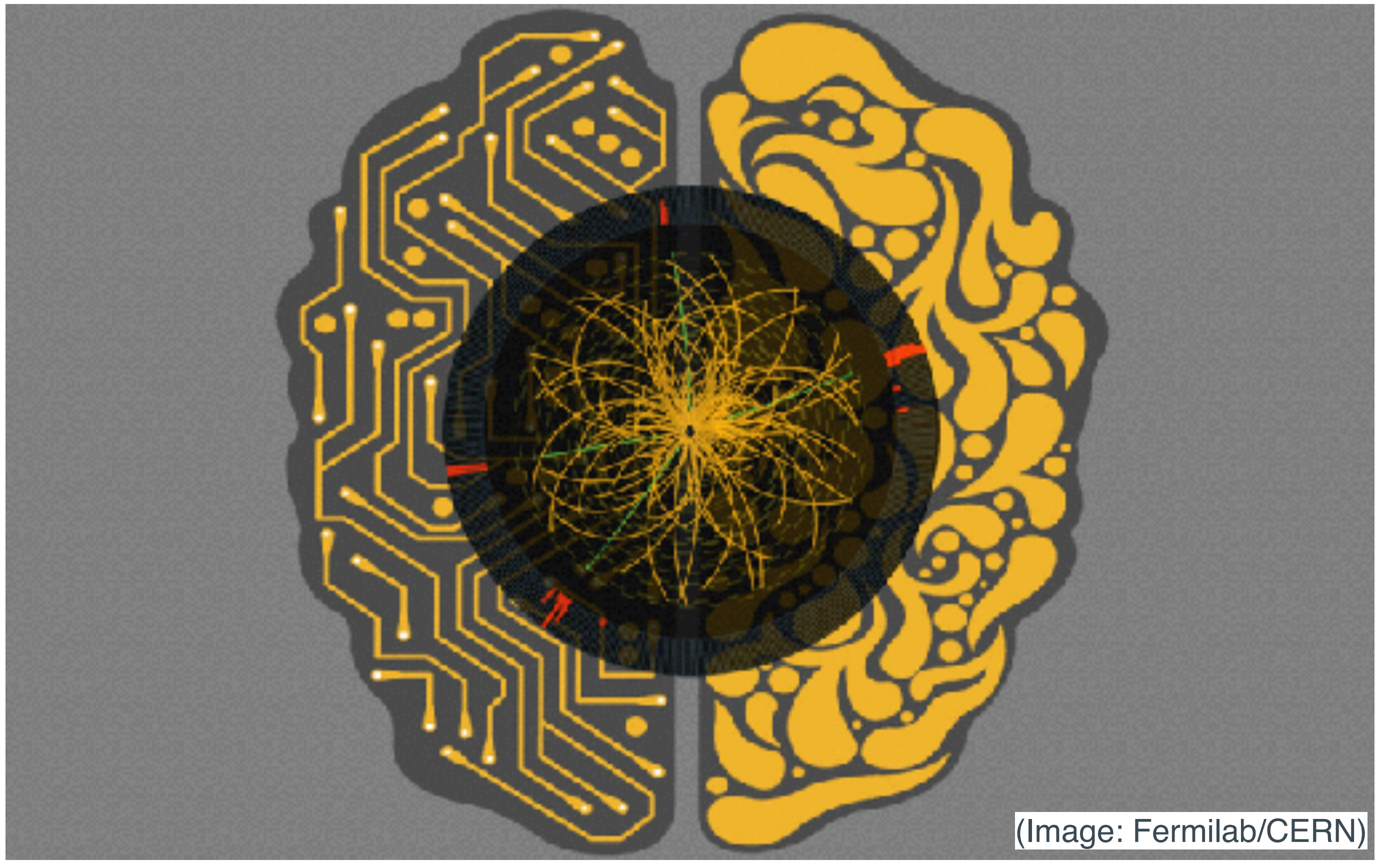
PHY 835: Collider Physics Phenomenology

Machine Learning in Fundamental Physics

Gary Shiu, UW-Madison



(Image: Fermilab/CERN)

Lecture 5: Logistic Regression

Physics ∩ ML

a virtual hub at the interface of theoretical physics and deep learning.

