the training split are in the test split and vice versa. The data is split into two views, View 1 and View 2. View 1 consists of a training subset (pairsDevTrain.txt) with 1100 pairs of matched and 1100 pairs of mismatched images, and a test subset (pairsDevTest.txt) with 500 pairs of matched and mismatched images. Practitioners can train an algorithm on the training set and test on the test set, repeating as often as necessary. Final performance results should be reported on View 2 which consists of 10 subsets of the dataset. View 2 should only be used to test the performance of the final model. We recommend reporting performance on View 2 by using leave-one-out cross validation, performing 10 experiments. That is, in each experiment, 9 subsets should be used as a training set and the 10^{th} subset should be used for testing. At a minimum, we recommend reporting the **estimated mean accuracy,** $\hat{\mu}$ and the **standard error of the mean:** S_E for View 2.

$\hat{\mu}$ is given by:

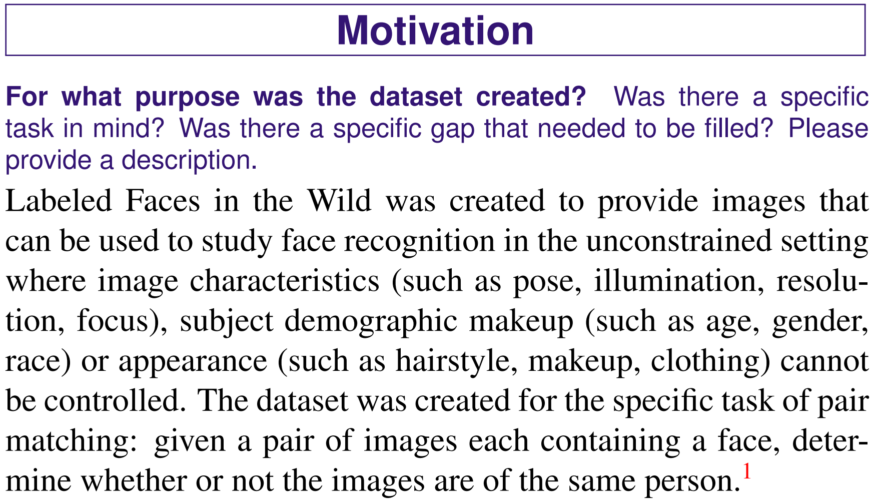
$$\hat{\mu} = \frac{\sum_{i=1}^{10} p_i}{10} \qquad (1)$$

where p_i is the percentage of correct classifications on View 2 using subset i for testing. S_E is given as:

$$S_E = \frac{\hat{\sigma}}{\sqrt{10}} \qquad (2)$$

[1] All information in this datasheet is taken from one of five sources. Any errors that were introduced from these sources are our fault.
Original paper: http://www.cs.cornell.edu/people/pabo/movie-review-data/; LFW survey: http://vis-www.cs.umass.edu/lfw/lfw.pdf; Paper measuring LFW demographic characteristics : http://biometrics.cse.msu.edu/Publications/Face/HanJain_UnconstrainedAgeGenderRaceEstimation_MSUTechReport2014.pdf; LFW website: http://vis-www.cs.umass.edu/lfw/.

Figure 1: Example datasheet for Labeled Faces in the Wild [25], page 1.

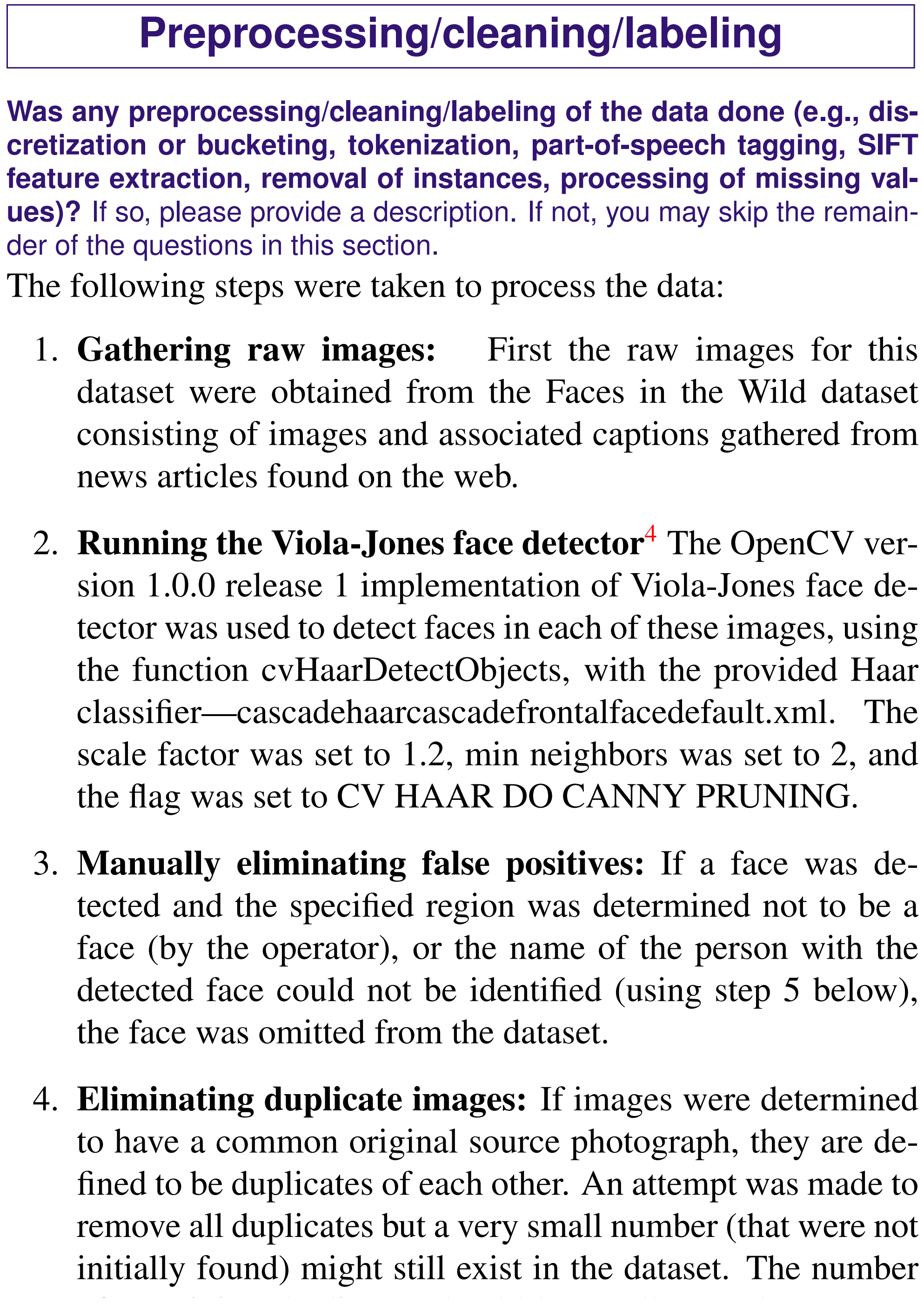


Motivation

For what purpose was the dataset created? Was there a specific task in mind? Was there a specific gap that needed to be filled? Please provide a description.

Labeled Faces in the Wild was created to provide images that can be used to study face recognition in the unconstrained setting where image characteristics (such as pose, illumination, resolution, focus), subject demographic makeup (such as age, gender, race) or appearance (such as hairstyle, makeup, clothing) cannot be controlled. The dataset was created for the specific task of pair matching: given a pair of images each containing a face, determine whether or not the images are of the same person.[1]

[Gebru et al., arXiv:1803.09010, 2019]



Preprocessing/cleaning/labeling

Was any preprocessing/cleaning/labeling of the data done (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)? If so, please provide a description. If not, you may skip the remainder of the questions in this section.

The following steps were taken to process the data:

1. **Gathering raw images:** First the raw images for this dataset were obtained from the Faces in the Wild dataset consisting of images and associated captions gathered from news articles found on the web.

2. **Running the Viola-Jones face detector**[4] The OpenCV version 1.0.0 release 1 implementation of Viola-Jones face detector was used to detect faces in each of these images, using the function cvHaarDetectObjects, with the provided Haar classifier—cascadehaarcascadefrontalfacedefault.xml. The scale factor was set to 1.2, min neighbors was set to 2, and the flag was set to CV HAAR DO CANNY PRUNING.

3. **Manually eliminating false positives:** If a face was detected and the specified region was determined not to be a face (by the operator), or the name of the person with the detected face could not be identified (using step 5 below), the face was omitted from the dataset.

4. **Eliminating duplicate images:** If images were determined to have a common original source photograph, they are defined to be duplicates of each other. An attempt was made to remove all duplicates but a very small number (that were not initially found) might still exist in the dataset. The number

[Gebru et al., arXiv:1803.09010, 2019]

Outline for lecture

1. Building population-level checks into deployment/transfer

2. Machine learning in anticipation of dataset shift
   - ***Transfer learning***
   - *Defenses against adversarial attacks*

Transfer learning

- We have a lot of data from p(x,y) **and** a little data from q(x,y)
- How can we quickly adapt?
  1. Linear models: original representation, modify weights
  2. Linear models: manually choose a good shared representation
  3. Deep models: re-use part of the learned representation, fine-tune
  4. Deep models: automatically find a good shared representation

Transfer learning

- We have a lot of data from p(x,y) **and** a little data from q(x,y)
- How can we quickly adapt?
  1. Linear models: original representation, modify weights
  2. Linear models: manually choose a good shared representation
  3. Deep models: re-use part of the learned representation, fine-tune
  4. Deep models: automatically find a good shared representation

Transfer learning for linear models

- Learn w_{old} using data drawn from p(x,y)
- Then, when learning using data from q, instead of using typical L1 or L2 regularization, use:

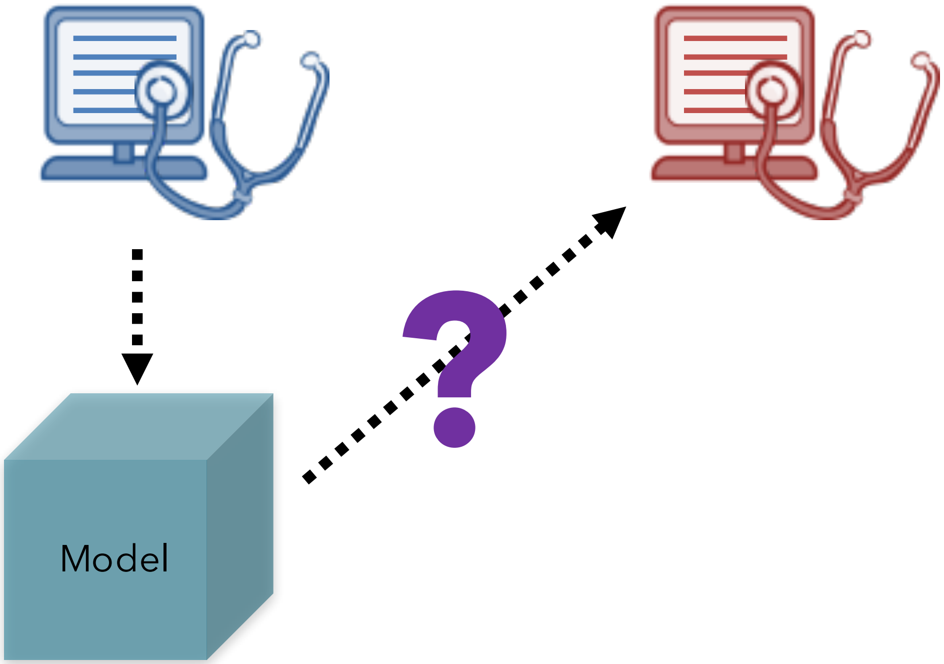
$$||w - w_{\text{old}}||_2^2 \quad \text{or} \quad ||w - w_{\text{old}}||_1$$

- Same as what we previously discussed for *multi-task learning* in the context of disease progression modeling

Transfer learning

- We have a lot of data from p(x,y) **and** a little data from q(x,y)
- How can we quickly adapt?
  1. Linear models: original representation, modify weights
  2. Linear models: manually choose a good shared representation
  3. Deep models: re-use part of the learned representation, fine-tune
  4. Deep models: automatically find a good shared representation

