Machine Learning for Healthcare
HST.956, 6.S897

Lecture 24: Robustness to dataset shift

David Sontag

Course announcements

- Projects
  - Poster session
  - Send posters to print
  - Final report due
- 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

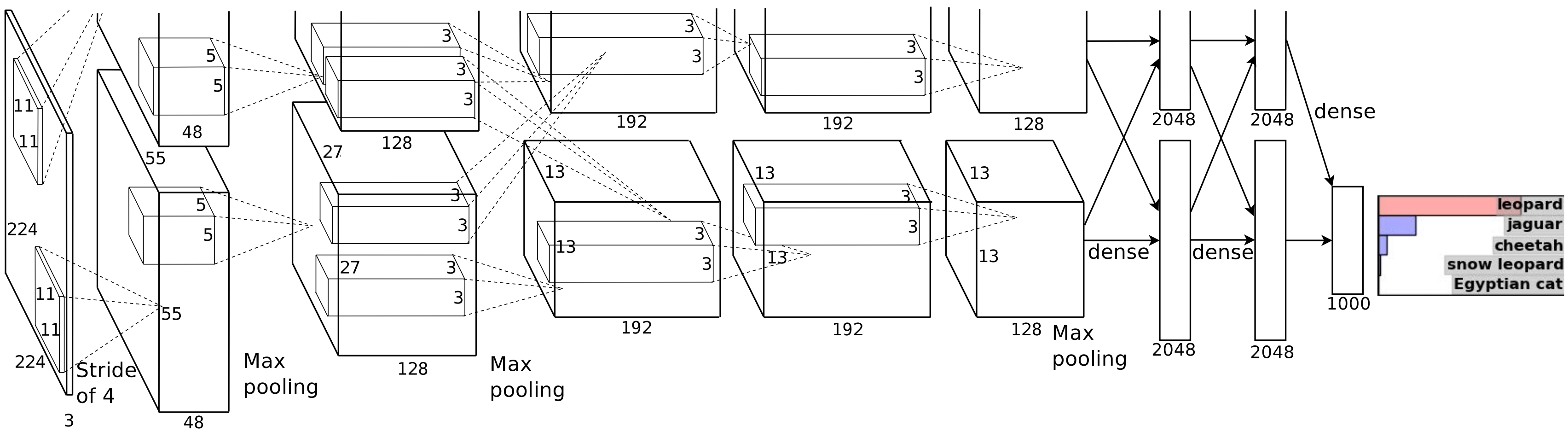
- So, you train your ML model and do a prospective evaluation at your institution → all looks good!
- What could go wrong at time of deployment?
  - Adversarial perturbations of inputs
  - Natural changes in the data (e.g. from transferring to a new place, or non-stationarity)

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

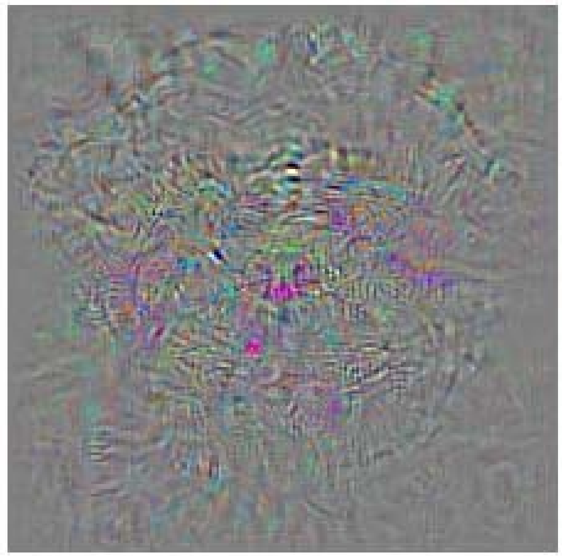
Correctly classified as a Dog

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

Machine learning is brittle: adversarial perturbations

Original image

+

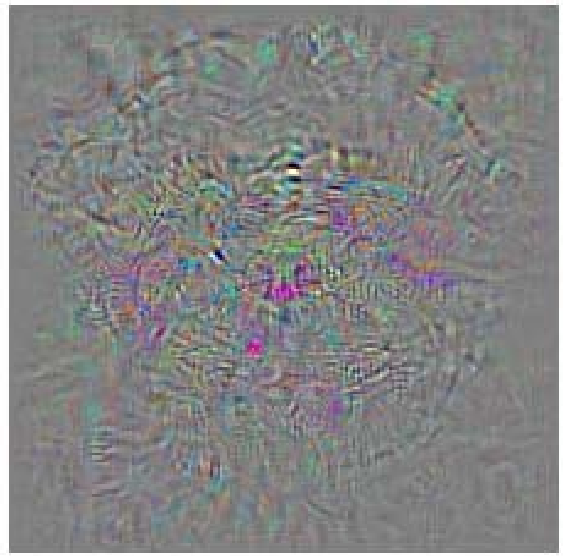


Noise (not random)

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

Machine learning is brittle: adversarial perturbations

+



=

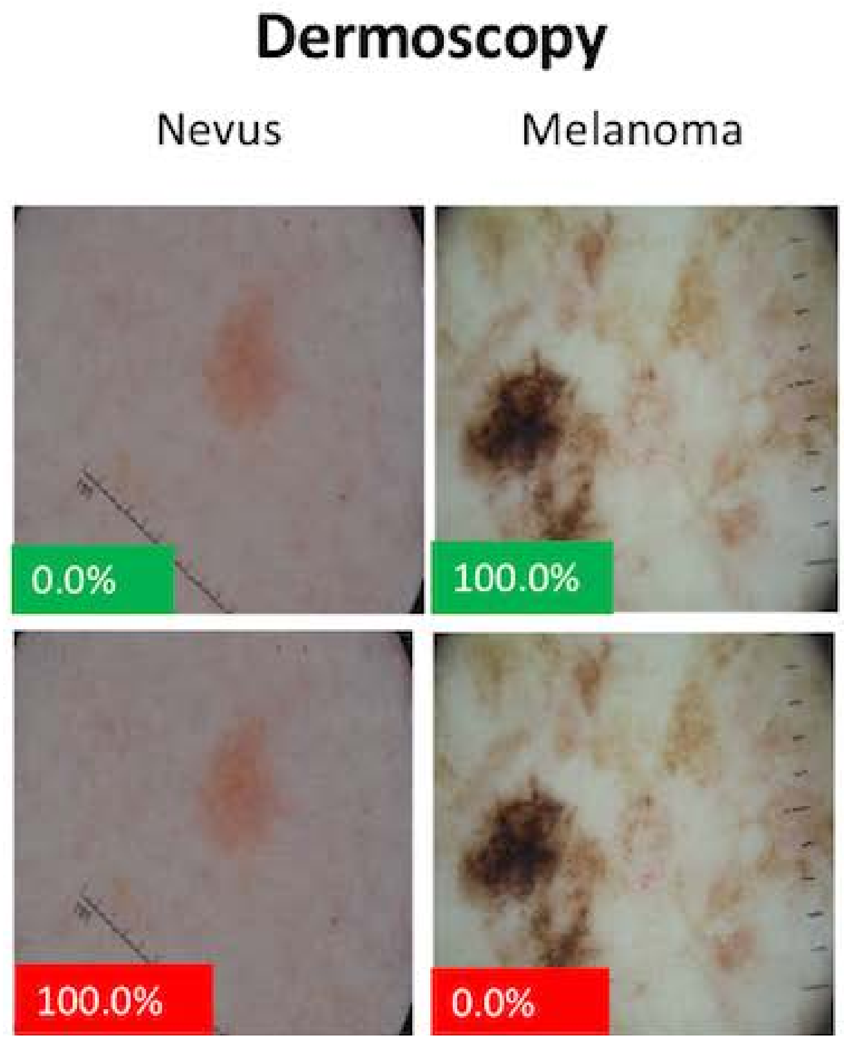
Original image

Noise (not random)

Classified as Ostrich!