---

10 Feb 2021

Physics meets ML to solve cosmological inference

Ben Wandelt, Institut d'Astrophysique de Paris / Institut Lagrange, Sorbonne University and Center for Computational Astrophysics, Flatiron Institute, New York, 12:00 EDT

Abstract: The goal of cosmological inference is to learn about the origin, composition, evolution, and fate of the cosmos from all accessible sources of astronomical data, such as the cosmic microwave background, galaxy surveys, or electromagnetic and gravitational wave transients. Traditionally, the field has progressed by designing and modeling intuitive summaries of the data, such as n-point correlations. This traditional approach has a number of risks and limitations: how do we know if we computed the most informative statistics? Did we forget any summaries that would have provided additional information or break parameter degeneracies? Did we take into account all the ways the model is affecting the data? To be feasible, the traditional approach imposes approximations on the statistical modeling (e.g. the likelihood form) and on the physical modeling. I will discuss a new mode of cosmological inference: simulation-based, full-physics modeling, made feasible through multiple advances in 1) machine-learning, 2) in the way we design and run simulations of cosmological observables, and 3) in how we compare models to data. The goal is to use current and next generation data to reconstruct the cosmological initial conditions and constrain cosmological physics much more completely than has been feasible in the past. I will discuss current status, and ways to meet the new challenges inherent in this approach, including robustness to model misspecification.

Recap of Lecture 4

- Linear regression
- Regularization (Ridge, LASSO)
- MLE
- MAP
- Relation of MLE and MAP with Least/Square and Ridge regression
- Linear regression will be replaced by more complicated/non-linear models
- Regression on the 1D Ising model

Outline for today

- Logistic classification (binary classification)
- Binary cross-entropy
- Multi-class classification
- MNIST

References: 1803.08823

Logistic Regression

Logistic Regression

- Discrete variables and not continuous output, determine categories (cat or dog, ordered or disordered phase, SUSY or background).

- Start with binary classification, will generalize later to multi-class.

- Data labels for M classes:

$$m \in \{0, \ldots, M-1\}$$

- Task: predict correct labels/features from input design matrix:

$$X \in \mathbb{R}^{n \times p}$$ **n samples, p features**

- Backbone of modern **supervised deep learning** models.

Linear Classifier

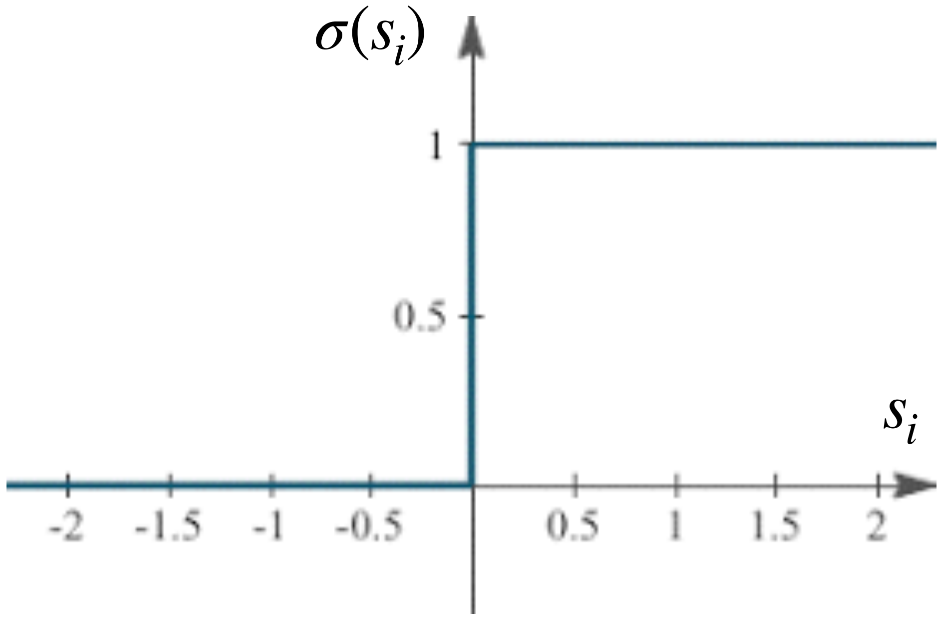
- Categorize data using a weighted linear-combination of features and an additive constant:

short-hand

$$s_i = \boldsymbol{x}_i^T \boldsymbol{w} + b_0 \equiv \mathbf{x}_i^T \mathbf{w},$$

$$\mathbf{x}_i = (1, \boldsymbol{x}_i) \text{ and } \mathbf{w} = (b_0, \boldsymbol{w}).$$