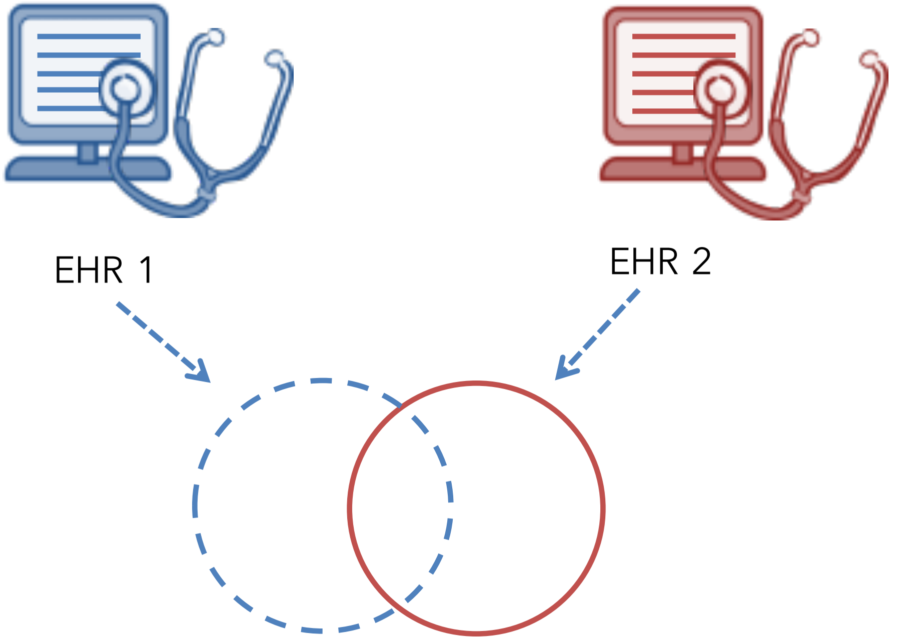
Predicting Clinical Outcomes Across Changing Electronic Health Record Systems



Jen J. Gong, **Tristan Naumann**, Peter Szolovits, John V. Guttag
Computer Science and Artificial Intelligence Laboratory, MIT

KDD 2017

Applying analytics across changing EHR systems is challenging

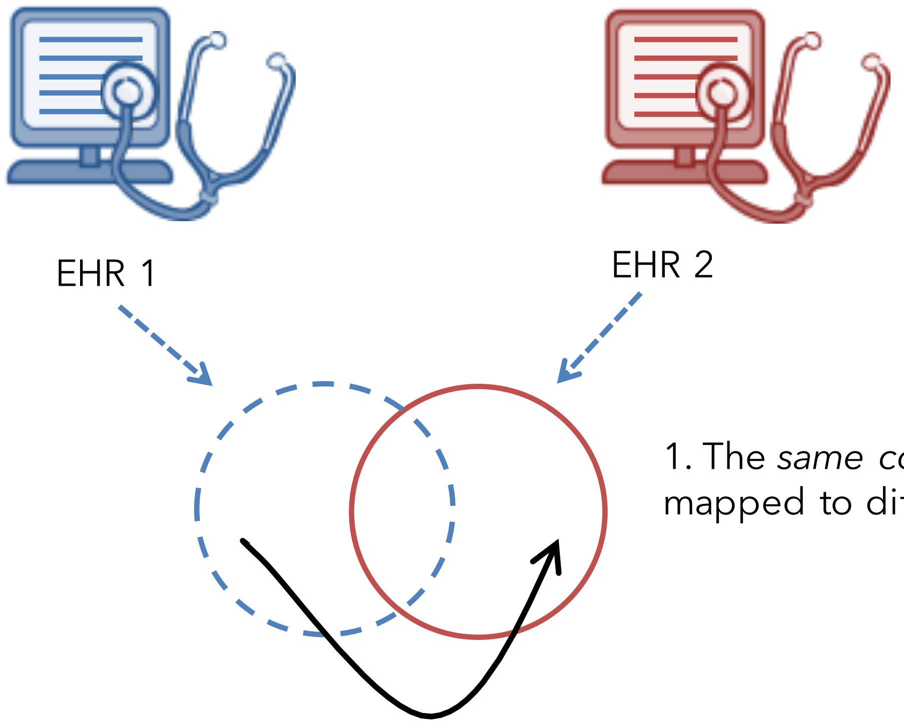


EHR 1

EHR 2

[Slides from Jen Gong and Tristan Naumann on KDD 2017 paper]

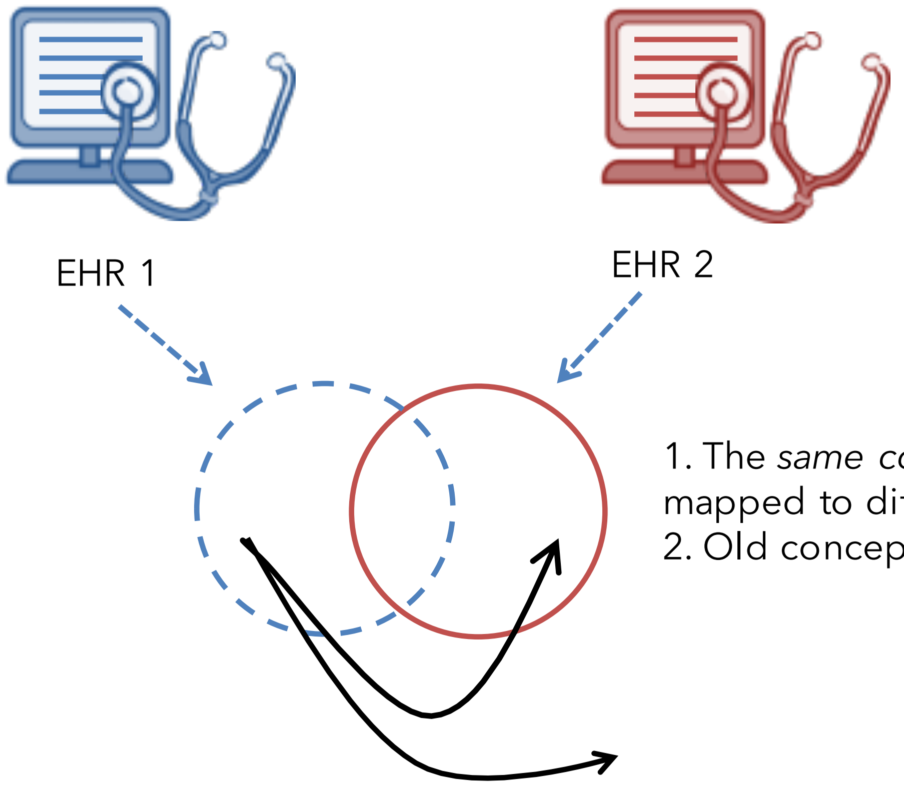
Applying analytics across changing EHR systems is challenging



1. The *same conceptual items* might be mapped to different *encodings*.

[Slides from Jen Gong and Tristan Naumann on KDD 2017 paper]

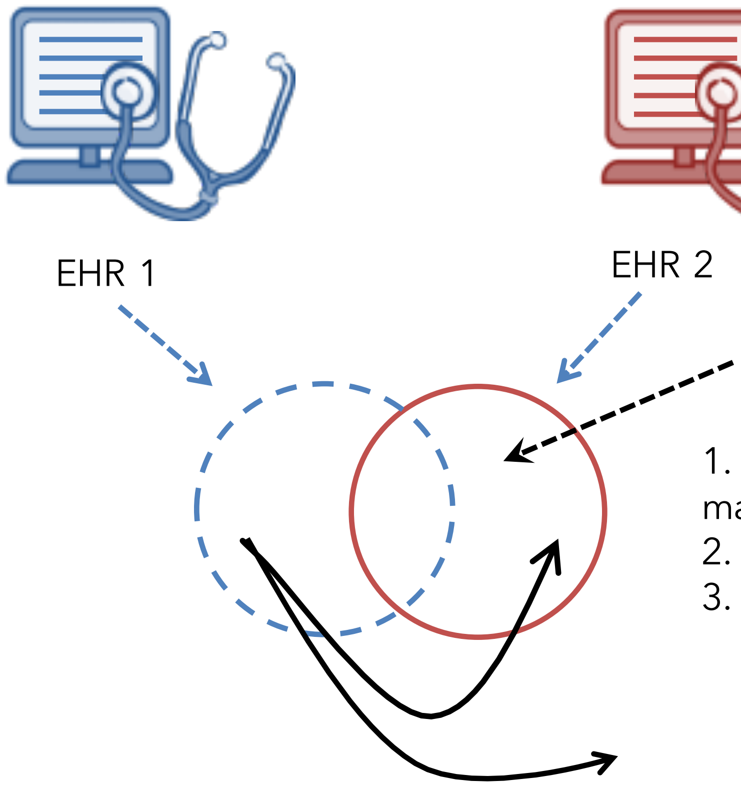
Applying analytics across changing EHR systems is challenging



1. The *same conceptual items* might be mapped to different *encodings*.
2. Old concepts are removed.

[Slides from Jen Gong and Tristan Naumann on KDD 2017 paper]

Applying analytics across changing EHR systems is challenging



EHR 1

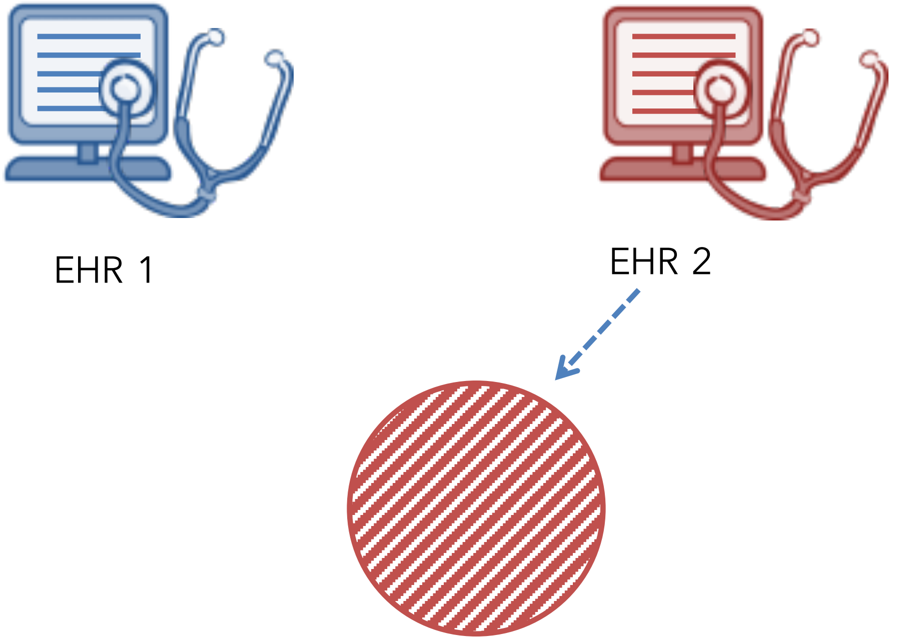
EHR 2

1. The *same conceptual items* might be mapped to different *encodings*.
2. Old concepts are removed.
3. New concepts are added.

. . .

[Slides from Jen Gong and Tristan Naumann on KDD 2017 paper]

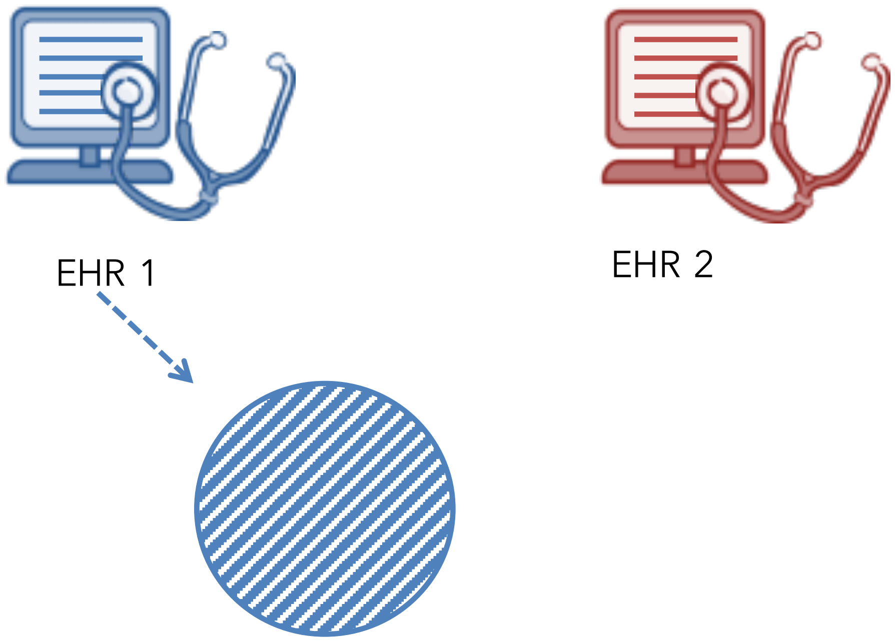
We can learn models using only EHR 2



But this results in throwing away valuable data.