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

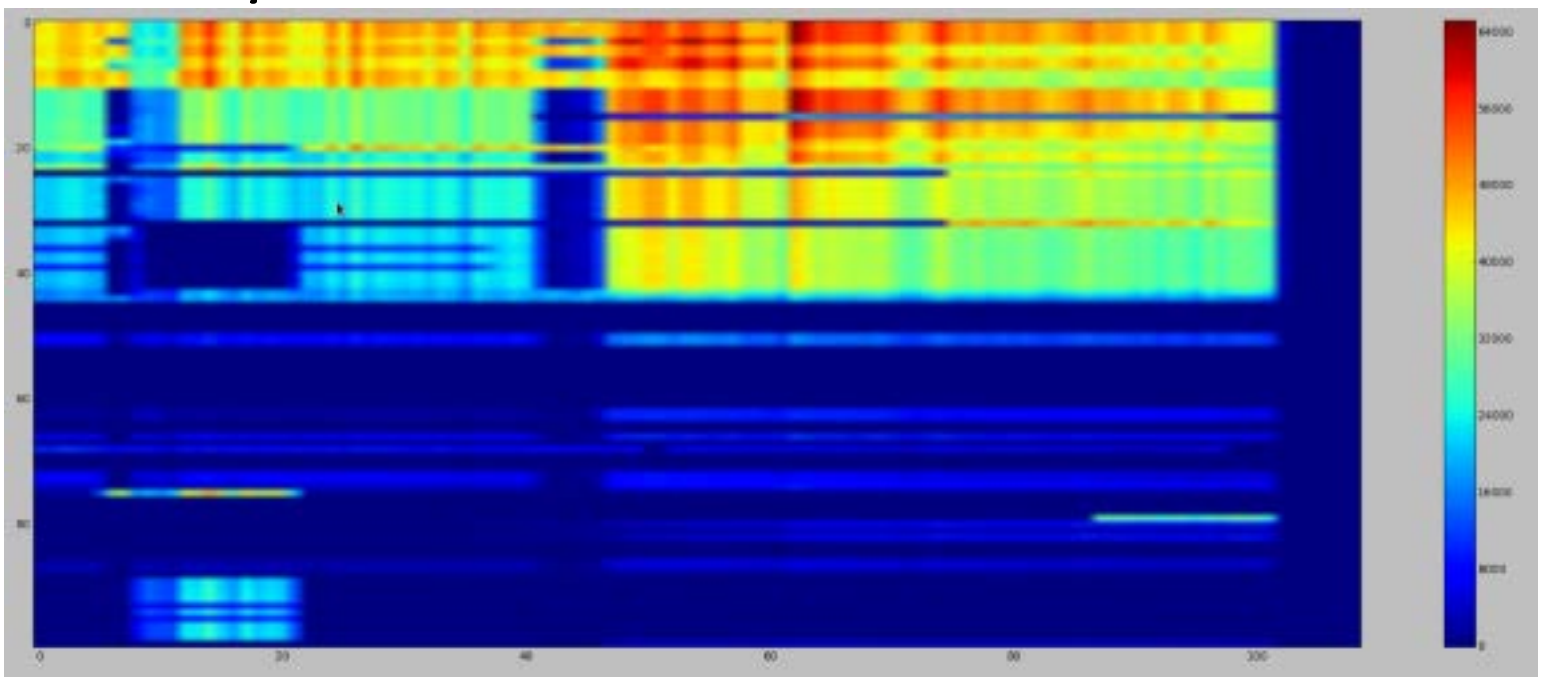
Machine learning is brittle: adversarial perturbations



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

Machine learning is brittle: natural changes in the data

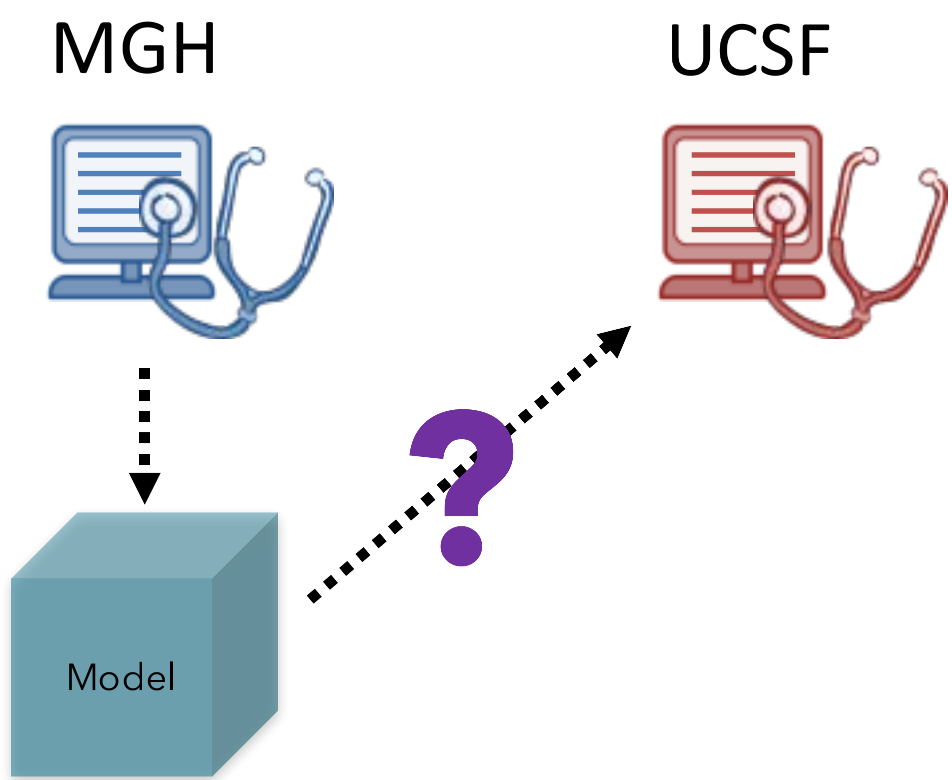
Top 100 lab measurements over time



→ 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*

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:

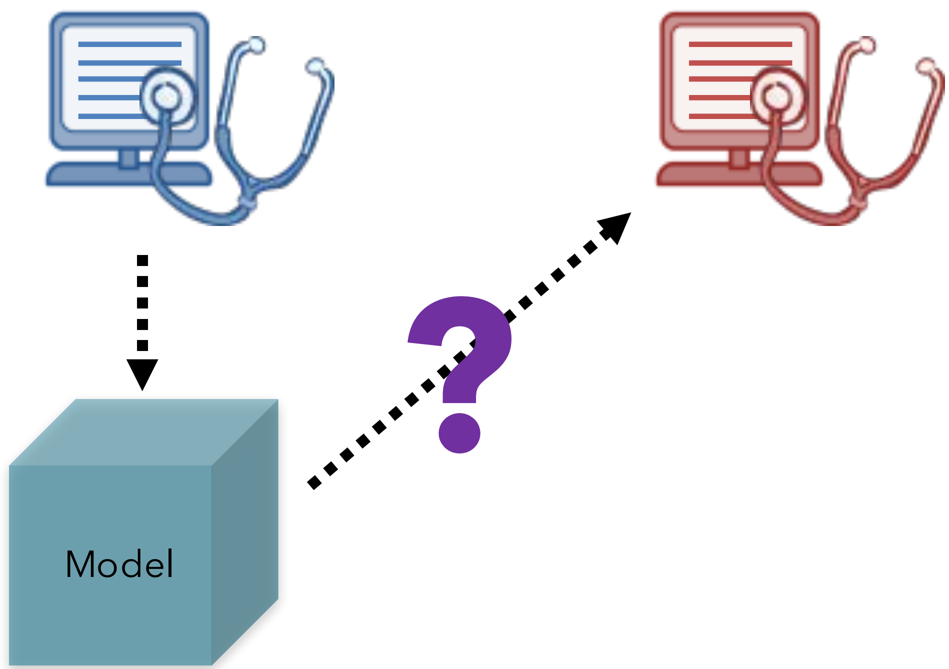
$$||w - w_{\text{old}}||_2^2 \quad \text{or} \quad ||w - w_{\text{old}}||_1$$

- 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

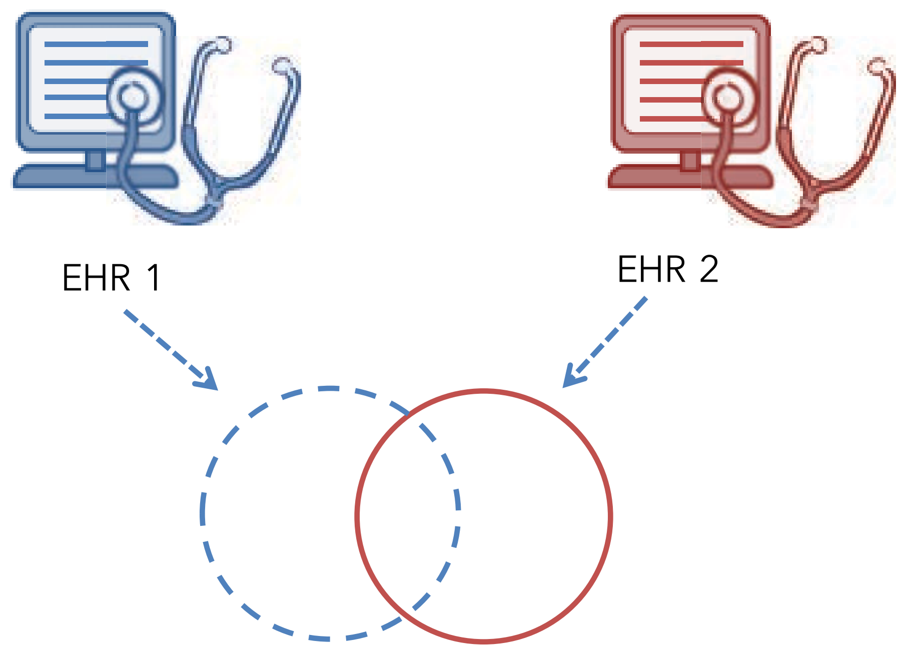
Predicting Clinical Outcomes Across Changing Electronic Health Record Systems