- Map output of a linear regression to



$\sigma(s_i) = 1$ if $s_i \geq 0$ and 0 otherwise

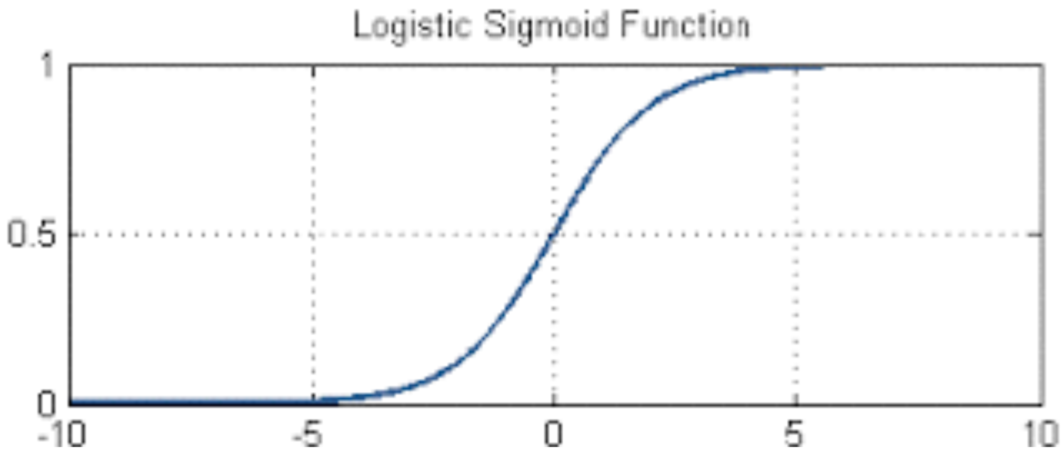
known as perceptron in the ML literature

- Perceptron is not differentiable (hard to train via gradient descent).

Sigmoid Function

- Soft classifier that is differentiable (allows for training). Instead of discrete output, the classifier returns the probability in a category.

- One such function is the logistic (or sigmoid) function:



$$\sigma(s) = \frac{1}{1 + e^{-s}}.$$

- The sigmoid function is differentiable, and satisfies some useful properties:

$$1 - \sigma(s) = \sigma(-s)$$

$$\sigma'(s) = \sigma(s)\big(1 - \sigma(s)\big)$$

$$\sigma'(s) = \sigma'(-s)$$

Soft Classifier

- Probability that a data point belongs to a category $y_i = \{0,1\}$:

$$P(y_i = 1|\boldsymbol{x}_i, \boldsymbol{\theta}) = \frac{1}{1 + \mathrm{e}^{-\mathbf{x}_i^T \boldsymbol{\theta}}},$$

$$P(y_i = 0|\boldsymbol{x}_i, \boldsymbol{\theta}) = 1 - P(y_i = 1|\boldsymbol{x}_i, \boldsymbol{\theta}),$$

- Motivated by the two-state system in statistical mechanics:

$$P(y_i = 0) = \frac{\mathrm{e}^{-\beta\epsilon_0}}{\mathrm{e}^{-\beta\epsilon_0} + \mathrm{e}^{-\beta\epsilon_1}} = \frac{1}{1 + \mathrm{e}^{-\beta\Delta\epsilon}},$$

$$P(y_i = 1) = 1 - P(y_i = 0).$$

only energy difference
$\Delta\epsilon = \epsilon_1 - \epsilon_0$
is observable

- In terms of the sigmoid function:

$$P(y_i = 1) = \sigma(\mathbf{x}_i^T \mathbf{w}) = 1 - P(y_i = 0).$$

Constructing the Loss Function

- The likelihood of observing the data:

$$P(\mathcal{D}|\mathbf{w}) = \prod_{i=1}^{n} \left[\sigma(\mathbf{x}_i^T \mathbf{w})\right]^{y_i} \left[1 - \sigma(\mathbf{x}_i^T \mathbf{w})\right]^{1-y_i}$$

- The log-likelihood:

$$l(\mathbf{w}) = \sum_{i=1}^{n} y_i \log \sigma(\mathbf{x}_i^T \mathbf{w}) + (1 - y_i) \log \left[1 - \sigma(\mathbf{x}_i^T \mathbf{w})\right].$$

- **Maximum Likelihood Estimation (MLE):**

$$\hat{\mathbf{w}} = \arg\max_{\theta} \sum_{i=1}^{n} y_i \log \sigma(\mathbf{x}_i^T \mathbf{w}) + (1 - y_i) \log \left[1 - \sigma(\mathbf{x}_i^T \mathbf{w})\right]$$

- **The cost (error) function:**

$$\begin{aligned} \mathcal{C}(\mathbf{w}) &= -l(\mathbf{w}) \\ &= \sum_{i=1}^{n} -y_i \log \sigma(\mathbf{x}_i^T \mathbf{w}) - (1 - y_i) \log \left[1 - \sigma(\mathbf{x}_i^T \mathbf{w})\right]. \end{aligned}$$

cross entropy

Optimizing the Loss Function

- The cross entropy is a **convex function** of the weights; any local minimizer is a global minimizer.