[Slides from Jen Gong and Tristan Naumann on KDD 2017 paper]

We can learn models on EHR 1 and apply them to EHR 2



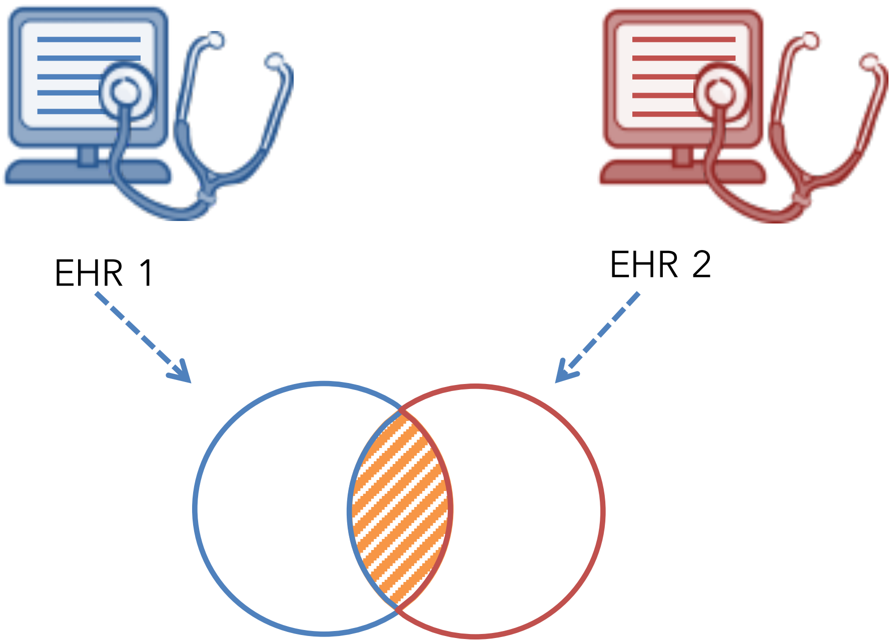
EHR 1

EHR 2

But concepts important in EHR 1 may not appear in EHR 2, and vice versa.

[Slides from Jen Gong and Tristan Naumann on KDD 2017 paper]

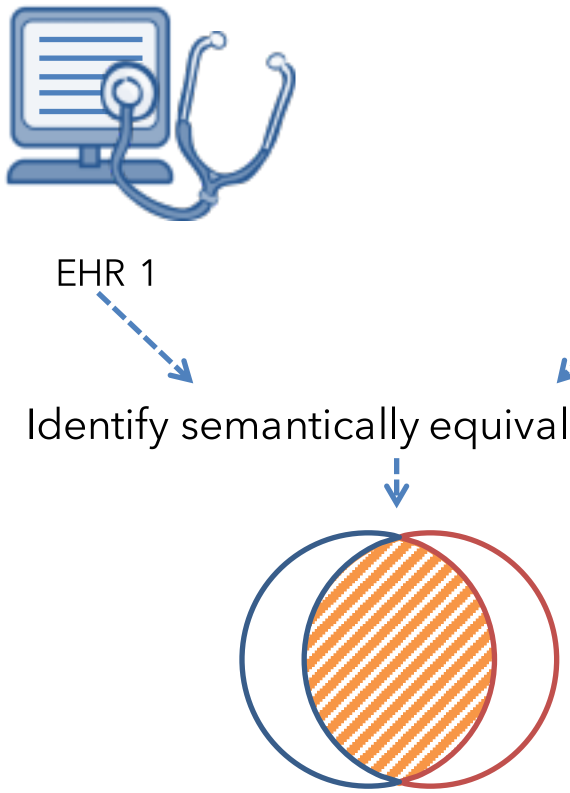
Or, we can develop a model on only the intersection of the elements in EHR 1 and EHR 2



But this could remove the majority of clinical concepts in both EHRs from our model.

[Slides from Jen Gong and Tristan Naumann on KDD 2017 paper]

Solution: Map semantically similar items to a shared vocabulary



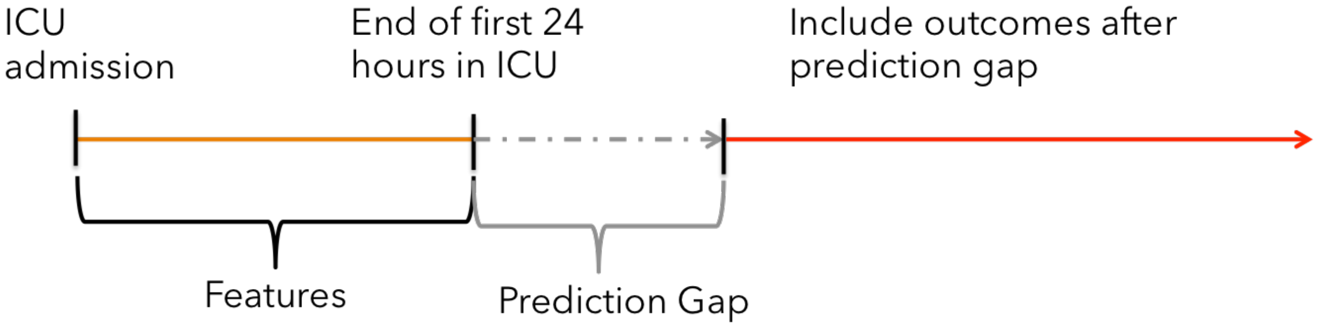
EHR 1

EHR 2

Identify semantically equivalent concepts

[Slides from Jen Gong and Tristan Naumann on KDD 2017 paper]

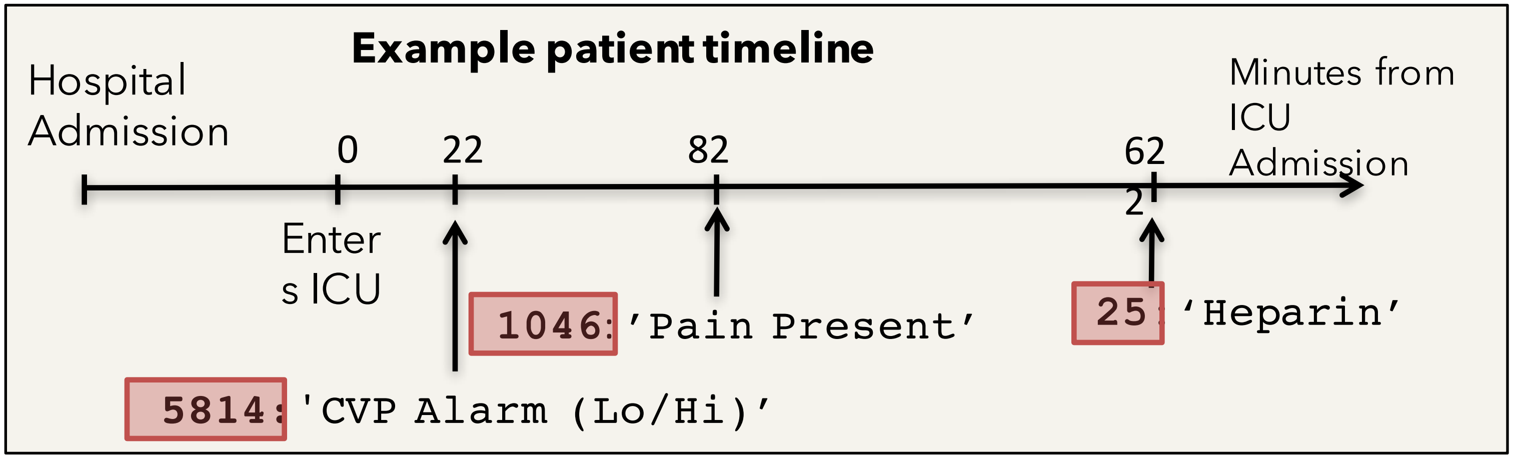
Predictive Models


ICU admission
End of first 24 hours in ICU
Include outcomes after prediction gap
Features
Prediction Gap

Outcomes: (1) In-Hospital Mortality, (2) Prolonged Length of Stay

[Slides from Jen Gong and Tristan Naumann on KDD 2017 paper]

Bag-of-events (BOE)



Example patient timeline

Hospital Admission

0

22

82

62
2

Minutes from ICU Admission

Enter
s ICU

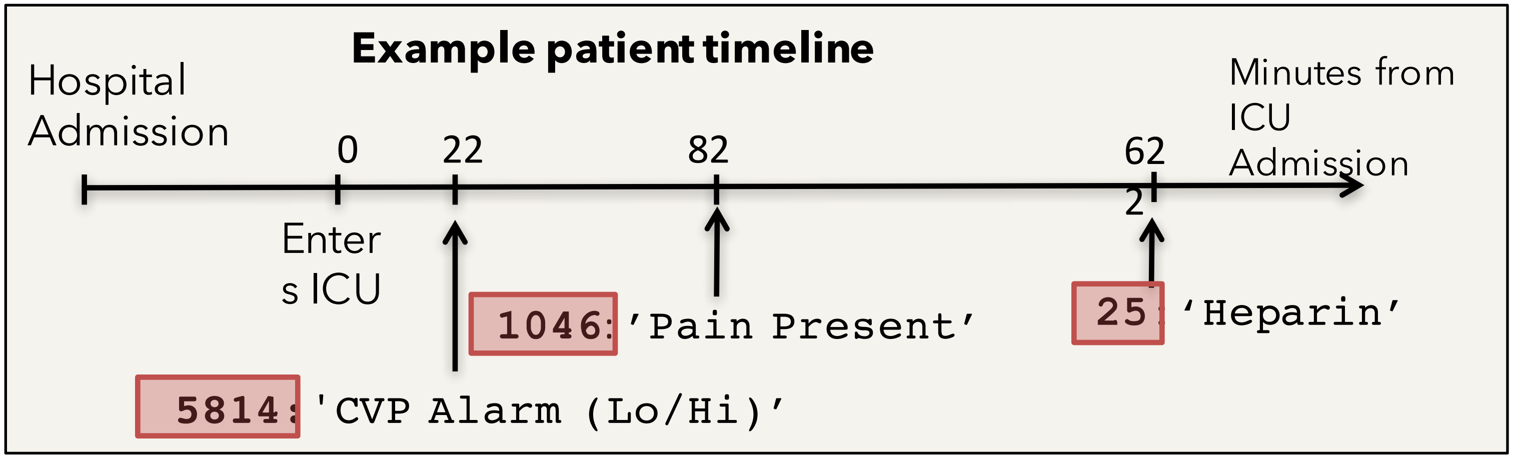
1046: ’Pain Present’

25: ‘Heparin’

5814: 'CVP Alarm (Lo/Hi)’

[Slides from Jen Gong and Tristan Naumann on KDD 2017 paper]

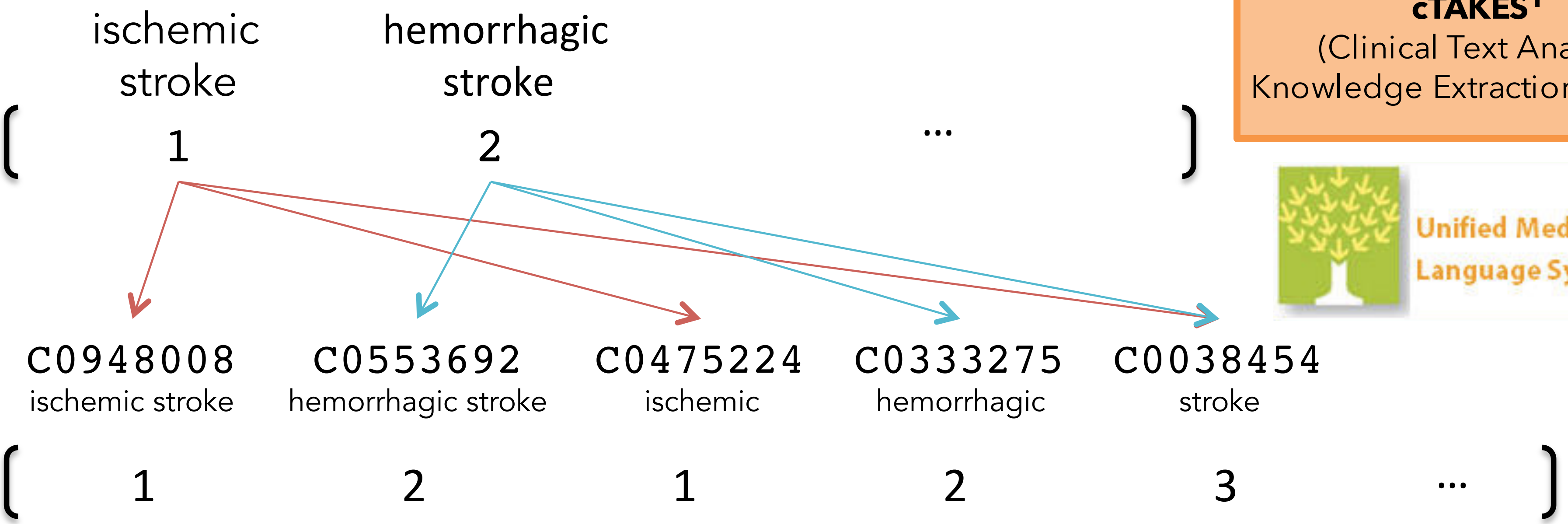
Bag-of-events (BOE)



Item IDs	5814	55	1046	25
Text description	central venous pressure (CVP) alarm	urine out foley	pain present	heparin
BOE	1	0	1	1

[Slides from Jen Gong and Tristan Naumann on KDD 2017 paper]

From EHR-specific events to a shared vocabulary



ischemic stroke | hemorrhagic stroke

(1 2 …)

C0948008 ischemic stroke | C0553692 hemorrhagic stroke | C0475224 ischemic | C0333275 hemorrhagic | C0038454 stroke

(1 2 1 2 3 …)

cTAKES[1]
(Clinical Text Analysis Knowledge Extraction System)

Unified Medical Language System

[1] Savova, G. K. et al. Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation, and applications. JAMIA, 2010.