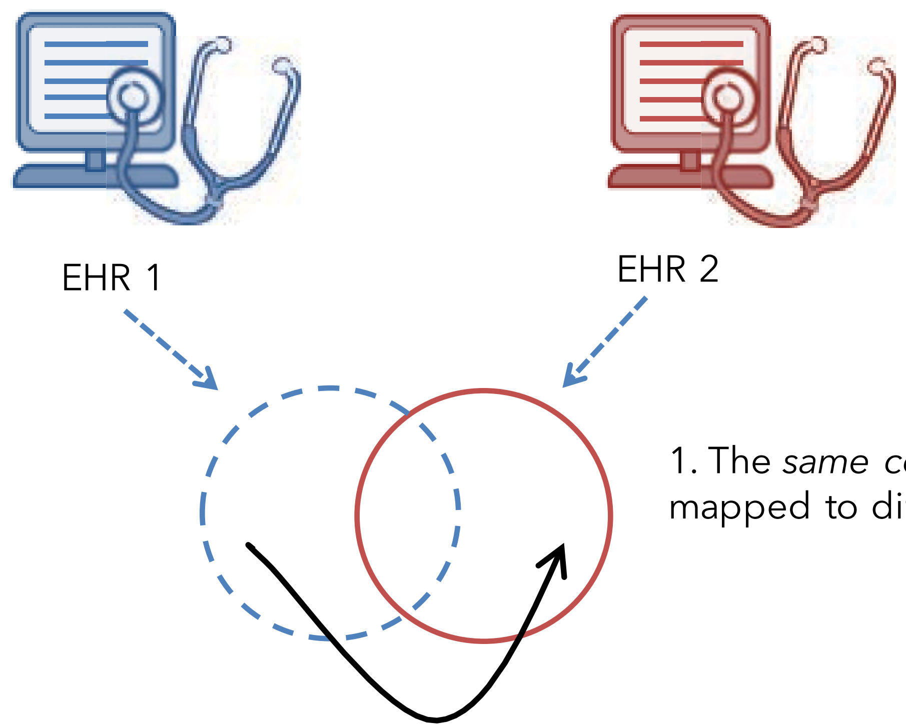
Jen J. Gong, Tristan Naumann, Peter Szolovits, John V. Guttag
Computer Science and Artificial Intelligence Laboratory, MIT

KDD 2017

Applying analytics across changing EHR systems is challenging

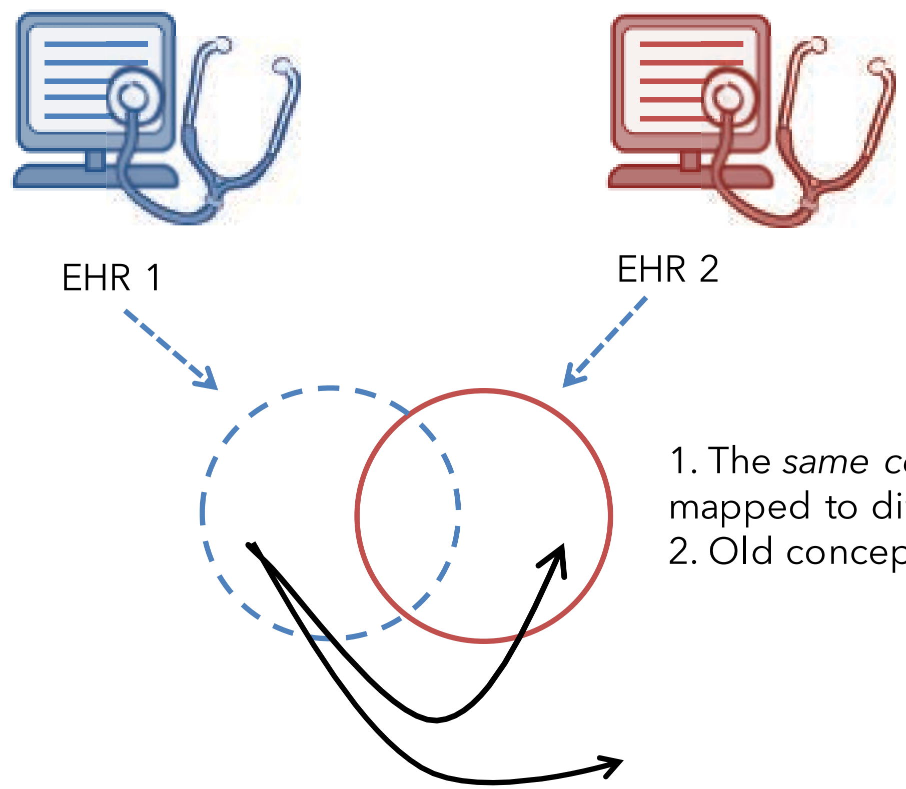


Applying analytics across changing EHR systems is challenging



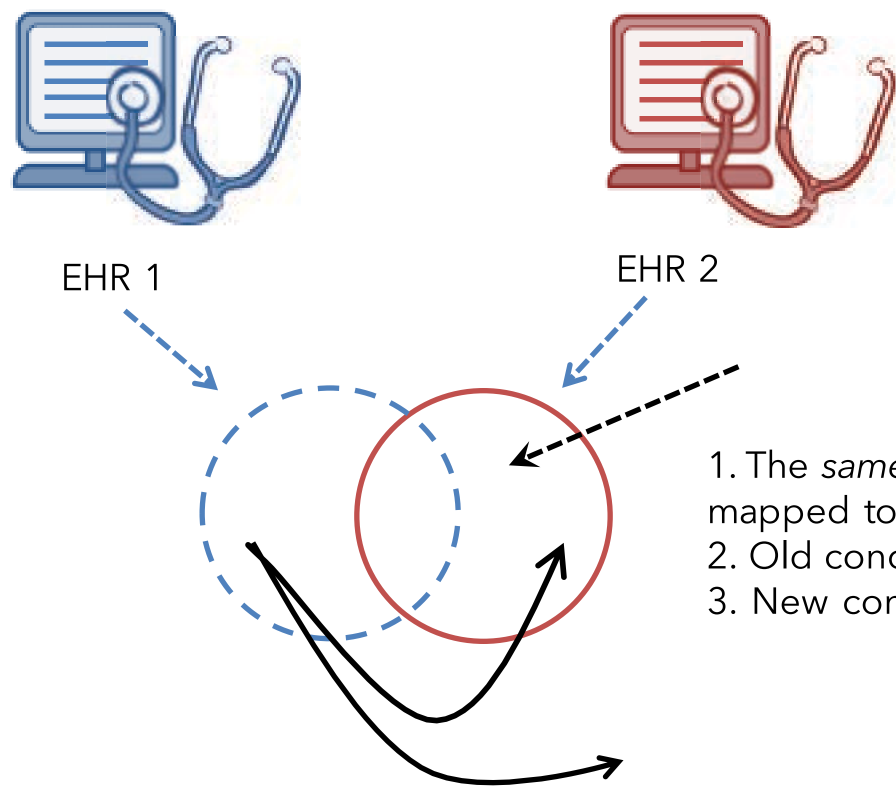
1. The *same conceptual items* might be mapped to different *encodings*.

Applying analytics across changing EHR systems is challenging



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

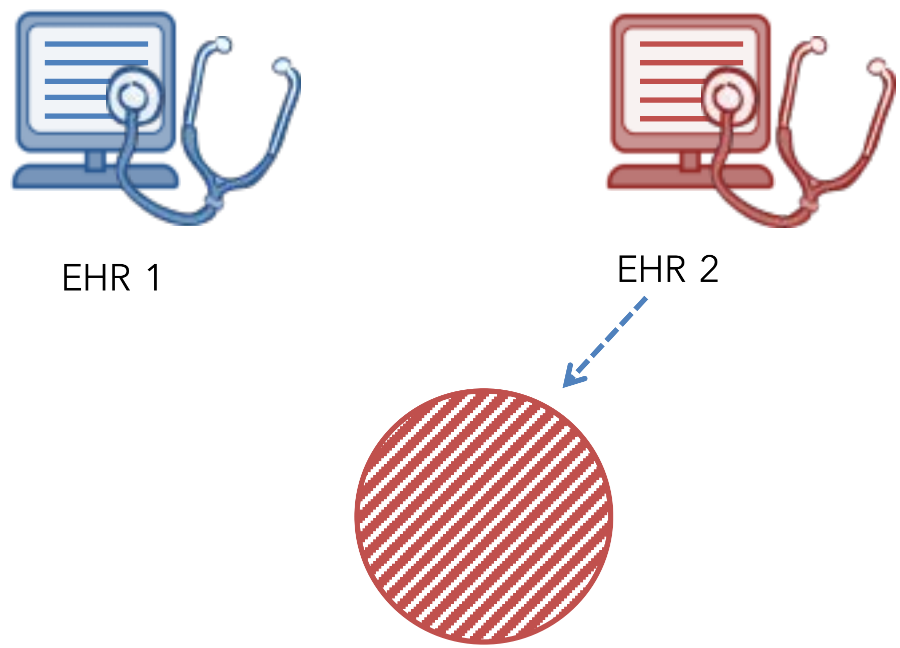
Applying analytics across changing EHR systems is challenging



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

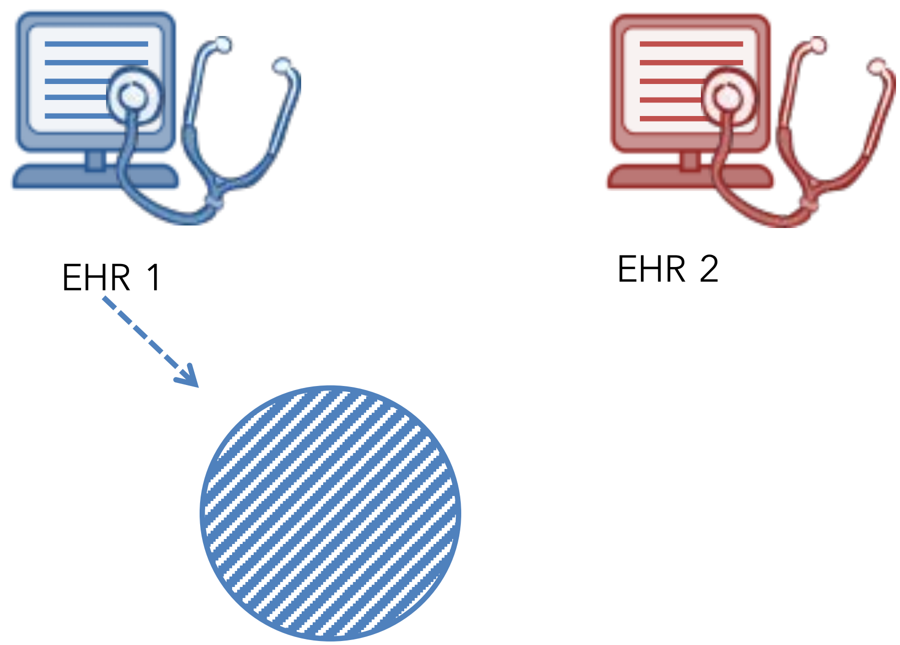
...