- The cross entropy is differentiable, can be minimized via SD:

$$\mathbf{0} = \nabla \mathcal{C}(\mathbf{w}) = \sum_{i=1}^{n} \left[\sigma(\mathbf{x}_i^T \mathbf{w}) - y_i \right] \mathbf{x}_i,$$

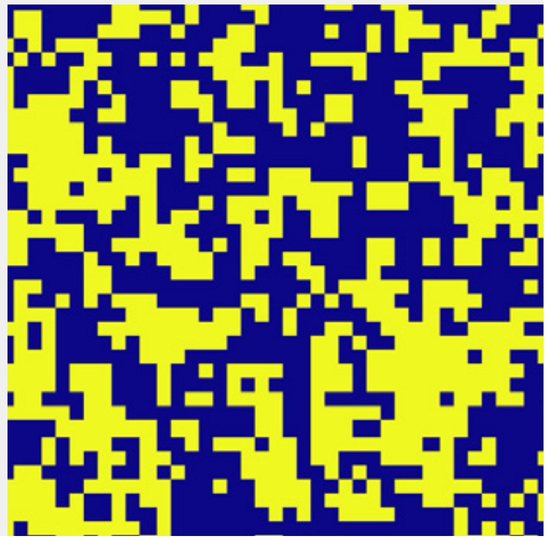
- We can supplement the cross entropy with additional **regularizers** such as L^1 and L^2 regularization.

- Modifications such as adding stochasticity (e.g., mini-batches) and momentum discussed in Lecture 3 also apply.

Phases of 2D Ising Model

Phases of the 2D Ising Model

- The Hamiltonian for the 2D Ising Model:



$$H = -J \sum_{\langle ij \rangle} S_i S_j, \qquad S_j \in \{\pm 1\},$$

nearest neighbors

- 2D lattice of L x L spins.

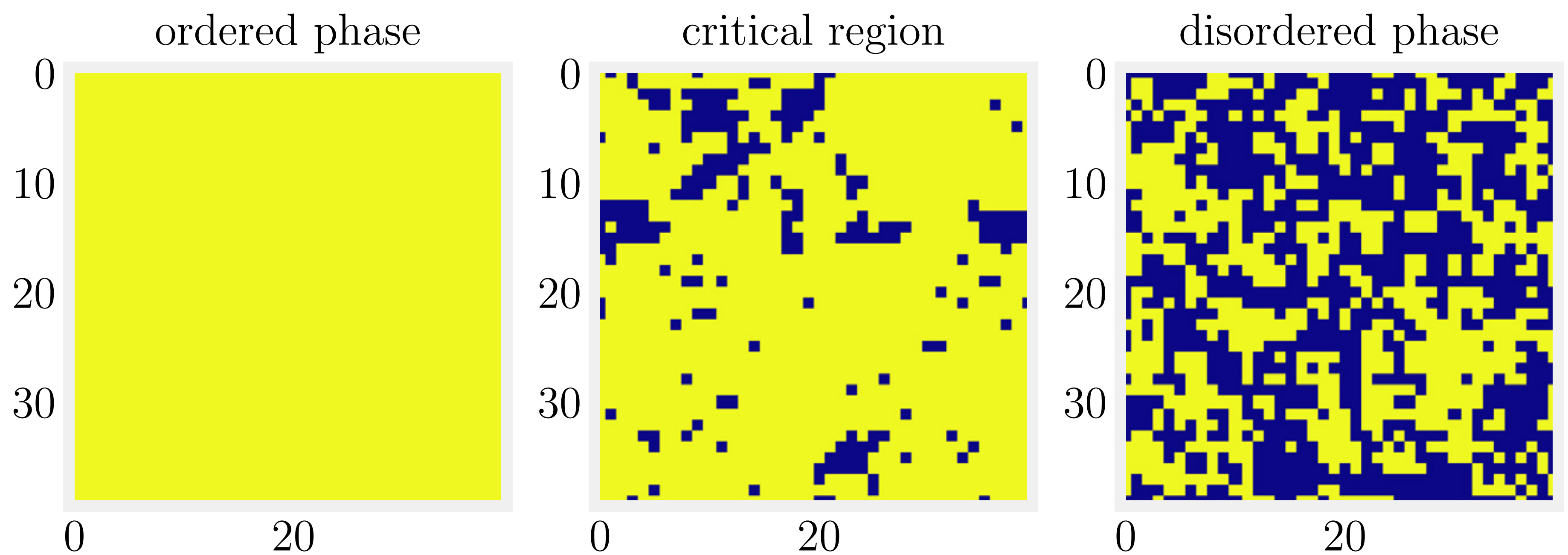
- Periodic boundary conditions.