[Slides from Jen Gong and Tristan Naumann on KDD 2017 paper]

Data & Experimental Setup

- **MIMIC-III dataset**:
  - Publicly available data **from 2 EHR systems** (CareVue and MetaVision) from ICUs.
  - "Item IDs" encode different events (e.g., lab tests, vital signs, medications, other charted observations).
  - Some "Item IDs" are shared between the two EHRs, but the majority are not

- **Models**
  - L2-regularized Logistic Regression, 5-fold cross-validation on training set to determine best hyperparameters

[Slides from Jen Gong and Tristan Naumann on KDD 2017 paper]

Three Experiments

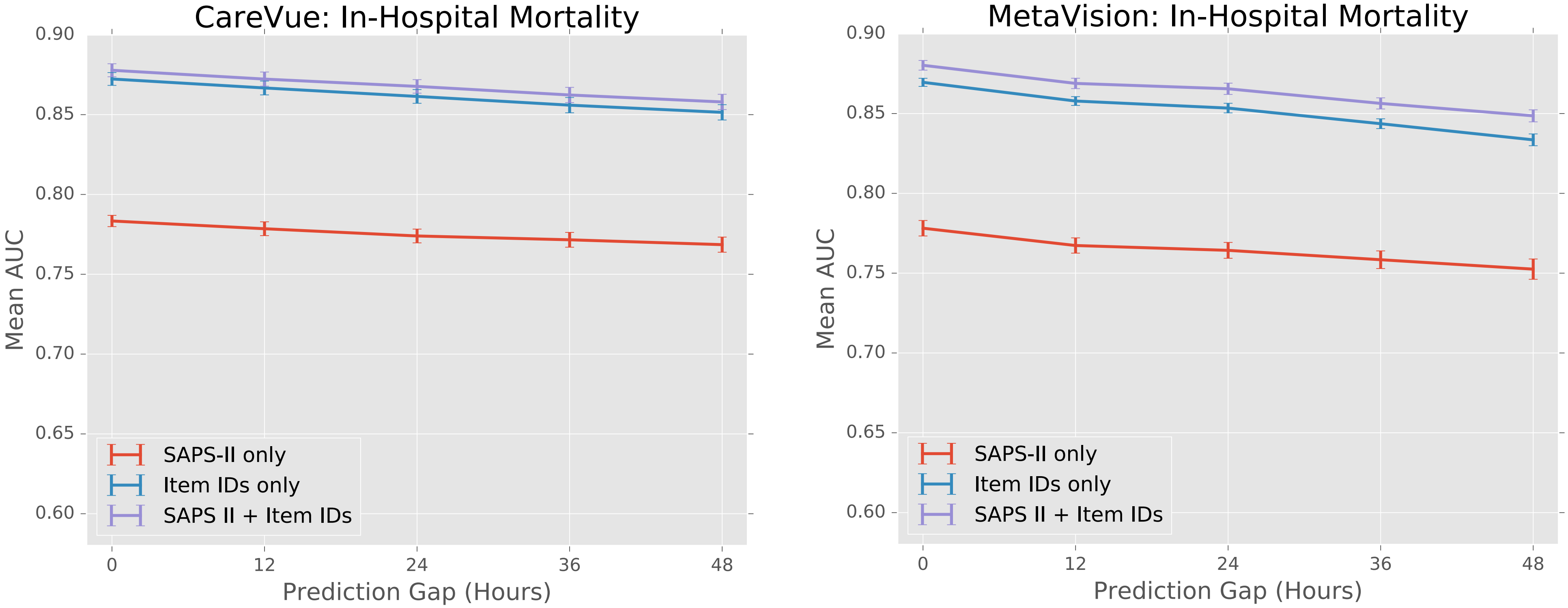
1. Show that a *Bag-of-Events* feature representation is useful in predicting clinical outcomes within each EHR version.

2. Compare performance of semantically equivalent concepts (CUIs) to EHR-specific Item IDs **within EHR versions**.

3. *Compare performance of semantically equivalent concepts (CUIs) to EHR-specific Item IDs* ***across EHR versions****.*

[Slides from Jen Gong and Tristan Naumann on KDD 2017 paper]

Does BOE feature representation have predictive value?



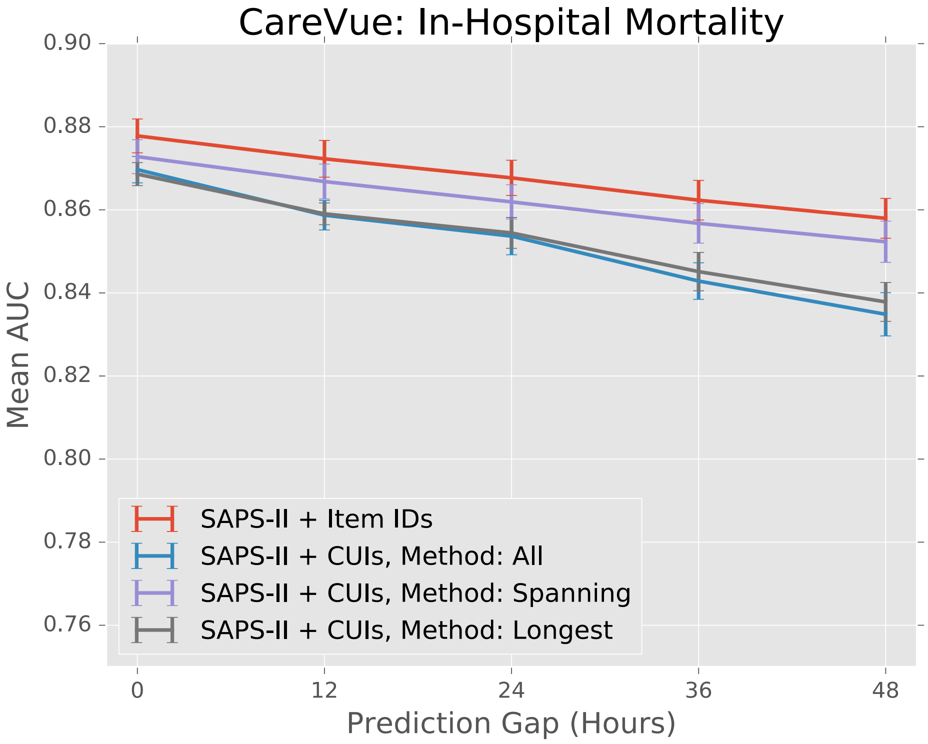
Simplified Acute Physiology Score (SAPS-II): Uses statistics about patient physiology (e.g., heart rate, blood pressure, urine output).

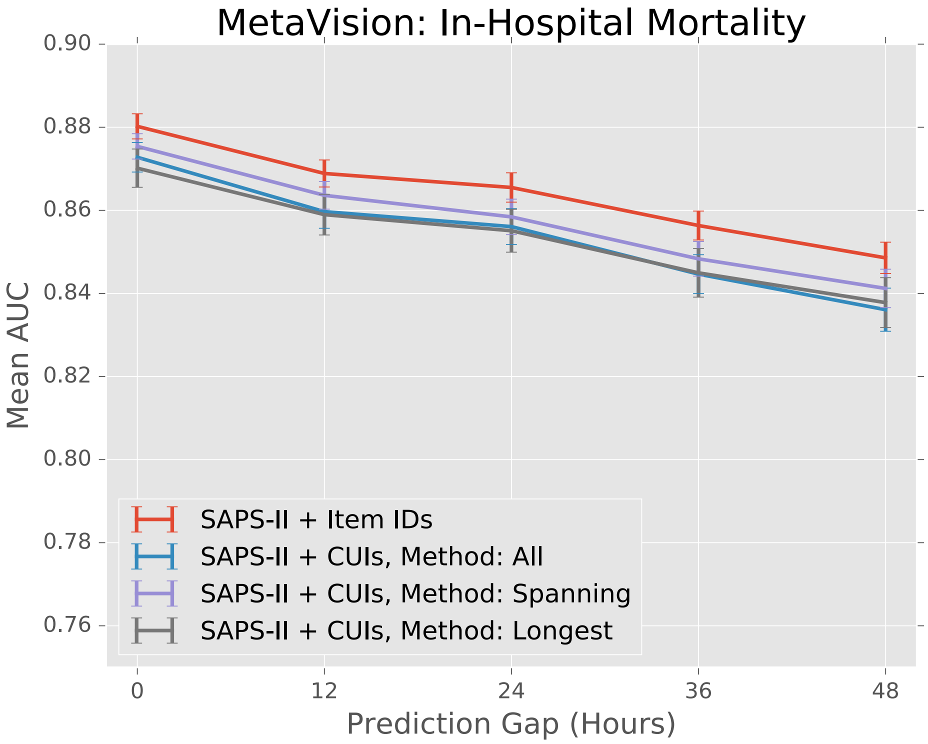
[Slides from Jen Gong and Tristan Naumann on KDD 2017 paper]

What is the impact of mapping BOEs to CUIs within single EHRs?

[Slides from Jen Gong and Tristan Naumann on KDD 2017 paper]

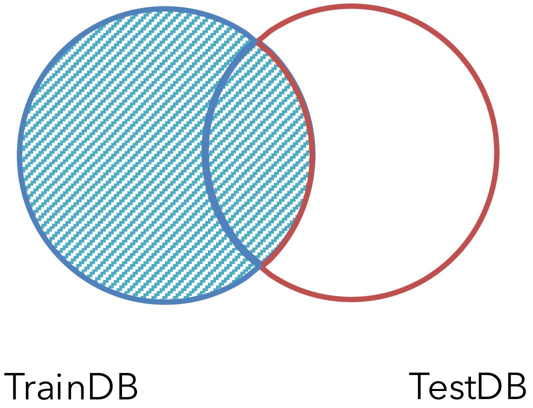
What is the impact of mapping BOEs to CUIs within single EHRs?



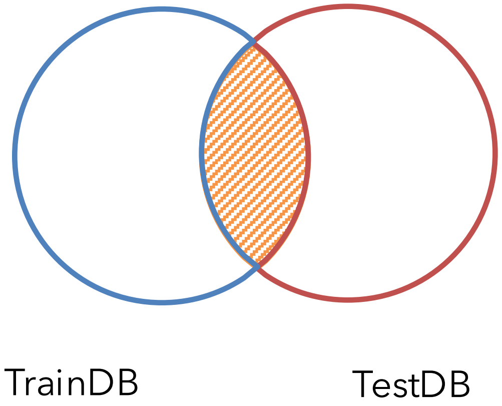


[Slides from Jen Gong and Tristan Naumann on KDD 2017 paper]

What happens when we apply models across EHRs?