We can learn models using only EHR 2



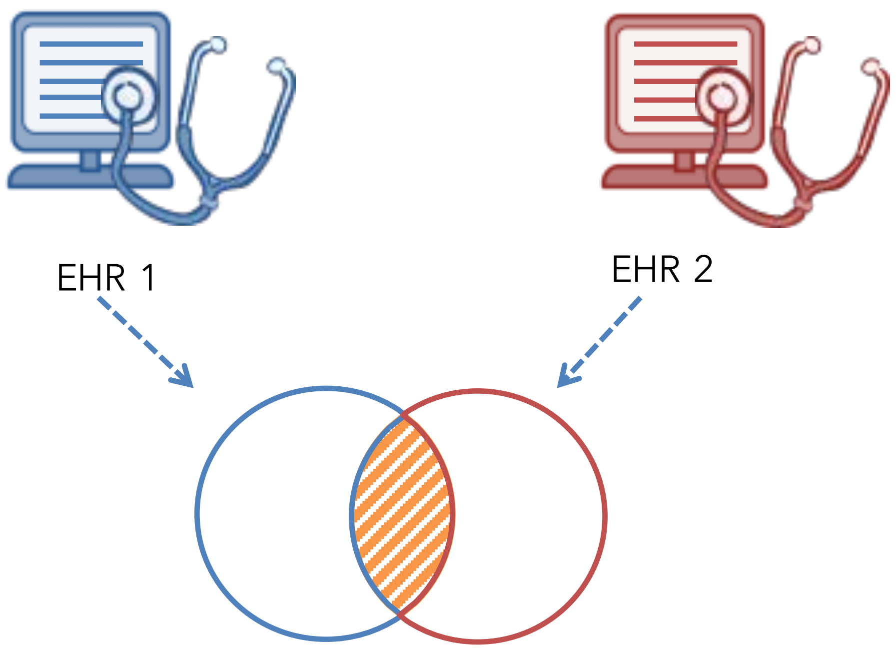
But this results in throwing away valuable data.

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



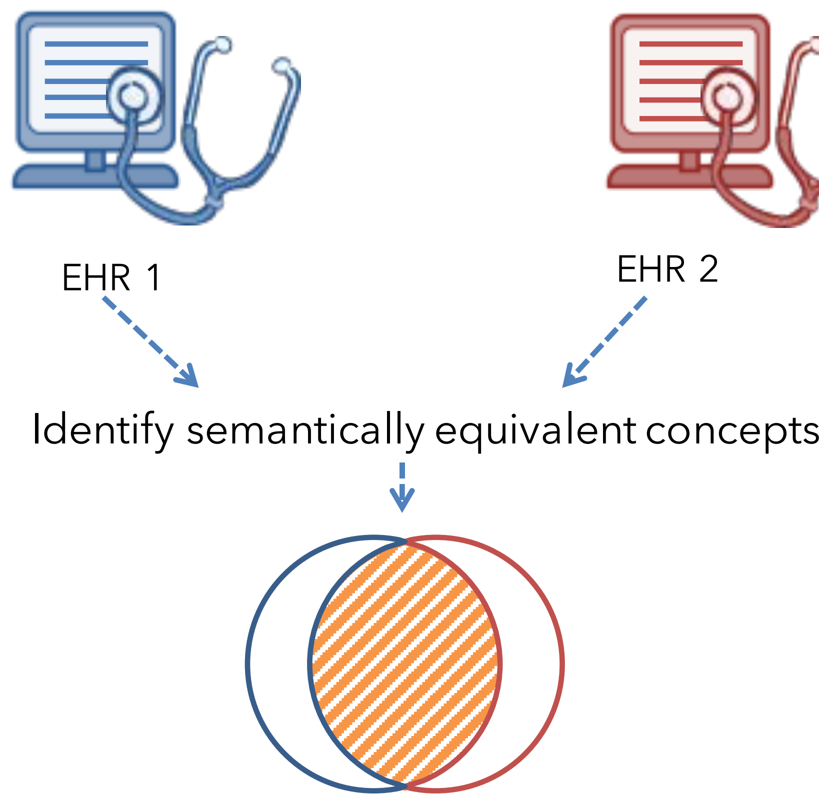
But concepts important in EHR 1 may not appear in EHR 2, and vice versa.

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

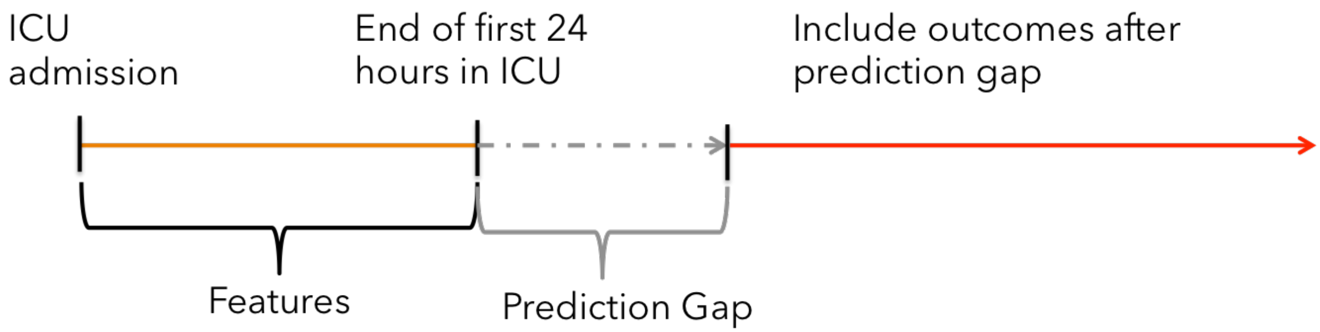


But this could remove the majority of clinical concepts in both EHRs from our model.

Solution: Map semantically similar items to a shared vocabulary

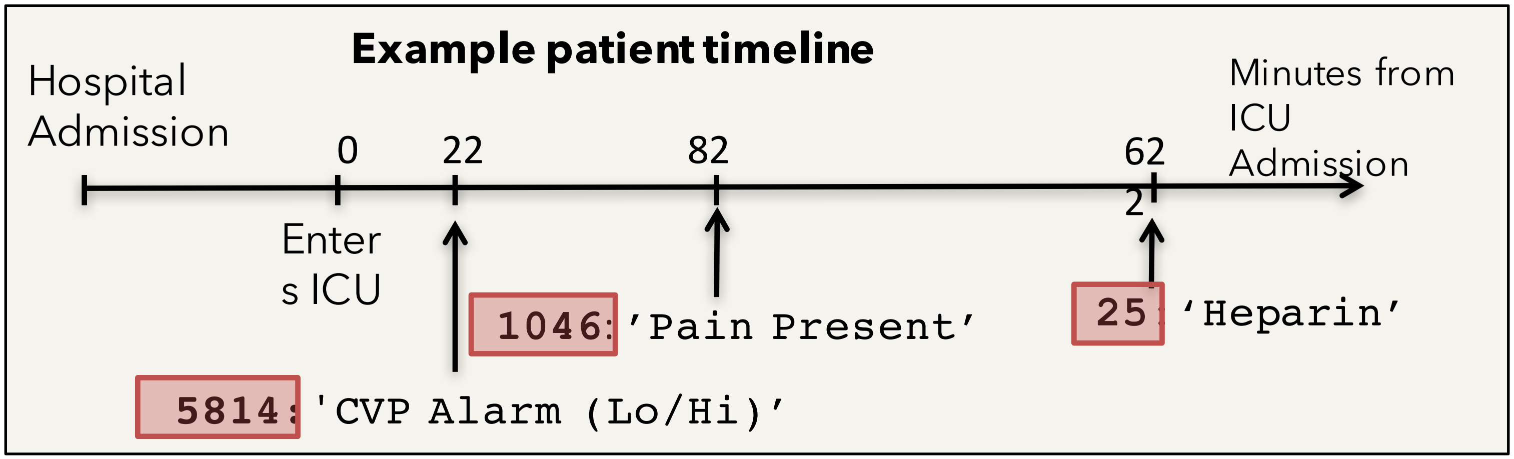


Predictive Models



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

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

From EHR-specific events to a shared vocabulary

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

Unified Medical Language System

ischemic stroke	hemorrhagic stroke	...
1	2	...

C0948008 ischemic stroke	C0553692 hemorrhagic stroke	C0475224 ischemic	C0333275 hemorrhagic	C0038454 stroke	
1	2	1	2	3	...