- Onsager's exact solution: a phase transition in the thermodynamic limit at the **critical temperature**:

$$T_c/J = 2/\log(1+\sqrt{2}) \approx 2.26$$

Phases of the 2D Ising Model

- Can we train a binary classifier to distinguish between two phases of the 2D Ising model?

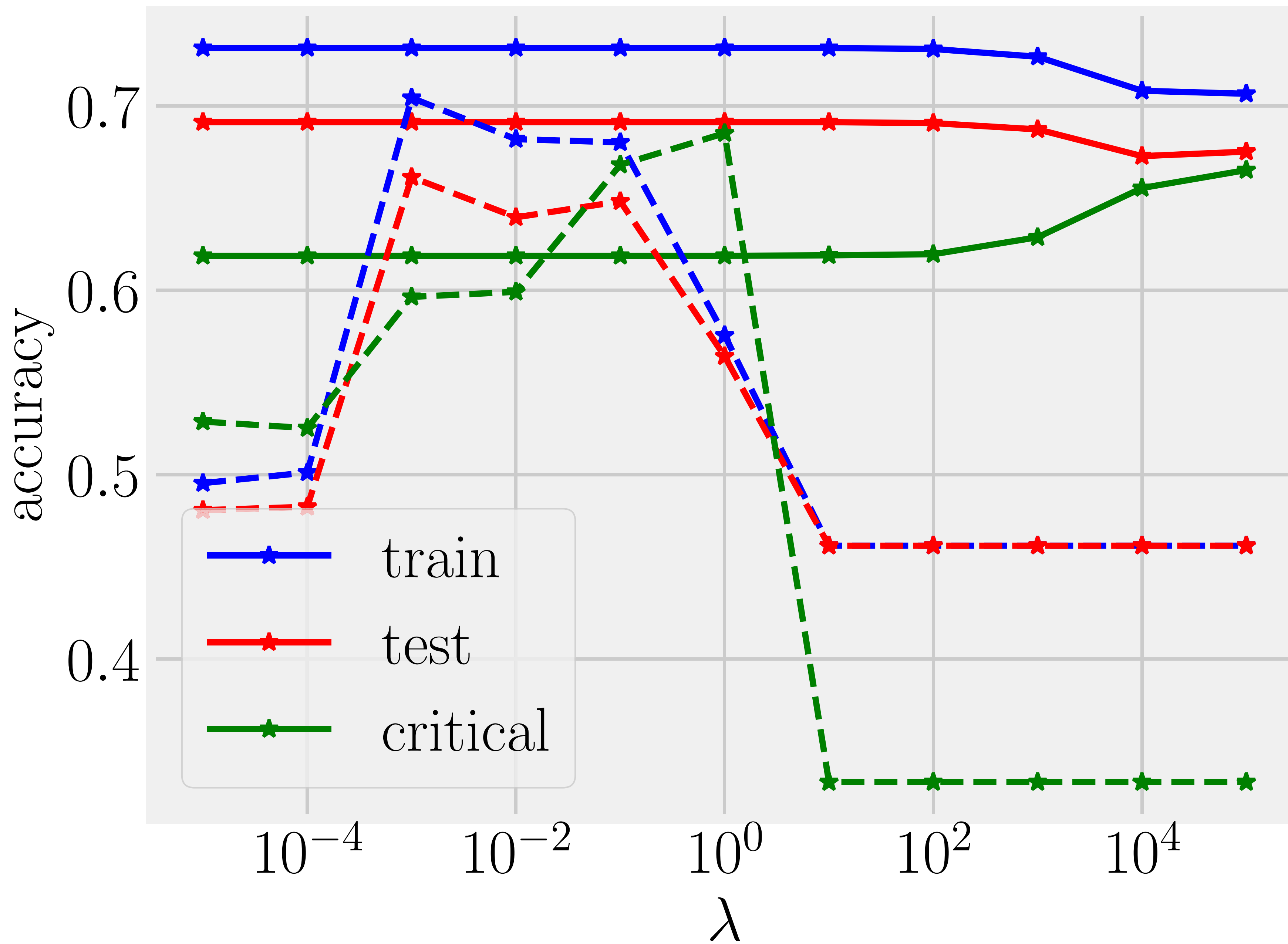


- We need a dataset, i.e. samples at a given temperature. How do we do this? One common way: Monte Carlo Simulations.
- Our binary classifier misses features like contiguous ordered 2D domains; such info can be incorporated using deep convoluted neural networks (CNNs) and topological data analysis.