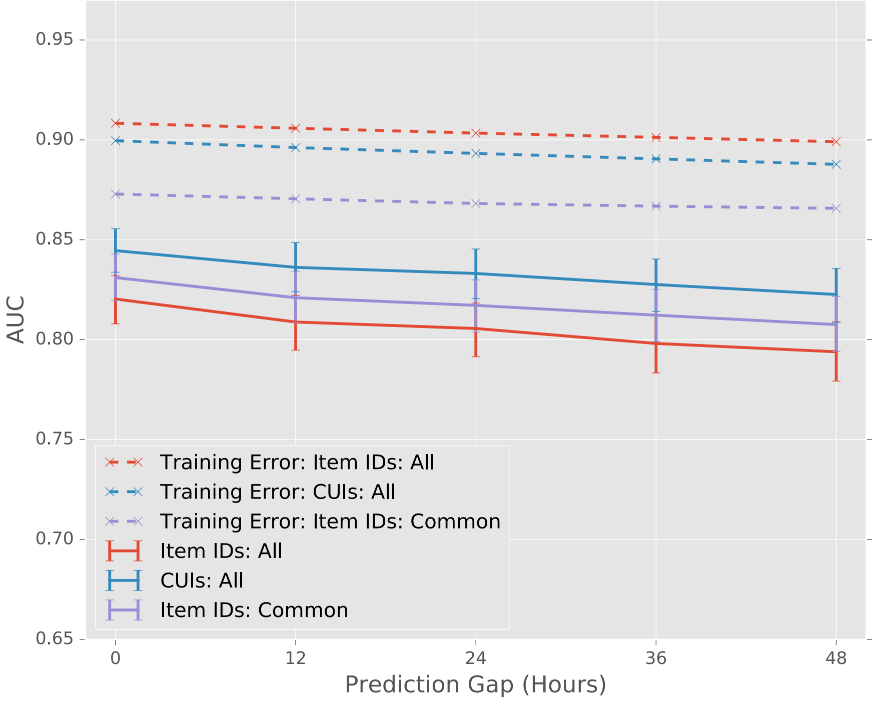
Baseline 1: all

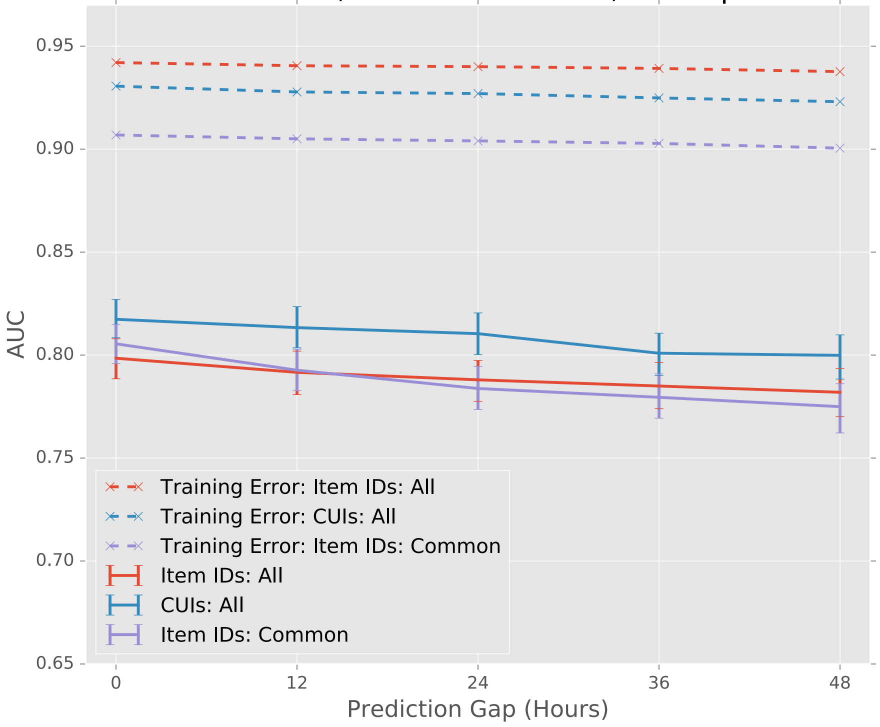


Baseline 2: common

[Slides from Jen Gong and Tristan Naumann on KDD 2017 paper]

What happens when we apply models across EHRs?


Train DB: CareVue, Test DB: MetaVision, In-Hospital Mortality


Train DB: MetaVision, Test DB: CareVue, In-Hospital Mortality

[Slides from Jen Gong and Tristan Naumann on KDD 2017 paper]

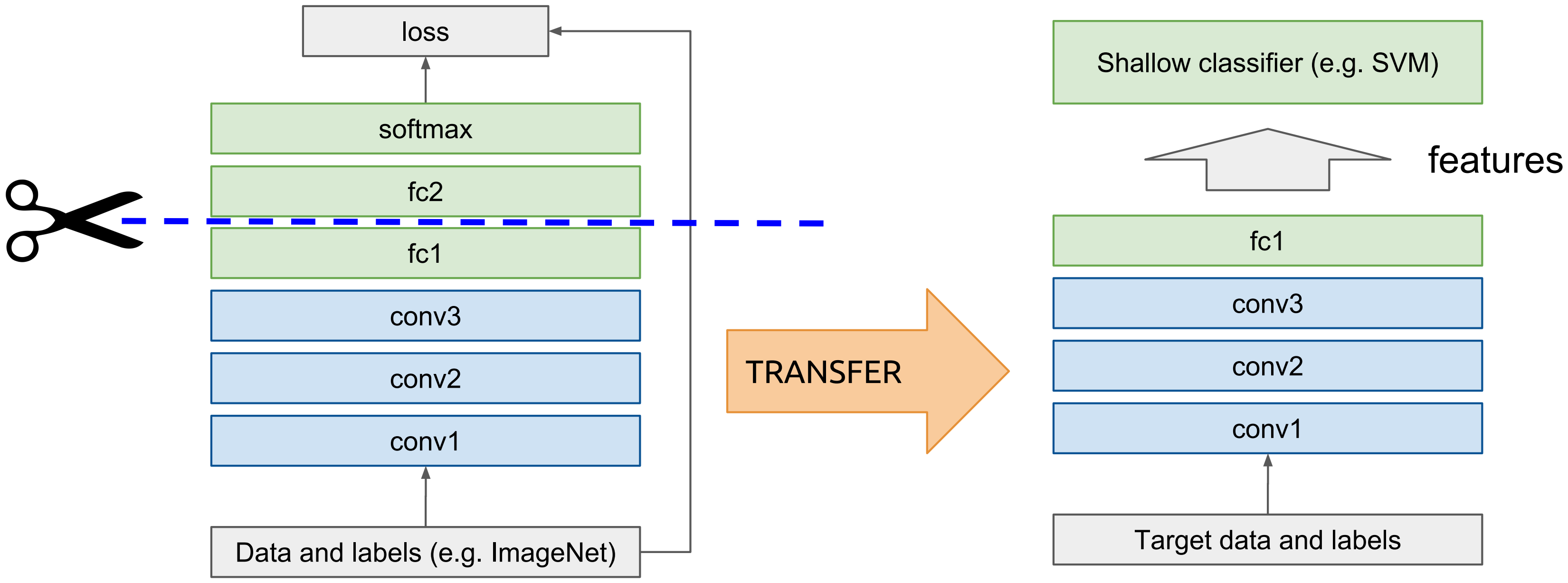
Transfer learning

- We have a lot of data from p(x,y) **and** a little data from q(x,y)
- How can we quickly adapt?

1. Linear models: original representation, modify weights
2. Linear models: manually choose a good shared representation
3. Deep models: re-use part of the learned representation, fine-tune
4. Deep models: automatically find a good shared representation

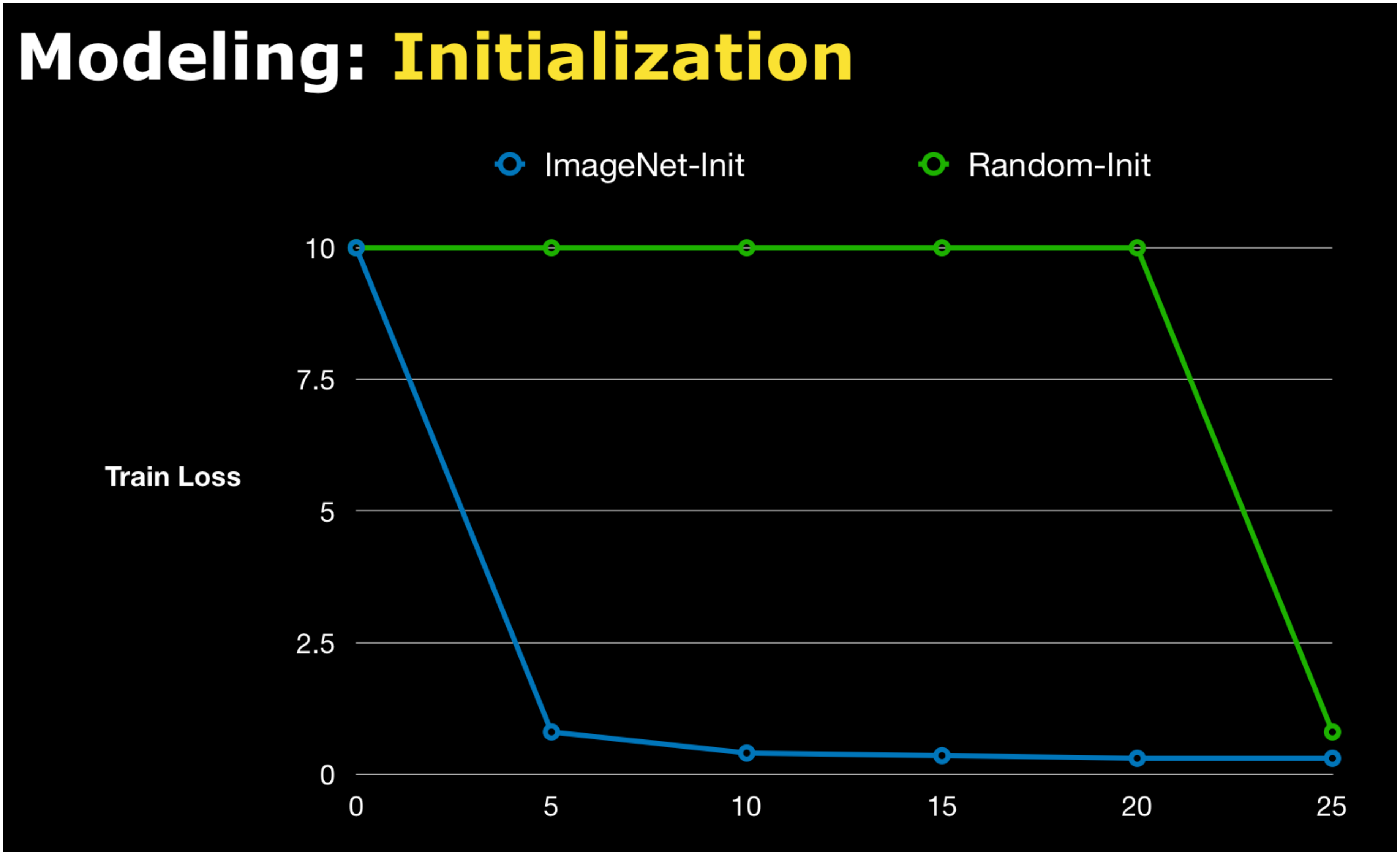
Transfer learning for feedforward networks

- Widely used technique in computer vision:
- Take a pre-trained model, chop off the top few layers, and train a new shallow model on the induced representation



Slide acknowledgement:TelecomBCN

Transfer learning for feedforward networks

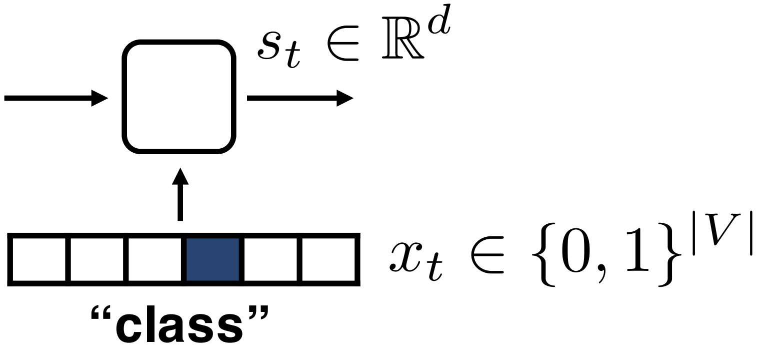


Modeling: Initialization

[Adam Yala, MIT 6.S897/HST.956 Lecture 13, 2019.]

Transfer learning for recurrent neural networks

- Naïve encoding of inputs for a RNN might use a one-hot encoding



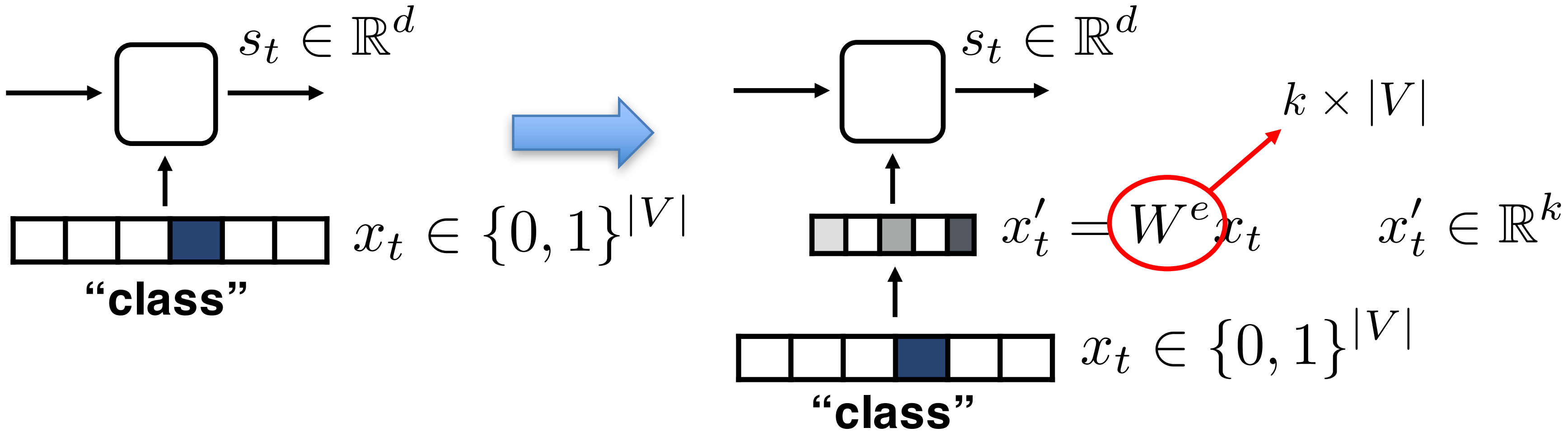
- An example of a (simplified) recurrent unit:

$$s_t = \tanh(W^{s,s} s_{t-1} + \underbrace{\boxed{W^{s,x}}}_{\substack{\text{dimension} \\ d \times |V|}} x_t)$$

- **Challenge:** how do we make hidden dimension *d* large, yet not overfit with rare words?