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

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

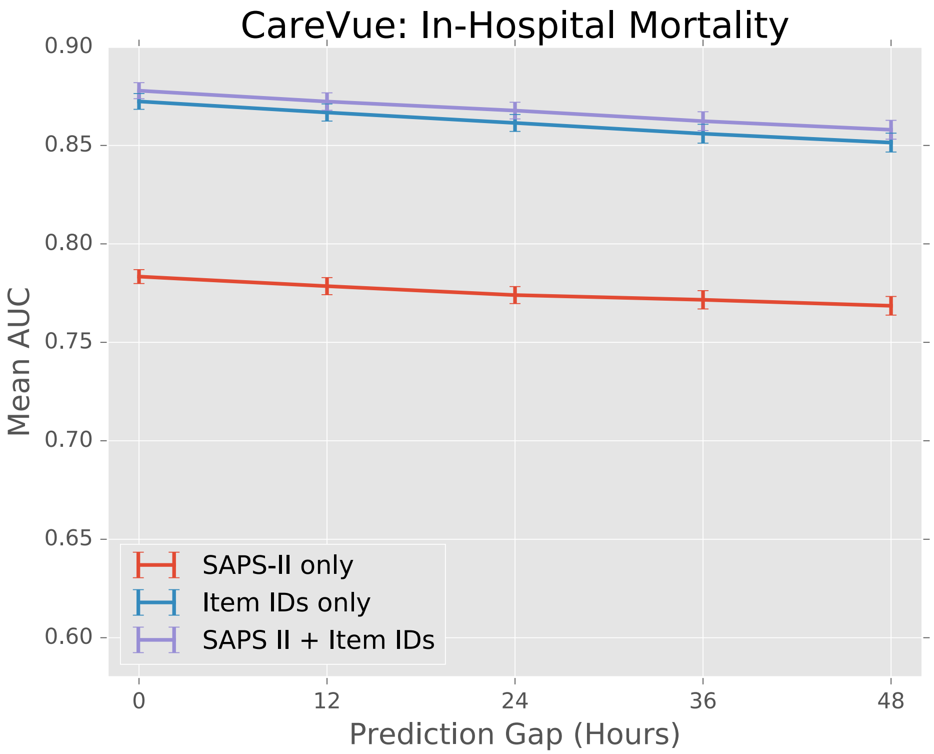
Three Experiments

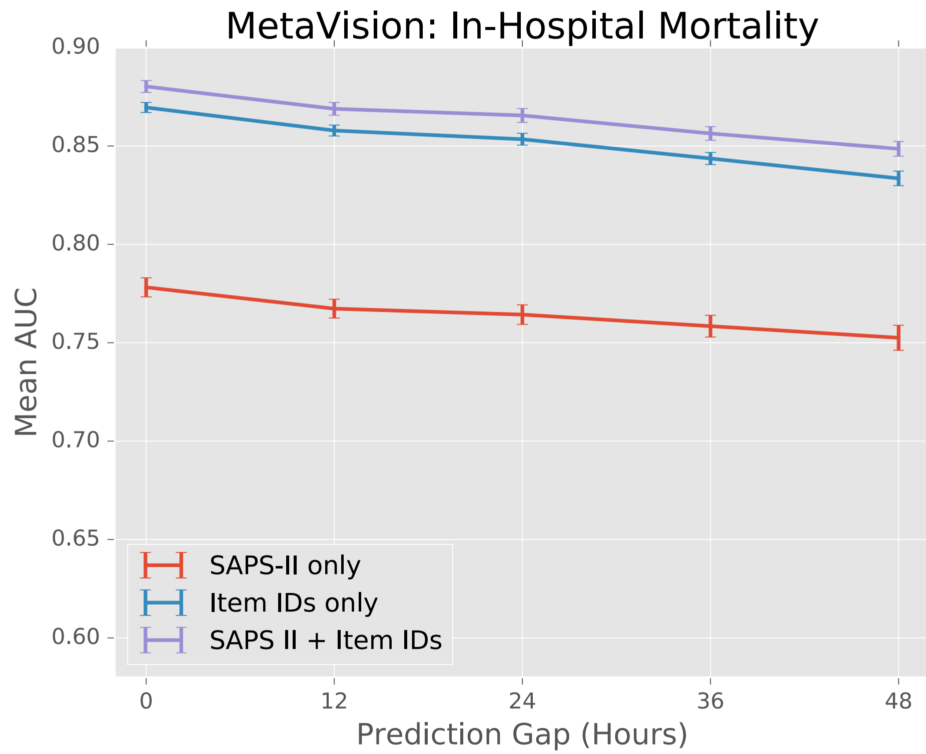
1. Show that a *Bag-of-Events* feature representation is useful in predicting clinical outcomes within each EHR version.

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**.*

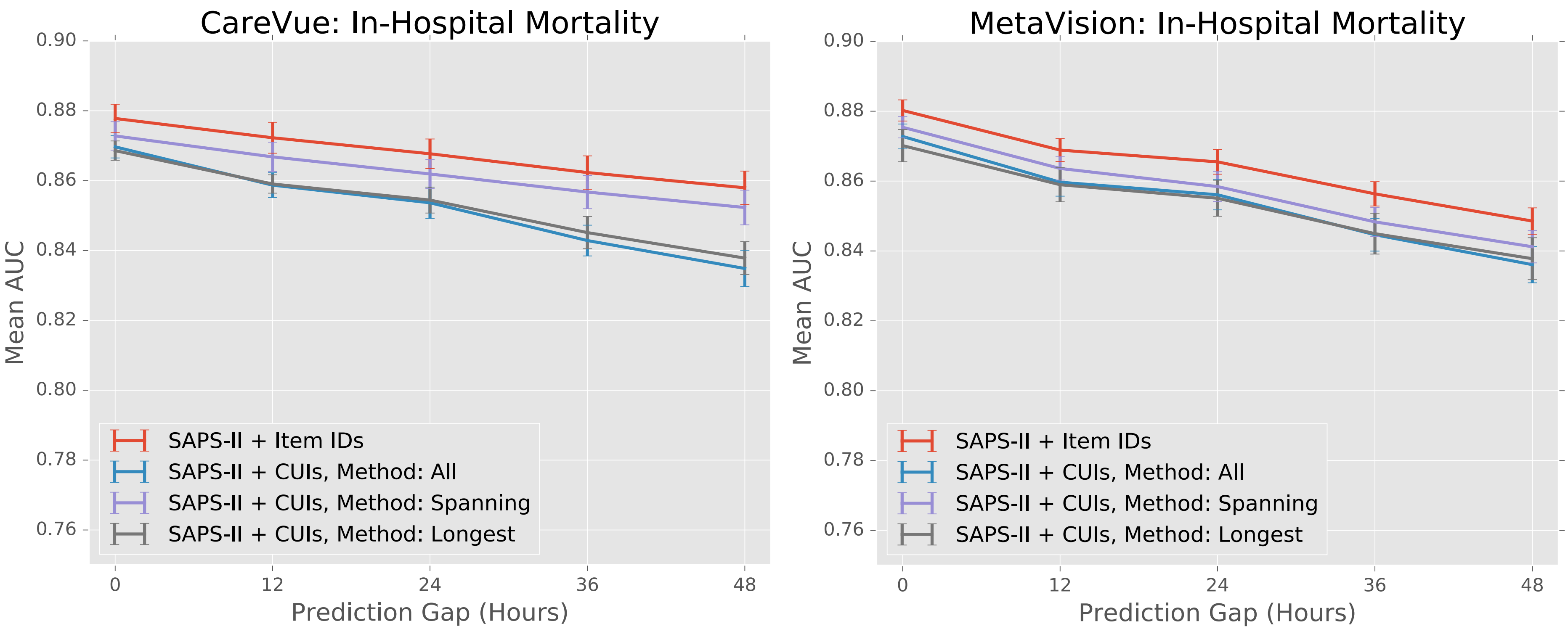
Does BOE feature representation have predictive value?



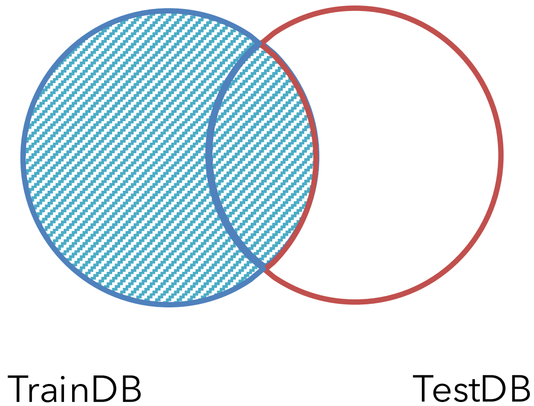


Simplified Acute Physiology Score (SAPS-II): Uses statistics about patient physiology (e.g., heart rate, blood pressure, urine output).

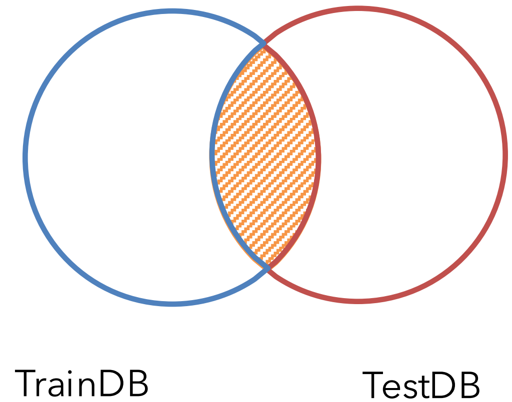
What is the impact of mapping BOEs to CUIs within single EHRs?



What happens when we apply models across EHRs?