Phases of the 2D Ising Model

- Generate a dataset for 40x40 grid using MC simulations to prepare 10^4 states at every temperature T.
- We know which temperatures the samples are from, and their labels (e.g., 0=disordered, 1=ordered).
- What we are doing is called **supervised learning.**
- Later in the course we will see methods which do not need these labels, i.e. **unsupervised learning**.
- For physics in practice: supervised learning can teach you how well a method is working for a desired task. To do something "new", we usually have to use unsupervised learning.

Phases of the 2D Ising Model



lib linear (Schkit)

SGD

Experiment with Juypter Notebook 6:

https://physics.bu.edu/%7Epankajm/MLnotebooks.html

SUSY vs SM Background

SUSY vs SM Background

- Using the dataset from the UC Irvine ML repository produced by MC simulations to contain events with 2 leptons (electrons or muons)
- These events with 2 leptons with large p_T can occur in SUSY models or within the SM.
- 18 kinematic variables ("features") are recorded for each event.
- Can train a logistic regressor to classify the events into SUSY or SM background.

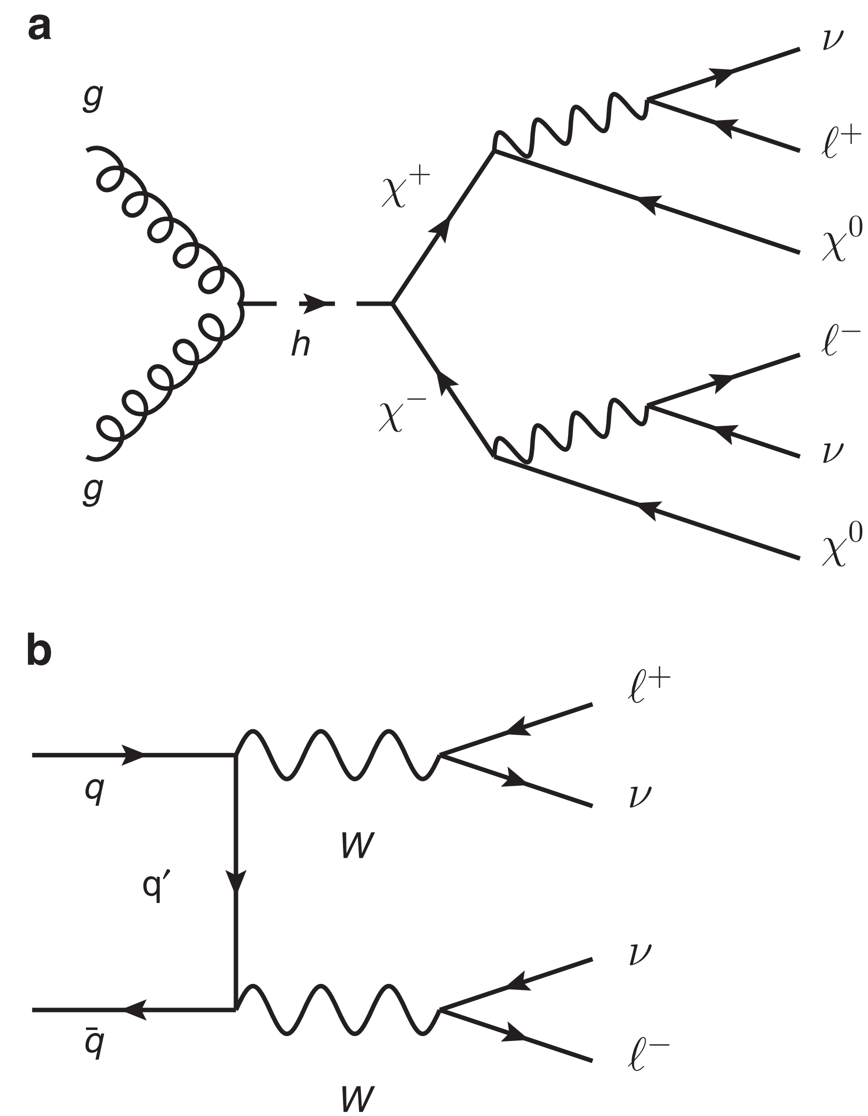
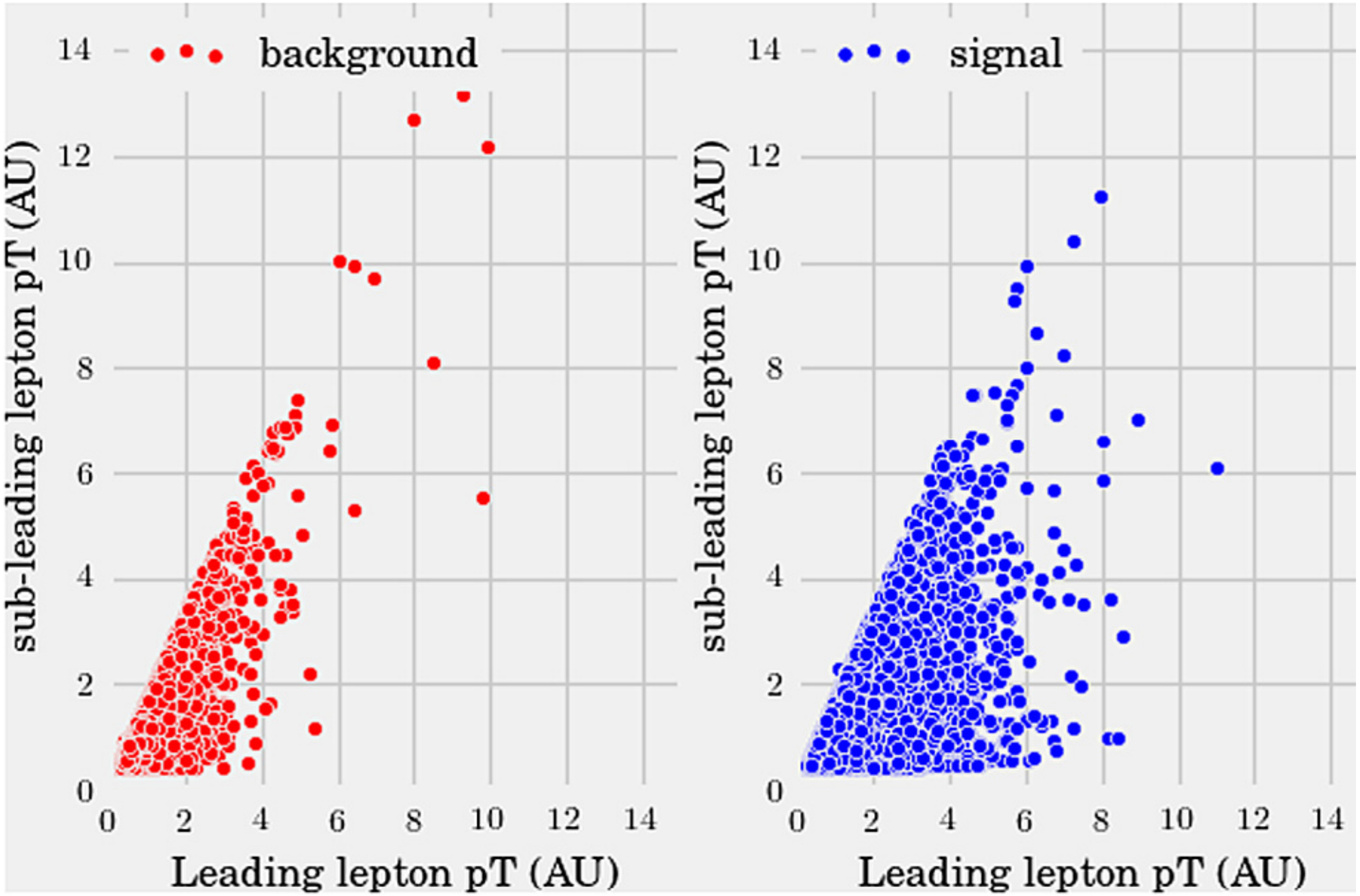


Figure 4 | Diagrams for SUSY benchmark. Example diagrams describing the signal process involving hypothetical supersymmetric particles $\chi^{\pm}$ and χ^0 along with charged leptons $\ell^{\pm}$ and neutrinos ν (**a**) and the background process involving W bosons (**b**). In both cases, the resulting observed particles are two charged leptons, as neutrinos and χ^0 escape undetected.