Transfer learning for recurrent neural networks

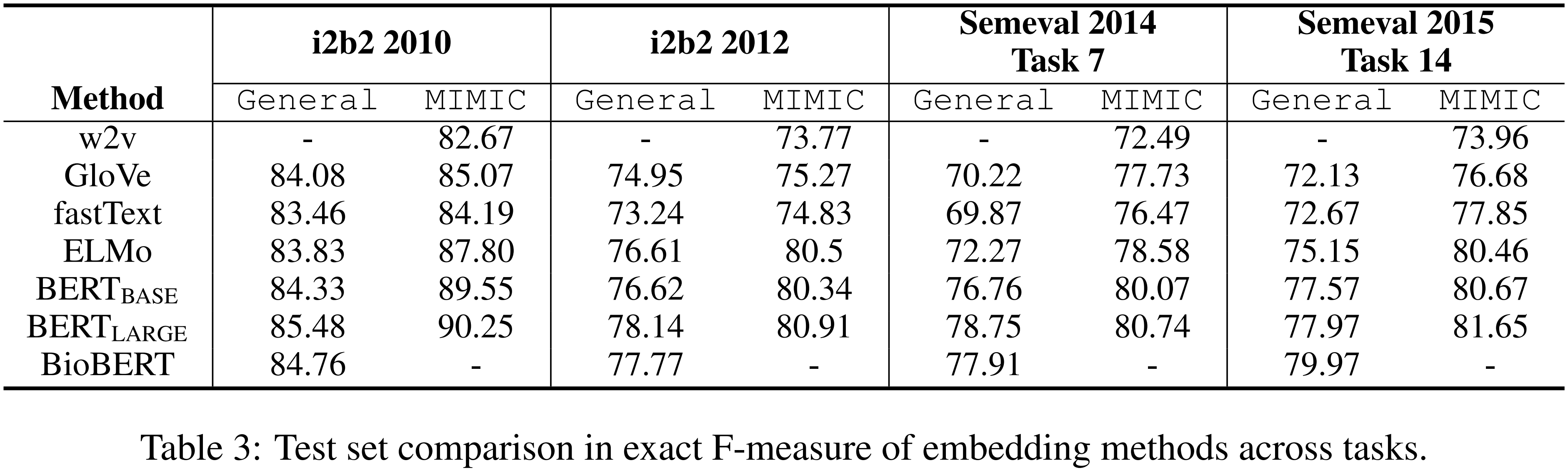
- Instead, do *linear transformation* of words prior to feeding to RNN



- Each column of *W*^e^ can be thought of as a *word embedding*, which can be trained end-to-end
- Can use *pre-trained* word embeddings, coming from learning a language model or another classification problem with a much larger dataset

Transfer learning for recurrent neural networks

Application: clinical concept extraction



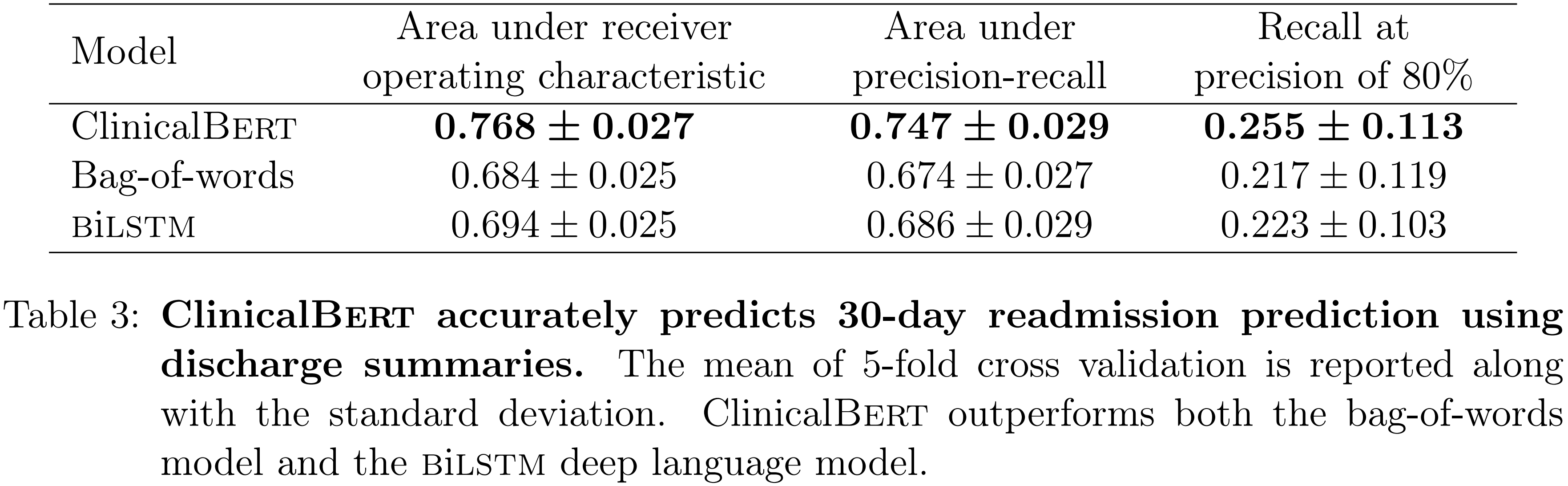
Method	i2b2 2010		i2b2 2012		Semeval 2014 Task 7		Semeval 2015 Task 14	
	`General`	`MIMIC`	`General`	`MIMIC`	`General`	`MIMIC`	`General`	`MIMIC`
w2v	-	82.67	-	73.77	-	72.49	-	73.96
GloVe	84.08	85.07	74.95	75.27	70.22	77.73	72.13	76.68
fastText	83.46	84.19	73.24	74.83	69.87	76.47	72.67	77.85
ELMo	83.83	87.80	76.61	80.5	72.27	78.58	75.15	80.46
$\text{BERT}_{\text{BASE}}$	84.33	89.55	76.62	80.34	76.76	80.07	77.57	80.67
$\text{BERT}_{\text{LARGE}}$	85.48	90.25	78.14	80.91	78.75	80.74	77.97	81.65
BioBERT	84.76	-	77.77	-	77.91	-	79.97	-

Table 3: Test set comparison in exact F-measure of embedding methods across tasks.

[Si, Wang, Xu, Roberts. Enhancing Clinical Concept Extraction with Contextual Embedding. arXiv:1902.08691, Feb 2019]

Transfer learning for recurrent neural networks

Application: classification from clinical notes



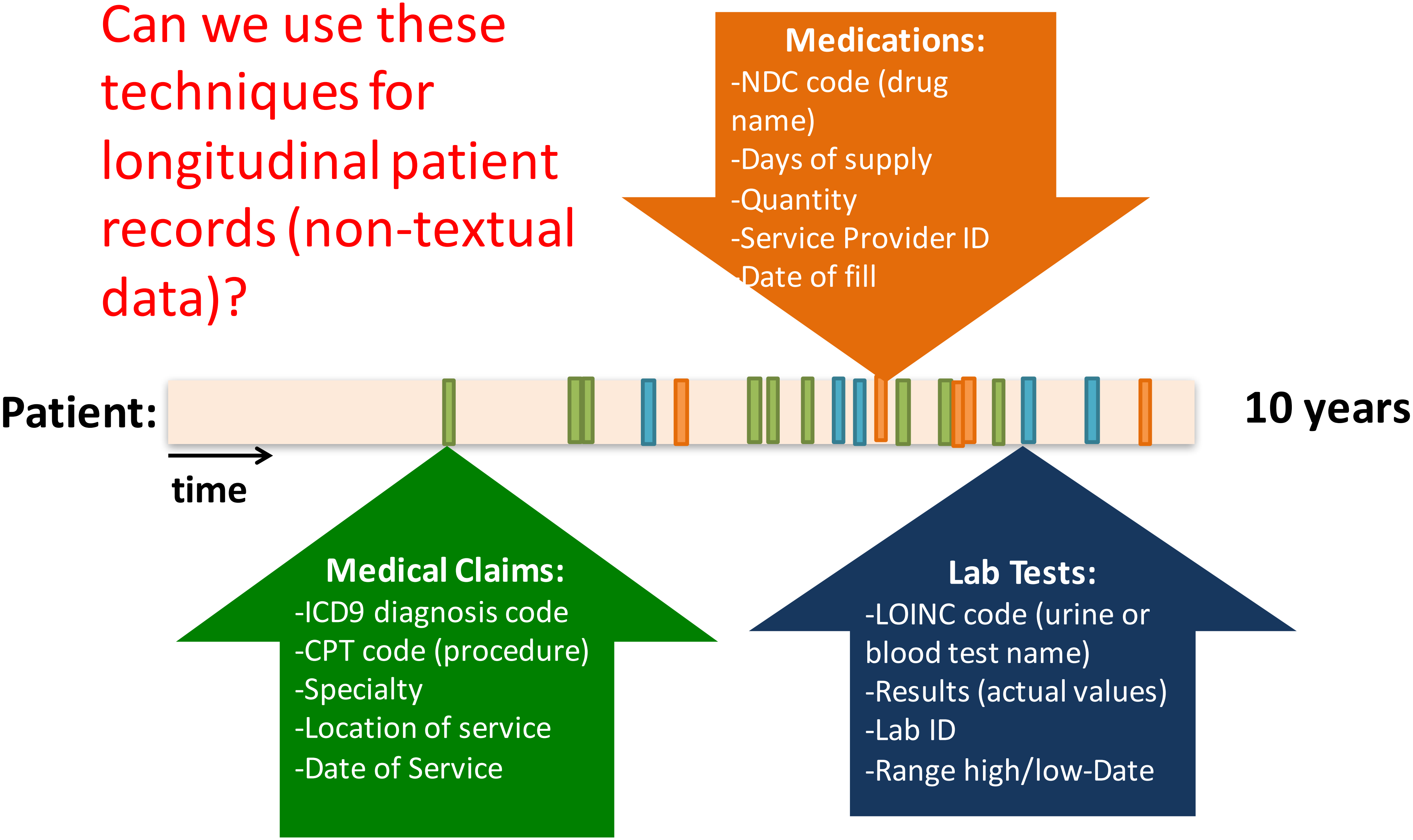
Model	Area under receiver operating characteristic	Area under precision-recall	Recall at precision of 80%
ClinicalBert	**0.768 ± 0.027**	**0.747 ± 0.029**	**0.255 ± 0.113**
Bag-of-words	0.684 ± 0.025	0.674 ± 0.027	0.217 ± 0.119
BiLSTM	0.694 ± 0.025	0.686 ± 0.029	0.223 ± 0.103

Table 3: **ClinicalBert accurately predicts 30-day readmission prediction using discharge summaries.** The mean of 5-fold cross validation is reported along with the standard deviation. ClinicalBert outperforms both the bag-of-words model and the BiLSTM deep language model.

[Huang, Altosaar, Ranganath. ClinicalBERT: Modeling Clinical Notes and Predicting Hospital Readmission. arXiv:1904.05342, Apr 2019]

Transfer learning for recurrent neural networks

Can we use these techniques for longitudinal patient records (non-textual data)?