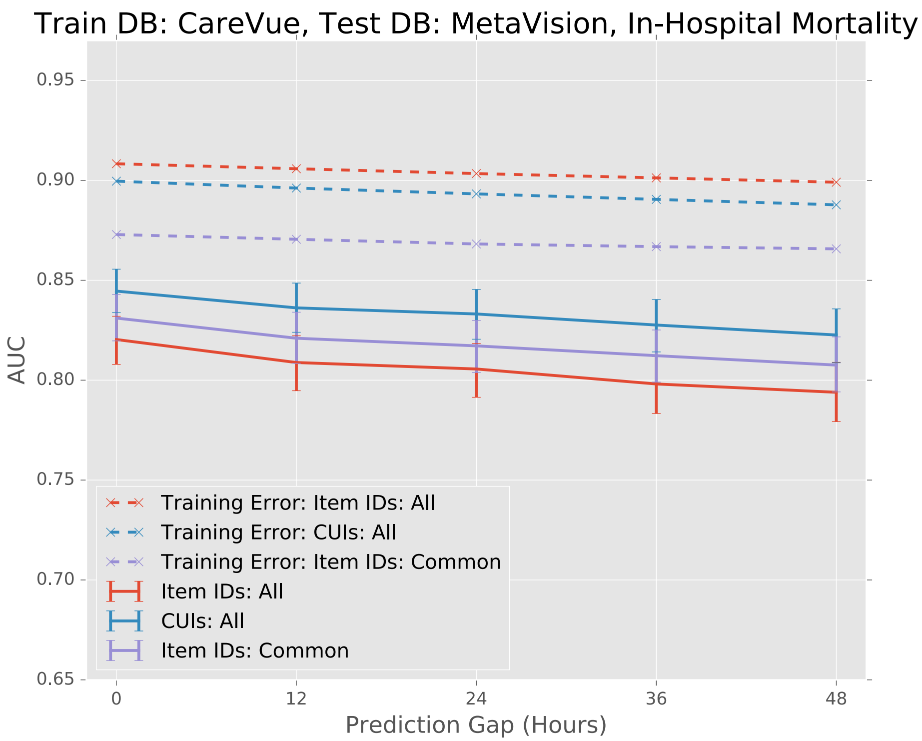


Baseline 1: all



Baseline 2: common

What happens when we apply models across EHRs?

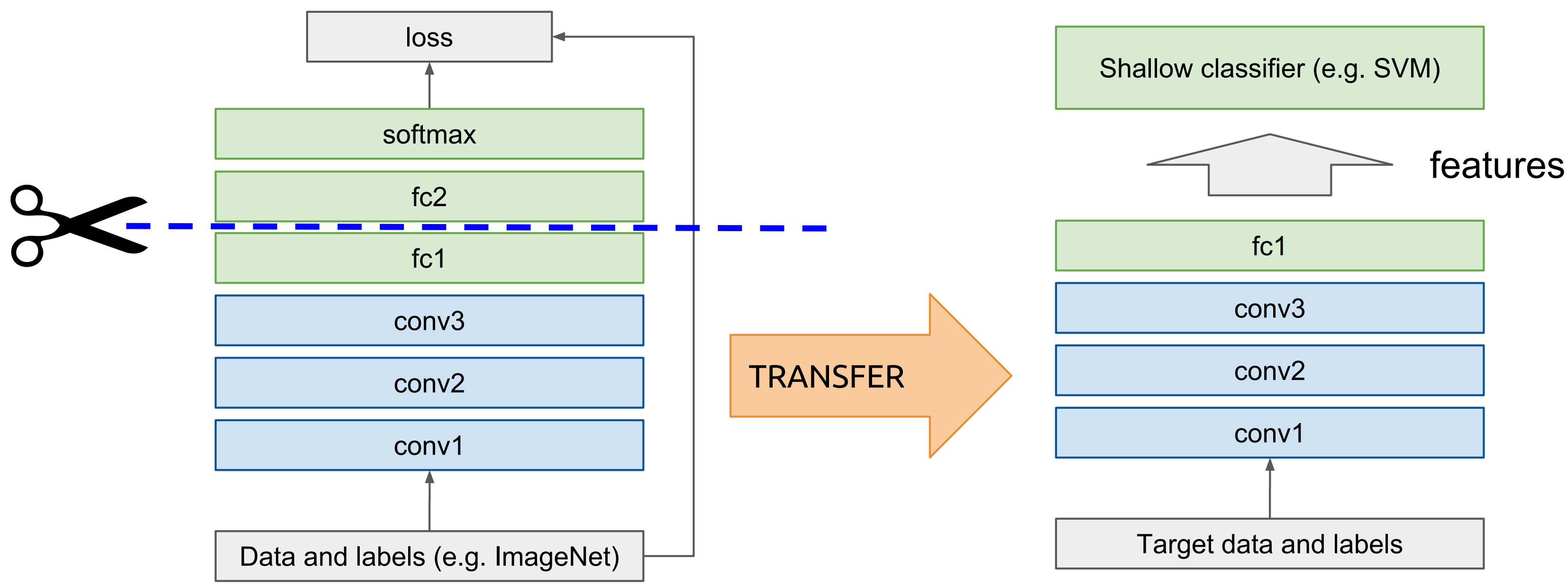


Transfer learning

- 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

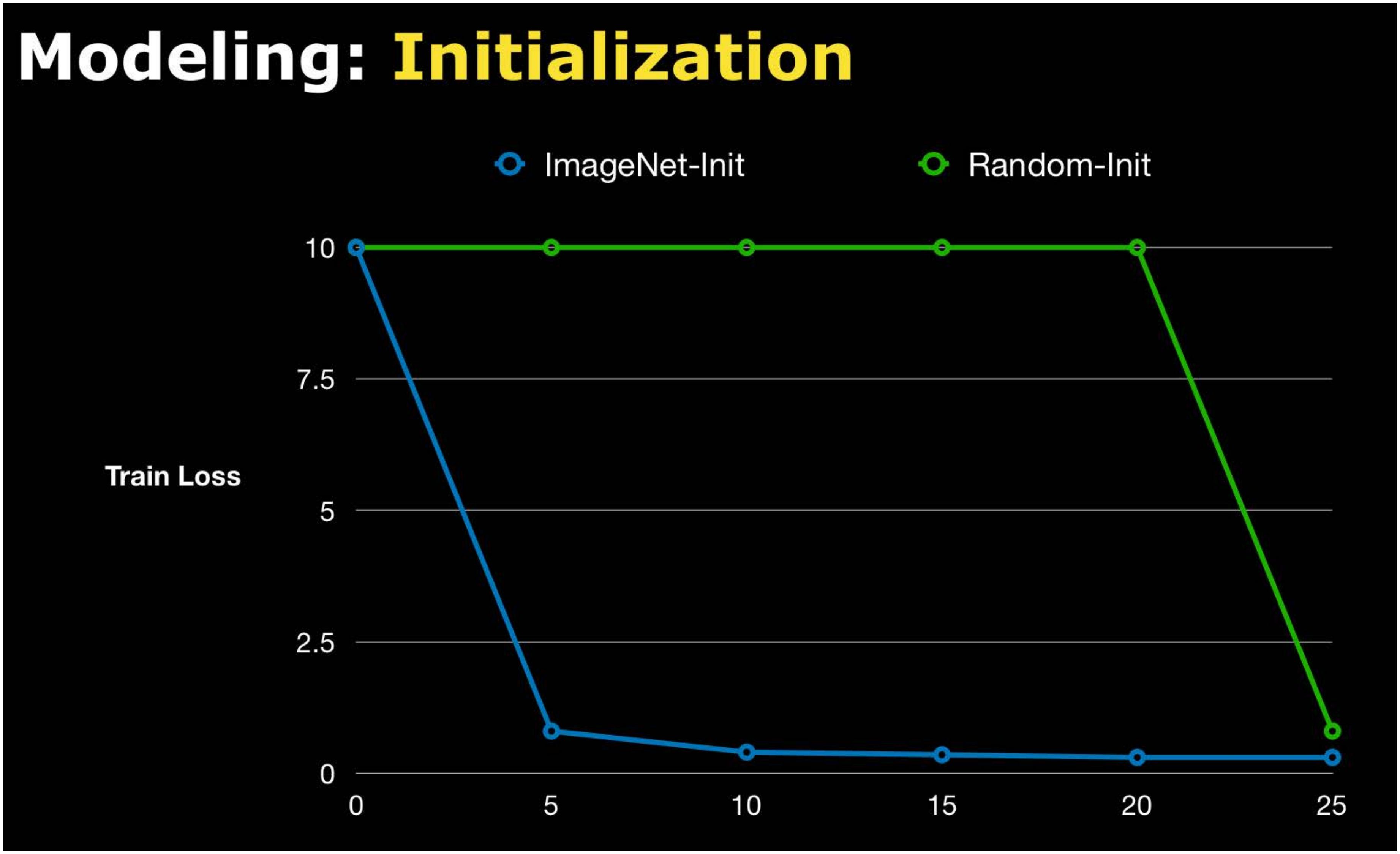
Transfer learning for feedforward networks

- 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

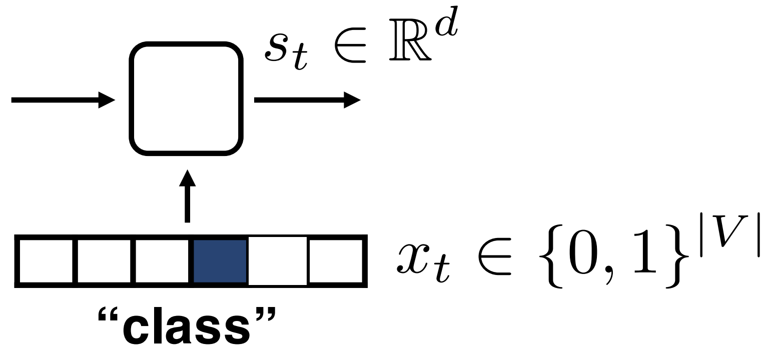
Transfer learning for feedforward networks