Baldi et al, Nature Communications, Volume 5, Article number: 4308 (2014)

SUSY vs SM Background



Juypter Notebook 5:
https://physics.bu.edu/%7Epankajm/MLnotebooks.html

Multi-class Classification

Softmax Regression

- Going from 2 labels to M labels.
- Treat the label as a vector $\mathbf{y}_i \in \mathbb{Z}_2^M$, i.e., **one hot-encoding** of labels:

$$0 \rightarrow \begin{pmatrix} 1 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{pmatrix} \quad , \quad 1 \rightarrow \begin{pmatrix} 0 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{pmatrix} \quad , \ldots, \quad M-1 \rightarrow \begin{pmatrix} 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 1 \end{pmatrix}$$

- The probability of being in class m' is the **softmax function**:

$$P(y_{im'} = 1 | \boldsymbol{x}_i, \{\mathbf{w}_k\}_{k=0}^{M-1}) = \frac{e^{-\mathbf{x}_i^T \mathbf{w}_{m'}}}{\sum_{m=0}^{M-1} e^{-\mathbf{x}_i^T \mathbf{w}_m}}$$

where $y_{im'} \equiv \{\mathbf{y}_i\}_{m'}$ is the m'-th component of vector $\mathbf{y}$.

Softmax Regression

- Likelihood of this M-class classifier:

$$P(\mathcal{D}|\{\mathbf{w}_k\}_{k=0}^{M-1}) = \prod_{i=1}^{n}\prod_{m=0}^{M-1}[P(y_{im}=1|\boldsymbol{x}_i,\mathbf{w}_m)]^{y_{im}} \times [1-P(y_{im}=1|\boldsymbol{x}_i,\mathbf{w}_m)]^{1-y_{im}}$$

- Cost function:

$$\mathcal{C}(\mathbf{w}) = -\sum_{i=1}^{n}\sum_{m=0}^{M-1} y_{im}\log P(y_{im}=1|\boldsymbol{x}_i,\mathbf{w}_m) + (1-y_{im})\log(1-P(y_{im}=1|\boldsymbol{x}_i,\mathbf{w}_m)) \,.$$

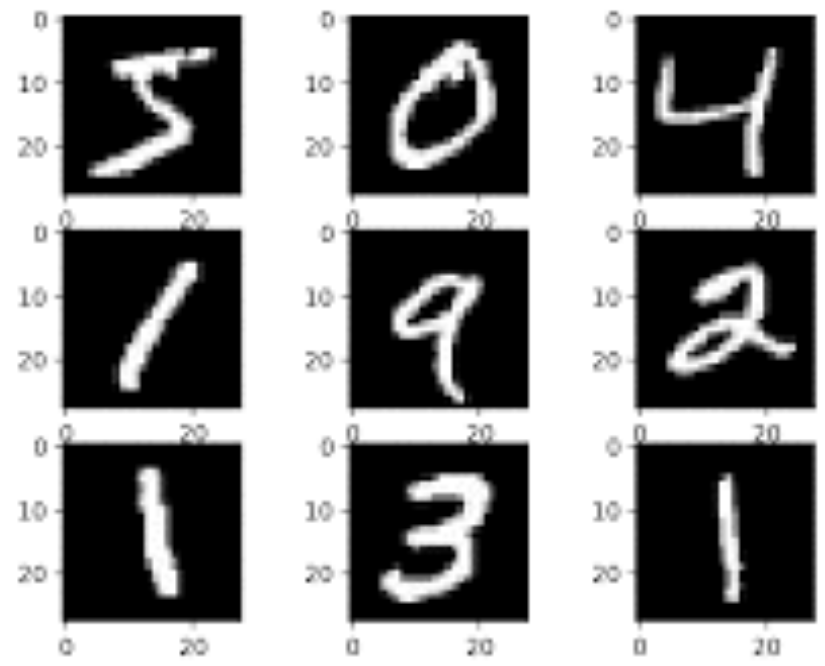
- Activations (one-hot encoding), can be thought of as activating particular cells (e.g. in your brain).
- Becomes a lot harder for a larger number of classes.

Classifying Digits — MNIST

MNIST

- Classifying digits ⇒ M=10 categories

- MNIST = Dataset of handwritten digits, 28x28=784 pixel grid, each assumes 256 grayscale values, interpolating between white and black.



Yann LeCun, Corinna Cortes, Christopher Burges

http://yann.lecun.com/exdb/mnist/

- 60,000 images: 50,000 for training, 10,000 for testing

Experiment with Notebook 7 using softmax:

https://physics.bu.edu/%7Epankajm/MLnotebooks.html

Summary

- Binary classification Logistic sigmoid
- Binary cross-entropy as loss function
- Multi-class classification
- 3 Examples: Phase classification 2D Ising, SUSY datasets, handwritten digits MNIST.