Medications:

-NDC code (drug name)

-Days of supply

-Quantity

-Service Provider ID

-Date of fill

Patient:

time

10 years

Medical Claims:

-ICD9 diagnosis code

-CPT code (procedure)

-Specialty

-Location of service

-Date of Service

Lab Tests:

-LOINC code (urine or blood test name)

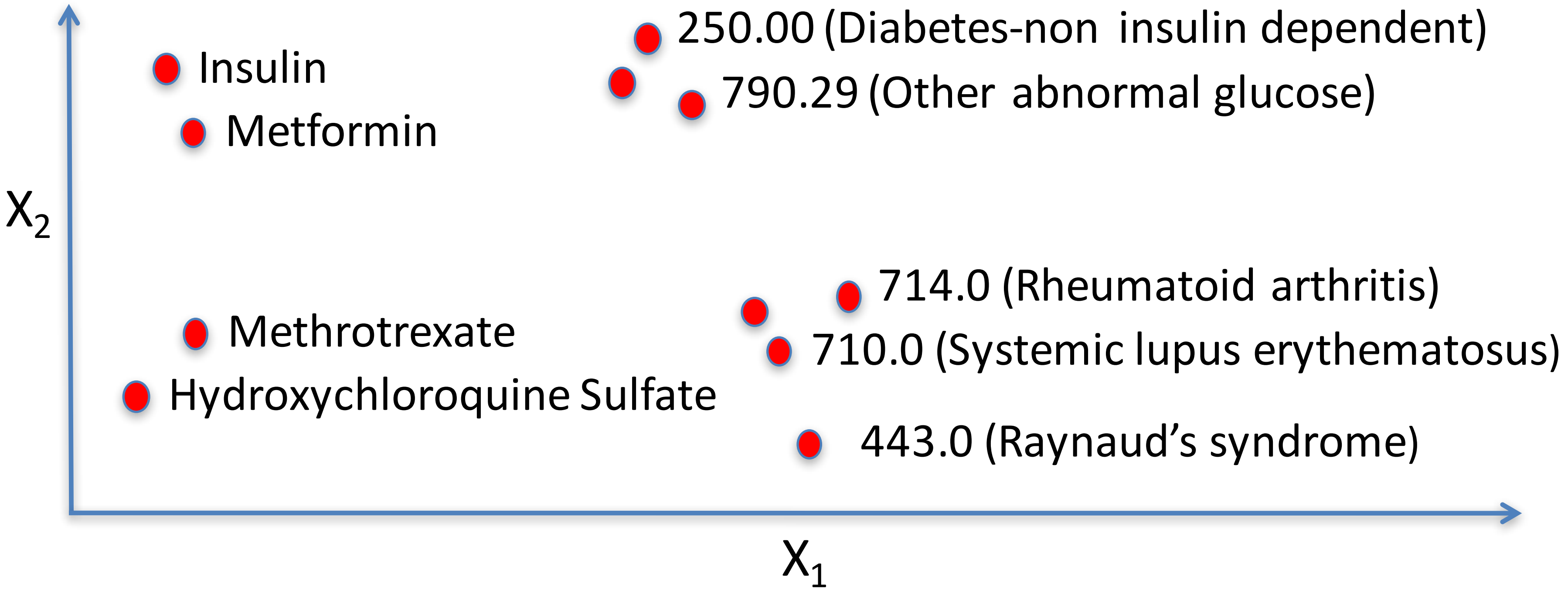
-Results (actual values)

-Lab ID

-Range high/low-Date

Transfer learning for recurrent neural networks

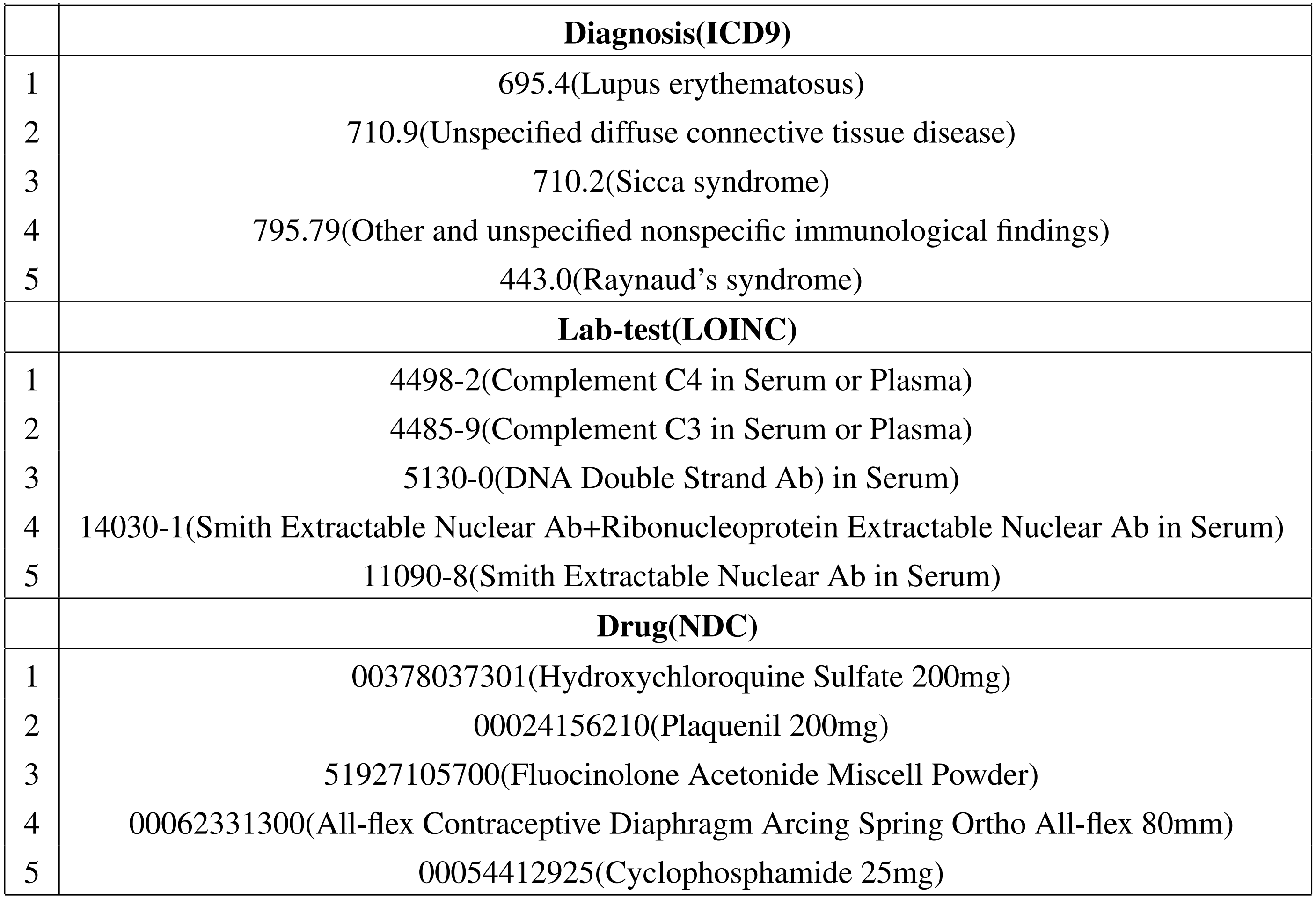
- Can we embed all 3 million+ concepts in the UMLS (Unified Medical Language System), 140,000 ICD-10-CM diagnosis and procedure codes, 360,000 NDC medication codes…?


250.00 (Diabetes-non insulin dependent)
Insulin
790.29 (Other abnormal glucose)
Metformin
X_2
714.0 (Rheumatoid arthritis)
Methrotrexate
710.0 (Systemic lupus erythematosus)
Hydroxychloroquine Sulfate
443.0 (Raynaud's syndrome)
X_1

[Choi, Chiu, Sontag, Learning Low-Dimensional Representations of Medical Concepts, AMIA CRI 2016;
Choi, Bahadori et al., Multi-Layer Representation Learning for Medical Concepts, KDD 2016;
Beam et al., Clinical Concept Embeddings Learned from Massive Sources..., arXiv:1804.01486, 2018]

Transfer learning for recurrent neural networks

- Nearest neighbors of 710.0 (Systemic lupus erythematosus):



	Diagnosis(ICD9)
1	695.4(Lupus erythematosus)
2	710.9(Unspecified diffuse connective tissue disease)
3	710.2(Sicca syndrome)
4	795.79(Other and unspecified nonspecific immunological findings)
5	443.0(Raynaud's syndrome)
	Lab-test(LOINC)
1	4498-2(Complement C4 in Serum or Plasma)
2	4485-9(Complement C3 in Serum or Plasma)
3	5130-0(DNA Double Strand Ab) in Serum)
4	14030-1(Smith Extractable Nuclear Ab+Ribonucleoprotein Extractable Nuclear Ab in Serum)
5	11090-8(Smith Extractable Nuclear Ab in Serum)
	Drug(NDC)
1	00378037301(Hydroxychloroquine Sulfate 200mg)
2	00024156210(Plaquenil 200mg)
3	51927105700(Fluocinolone Acetonide Miscell Powder)
4	00062331300(All-flex Contraceptive Diaphragm Arcing Spring Ortho All-flex 80mm)
5	00054412925(Cyclophosphamide 25mg)

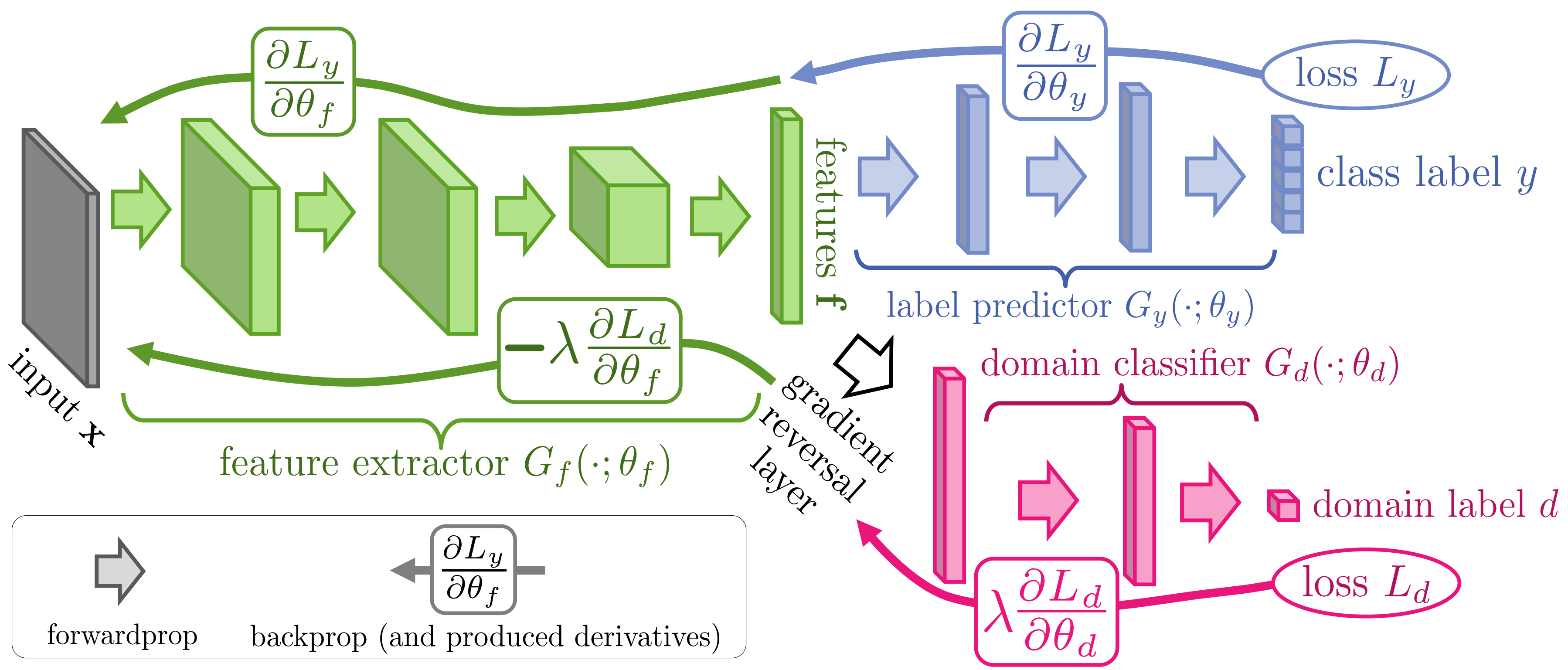
[Choi, Chiu, Sontag, Learning Low-Dimensional Representations of Medical Concepts, AMIA CRI 2016]

Transfer learning

- We have a lot of data from p(x,y) **and** a little data from q(x,y)
- How can we quickly adapt?
  1. Linear models: original representation, modify weights
  2. Linear models: manually choose a good shared representation
  3. Deep models: re-use part of the learned representation, fine-tune
  4. Deep models: automatically find a good shared representation

Automatically find a good shared representation

- Guided by learning theory (Ben-David et al. ‘06), recent work shows how to do domain adaptation *without labels in target set*:



$\frac{\partial L_y}{\partial \theta_f}$
$\frac{\partial L_y}{\partial \theta_y}$
loss L_y
class label y
features $\mathbf{f}$
label predictor $G_y(\cdot;\theta_y)$
input $\mathbf{x}$
$-\lambda\frac{\partial L_d}{\partial \theta_f}$
feature extractor $G_f(\cdot;\theta_f)$
gradient reversal layer
domain classifier $G_d(\cdot;\theta_d)$
domain label d
loss L_d
$\lambda\frac{\partial L_d}{\partial \theta_d}$
forwardprop
backprop (and produced derivatives)