[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:

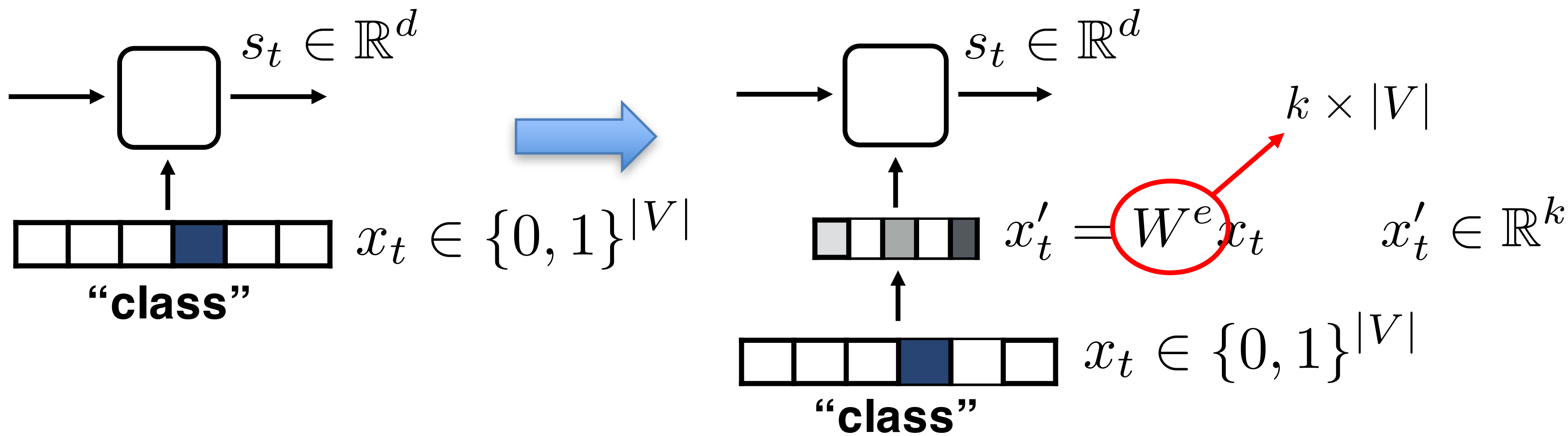
$$s_t = \tanh(W^{s,s} s_{t-1} + W^{s,x} x_t)$$

dimension
$d \times |V|$

- **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

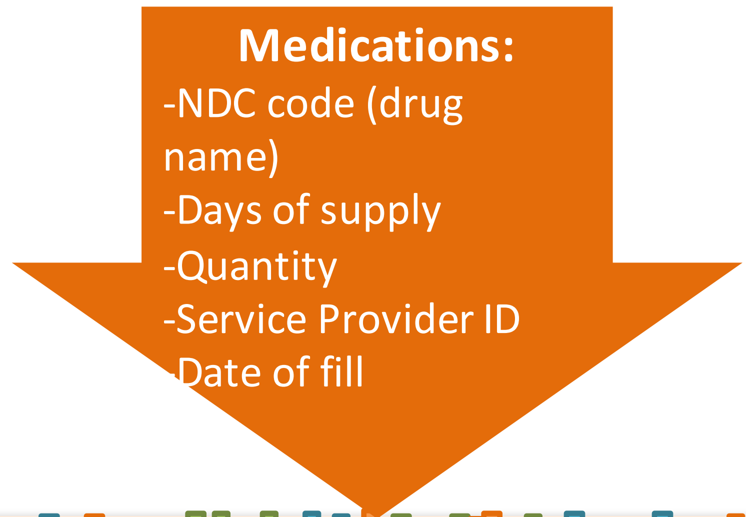
Method	i2b2 2010		i2b2 2012		Semeval 2014 Task 7		Semeval 2015 Task 14	
	General	MIMIC	General	MIMIC	General	MIMIC	General	MIMIC
w2v	-	82.67	-	73.77	-	72.49	-	73.96
GloVe	84.08	85.07	74.95	75.27	70.22	77.73	72.13	76.68
fastText	83.46	84.19	73.24	74.83	69.87	76.47	72.67	77.85
ELMo	83.83	87.80	76.61	80.5	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	-

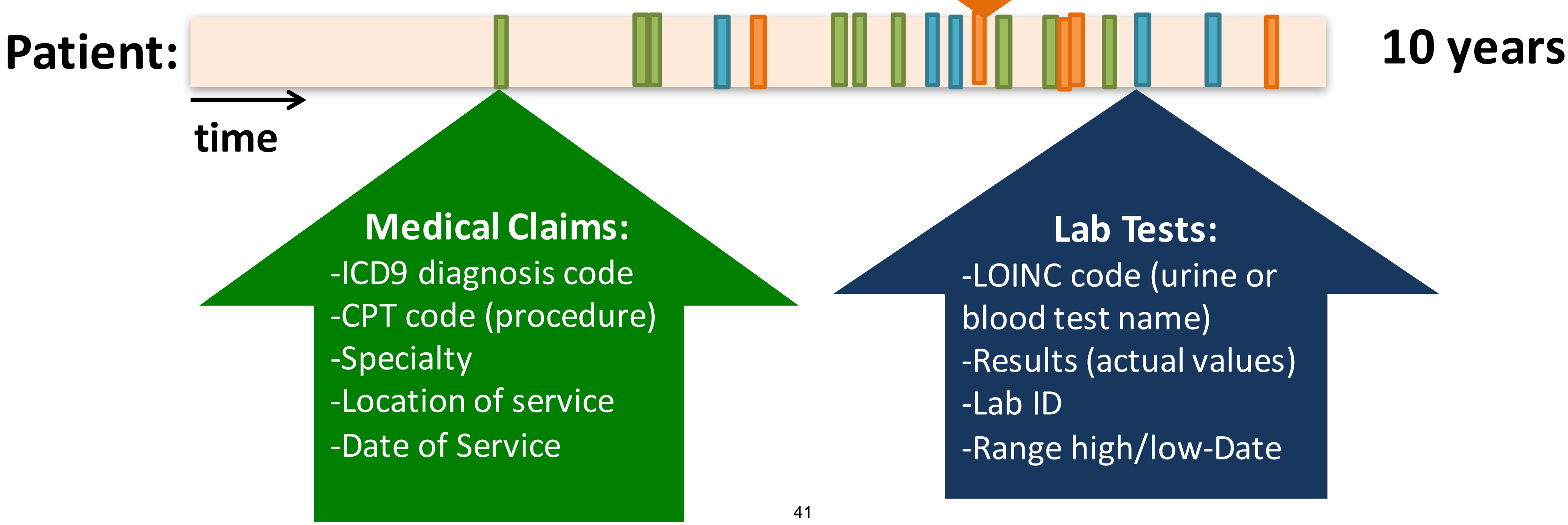
Table 3: Test set comparison in exact F-measure of embedding methods across tasks.

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

Transfer learning for recurrent neural networks

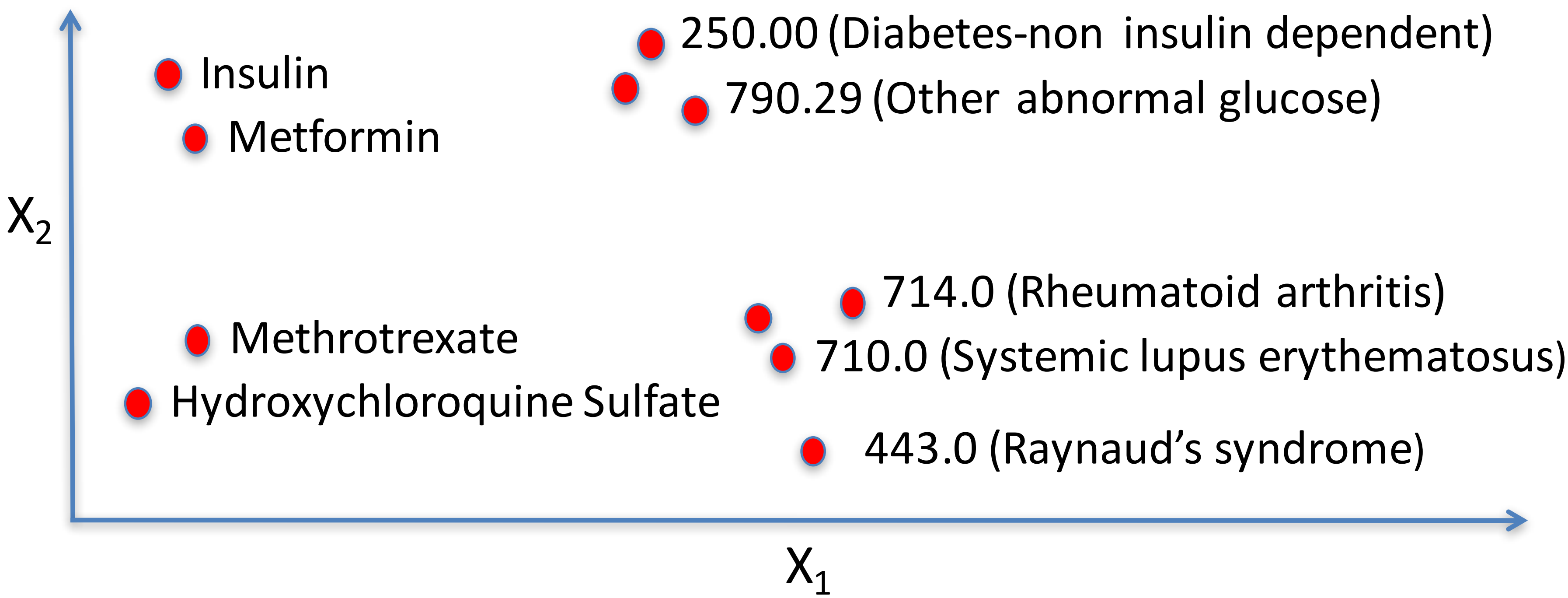
Can we use these techniques for longitudinal patient records (non-textual data)?