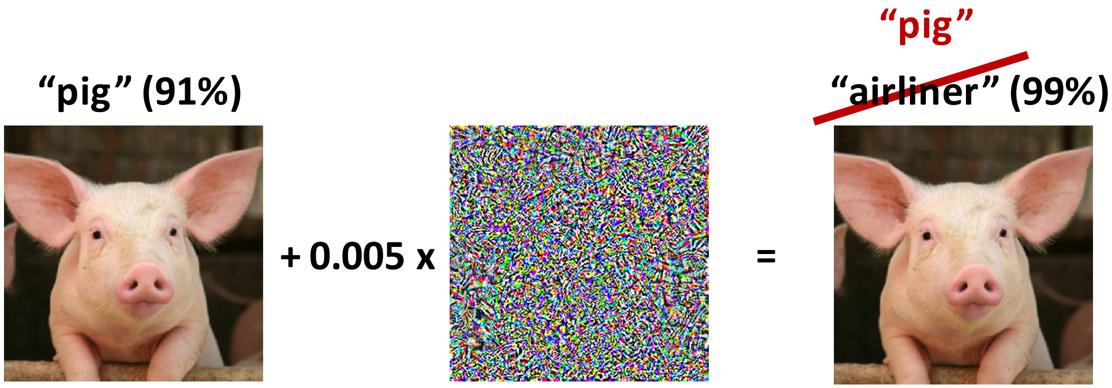
Ganin et al., Domain-Adversarial Training of Neural Networks. JMLR ‘16

Outline for lecture

1. Building population-level checks into deployment/transfer

2. Machine learning in anticipation of dataset shift
   - *Transfer learning*
   - ***Defenses against adversarial attacks***

Towards Adversarially Robust Models



Acknowledgement: Slides from Aleksander Madry, MIT

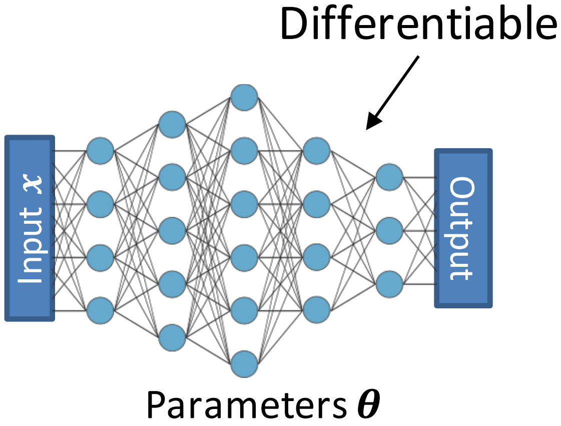
Where Do Adversarial Examples Come From?

To get an adv. example

~~Goal of~~ training:

Model Parameters Input Correct Label

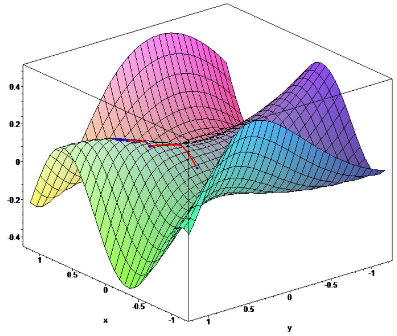
$$\min_{\theta}\ loss(\theta, x, y)$$



Differentiable

Parameters $\boldsymbol{\theta}$

Can use gradient descent method to find good θ



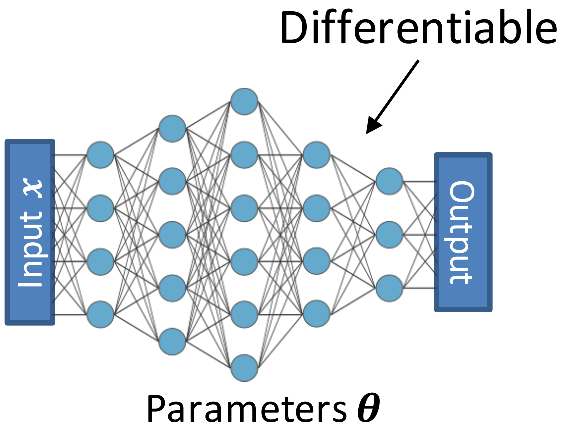
Slide credit: Aleksander Madry

Where Do Adversarial Examples Come From?

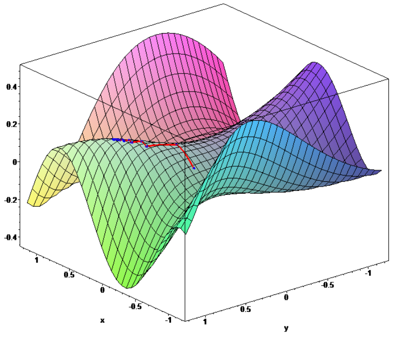
To get an adv. example

~~Goal of training:~~

$$loss(\theta, x + \delta, y)$$



Can use gradient descent method to find good θ



Slide credit: Aleksander Madry

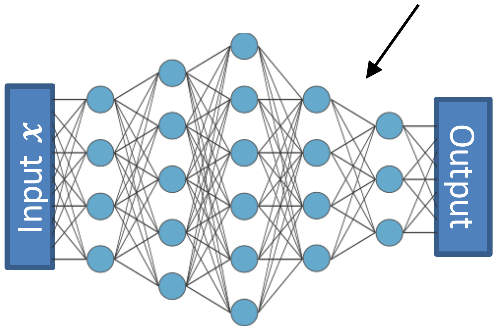
Where Do Adversarial Examples Come From?

To get an adv. example

~~Goal of~~

training:

$$max_{\delta}\ loss(\theta, x + \delta, y)$$



Parameters $\boldsymbol{\theta}$

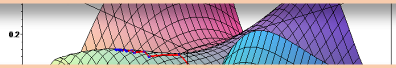
Which δ are allowed?

Examples: δ that is small wrt

- ℓ_p-norm
- Rotation and/or translation
- VGG feature perturbation
- (add the perturbation you need here)

Can use gradient descent

This choice is important (but we put it aside)



In any case: We have to confront (small) ℓ_p-norm perturbations

Slide credit: Aleksander Madry

Towards ML Models that Are Adv. Robust

[M Makelov Schmidt Tsipras Vladu 2018]

Key observation: Lack of adv. robustness is **NOT** at odds with what we currently want our ML models to achieve

~~Standard~~ generalization: $\mathbb{E}_{(x,y)\sim D}\left[loss(\theta, x, y)\right]$

Adversarially robust

But: Adversarial noise is a "needle in a haystack"

Slide credit: Aleksander Madry

Towards ML Models that Are Adv. Robust

[M Makelov Schmidt Tsipras Vladu 2018]

Key observation: Lack of adv. robustness is **NOT** at odds with what we currently want our ML models to achieve

~~Standard~~ generalization: $\mathbb{E}_{(x,y)\sim D}\left[\max_{\boldsymbol{\delta}\in\boldsymbol{\Delta}} loss(\theta, x+\boldsymbol{\delta}, y)\right]$

Adversarially robust

But: Adversarial noise is a "needle in a haystack"

Slide credit: Aleksander Madry

Towards ML Models that Are Adv. Robust

[M Makelov Schmidt Tsipras Vladu 2018]

Resulting training primitive:

$$\min_{\theta} \max_{\delta \in \Delta} \; loss(\theta, x + \boldsymbol{\delta}, y)$$

Finding a robust model   Finding a "bad" perturbation

To improve the model: Train on **perturbed** inputs
(aka as "adversarial training" **[Goodfellow Shlens Szegedy '15]**)

Does this work?

Yes! (In practice)
But certain care is required

Slide credit: Aleksander Madry

ConvNet for MNIST that provably has less than 5.8% test error for any adversarial attack with bounded l_inf norm less than 0.1

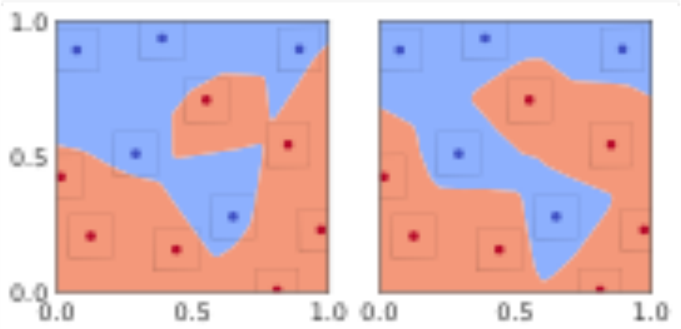


Figure 3. Illustration of classification boundaries resulting from standard training (left) and robust training (right) with ℓ_∞ balls of size $\epsilon = 0.08$ (shown in figure).

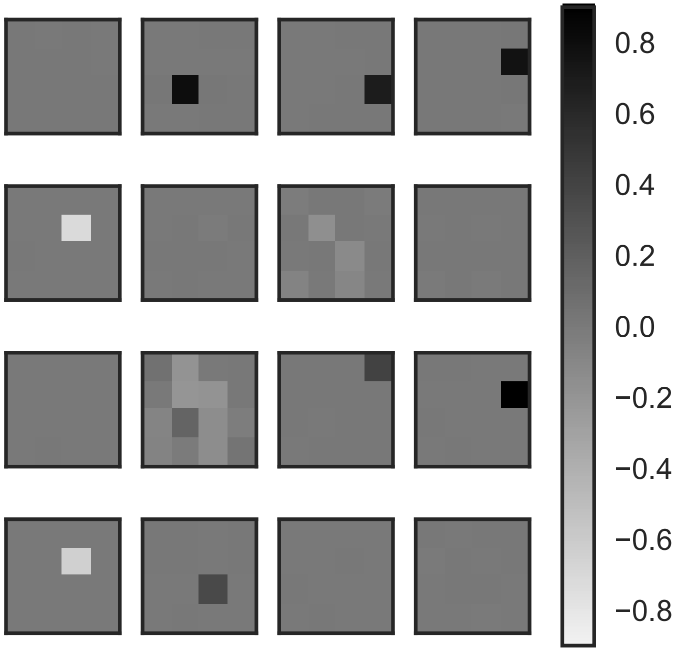
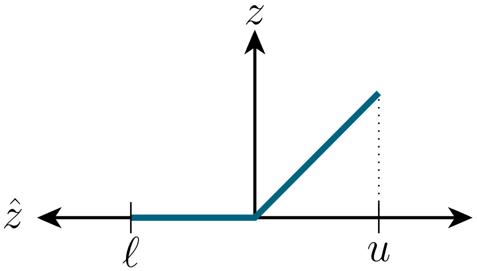
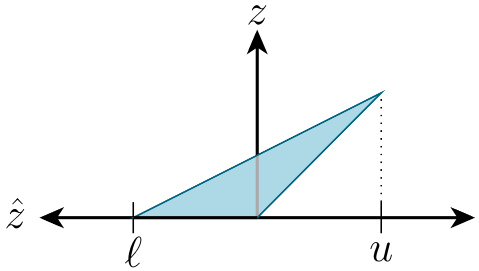


Figure 8. Learned convolutional filters for MNIST of the first layer of a trained robust convolutional network, which are quite sparse due to the ℓ_1 term in (6).



Bounded ReLU set

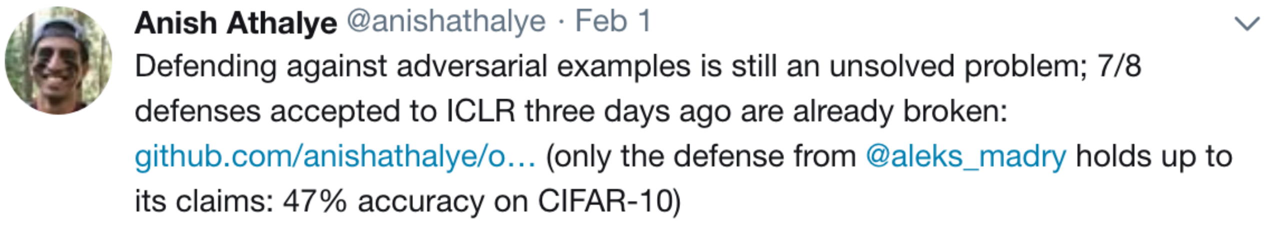


Convex relaxation

[Wong & Kolter, Provable Defenses against Adversarial Examples via the Convex Outer Adversarial Polytope, ICML 2018.]

How do we know this really works?

→ Seems to be a recurring problem…



Anish Athalye @anishathalye · Feb 1

Defending against adversarial examples is still an unsolved problem; 7/8 defenses accepted to ICLR three days ago are already broken: github.com/anishathalye/o… (only the defense from @aleks_madry holds up to its claims: 47% accuracy on CIFAR-10)

Robustness by obscurity/complexity just does NOT work

→ Apply the standard security methodology:

- Evaluate with multiple **adaptive** attacks
- Use public security challenges

RobustML

(see robust-ml.org)

→ Use formal verification (where feasible):

- There is a steady progress on scaling these techniques up

[Katz et al ‘17, Wong Kolter ’18, Tjeng et al ’18, Dvijotham et al ‘18, Xiao Tjeng Shafiullah **M** ‘18]

Slide credit: Aleksander Madry