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…?



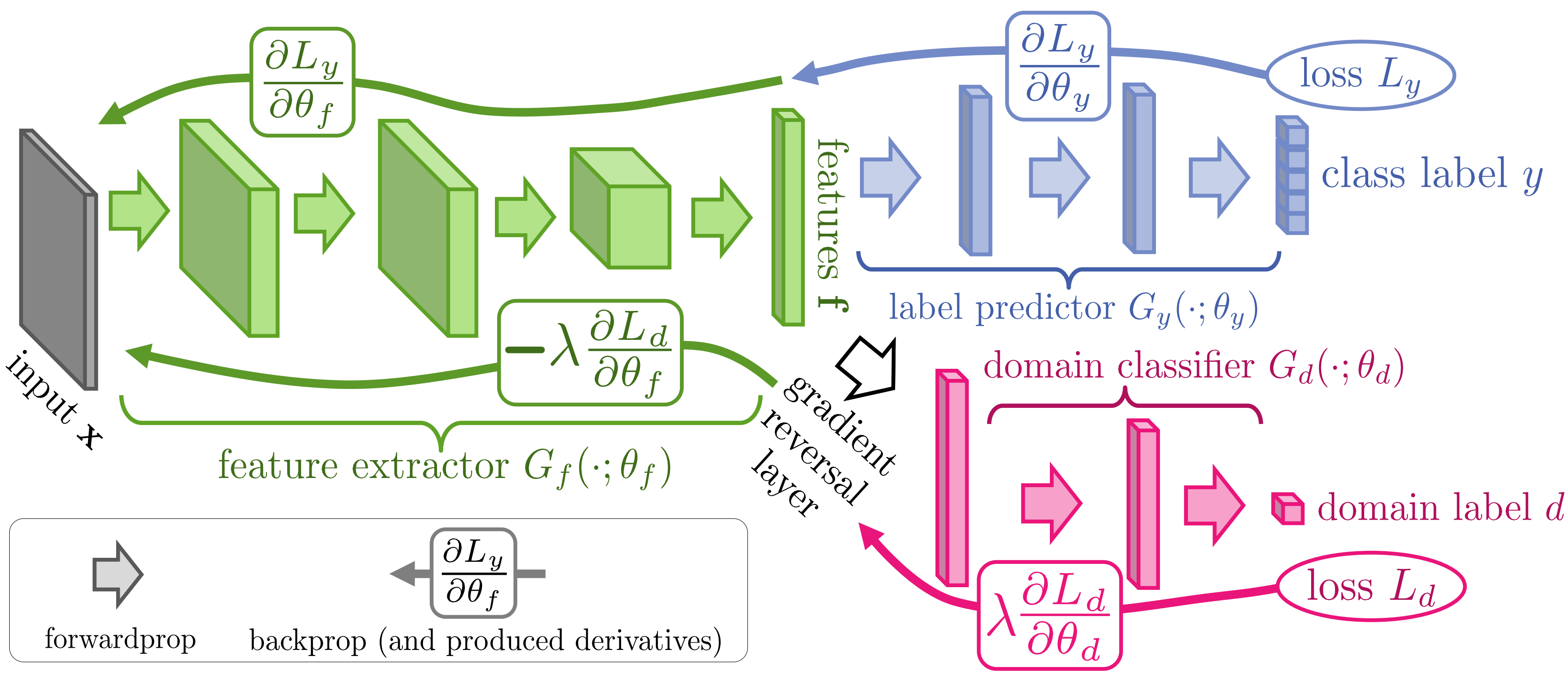
[Choi, Chiu, Sontag, Learning Low-Dimensional Representations of Medical Concepts, AMIA CRI 2016;
Choi, Bahadori et al., Multi-Layer Representation Learning for Medical Concepts, KDD 2016;
Beam et al., Clinical Concept Embeddings Learned from Massive Sources…, arXiv:1804.01486, 2018]

Transfer learning

- 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*:



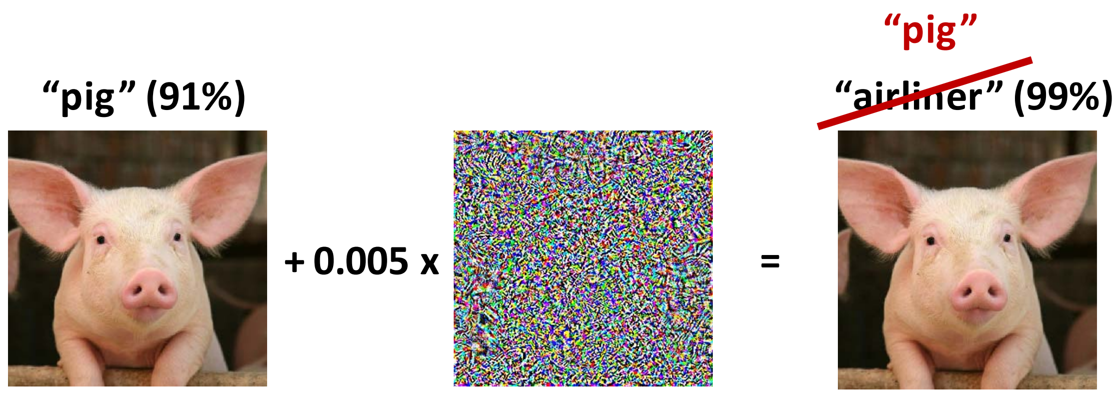
Ganin et al., Domain-Adversarial Training of Neural Networks. JMLR '16

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?

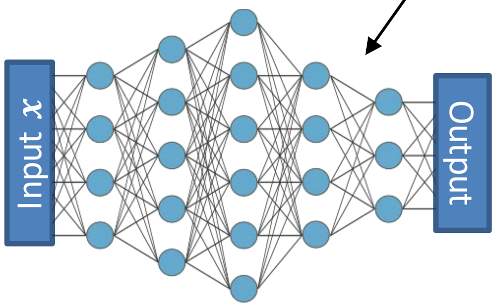
To get an adv. example

~~**Goal of**~~ **training:**

Model Parameters Input Correct Label

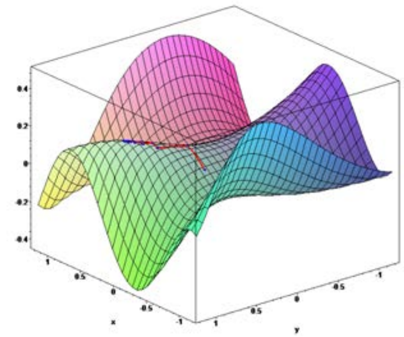
$$min_{\theta}\ loss(\theta, x, y)$$

Differentiable



Parameters $\boldsymbol{\theta}$

Can use gradient descent method to find good θ



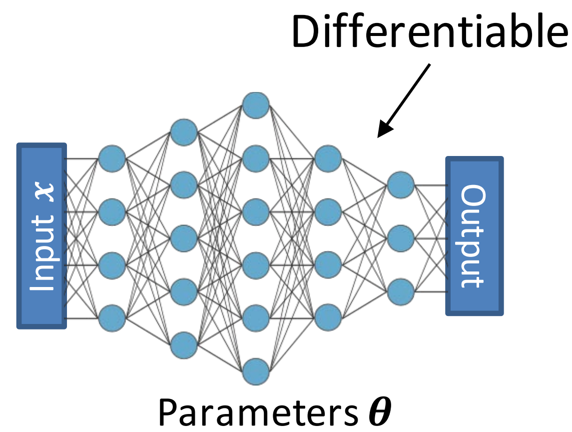
Slide credit: Aleksander Madry

Where Do Adversarial Examples Come From?

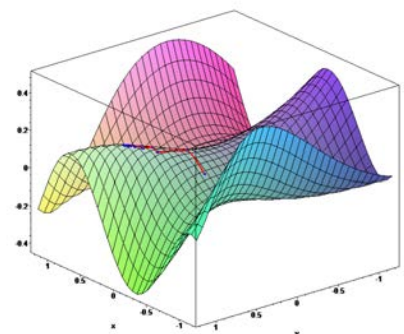
To get an adv. example

~~**Goal of training:**~~

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



Can use gradient descent method to find good θ



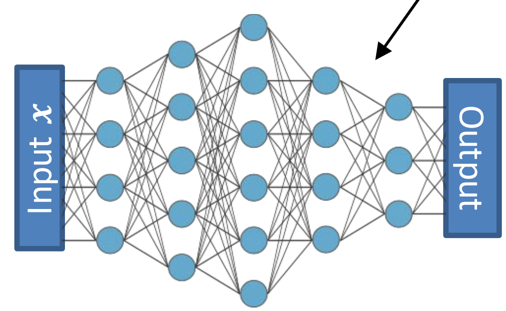
Slide credit: Aleksander Madry
Used with permission.

Where Do Adversarial Examples Come From?

To get an adv. example

~~Goal of~~ training:

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



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
Used with permission.

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
Used with permission.

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
Used with permission.

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
Used with permission.