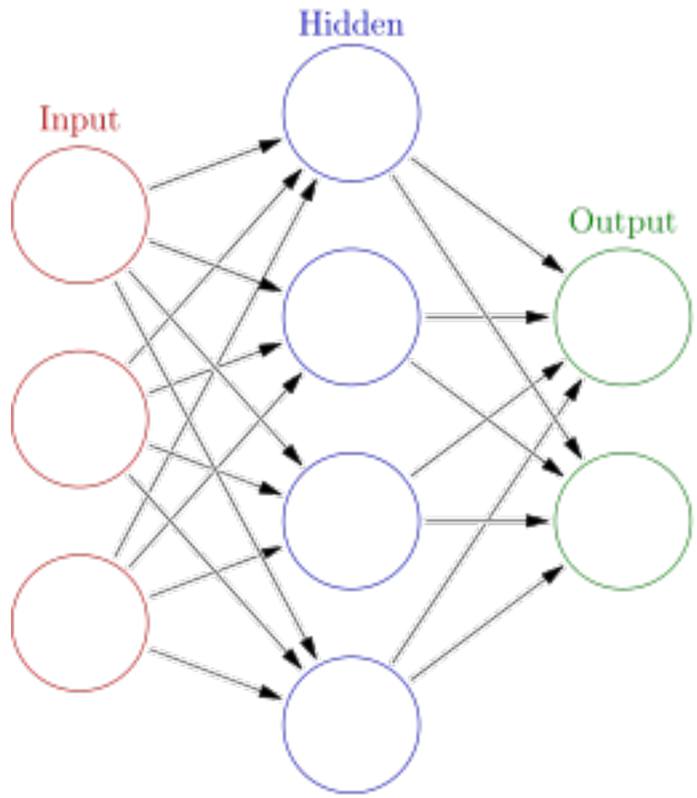
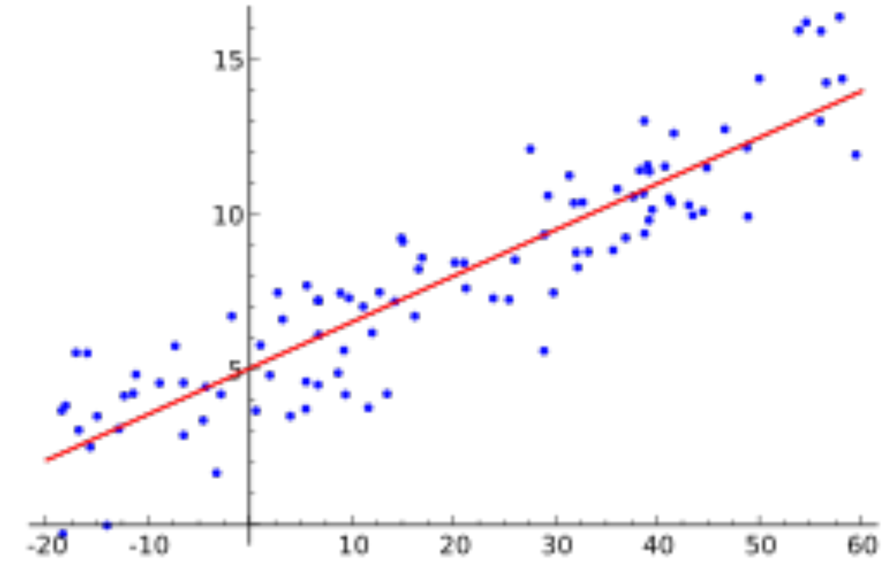
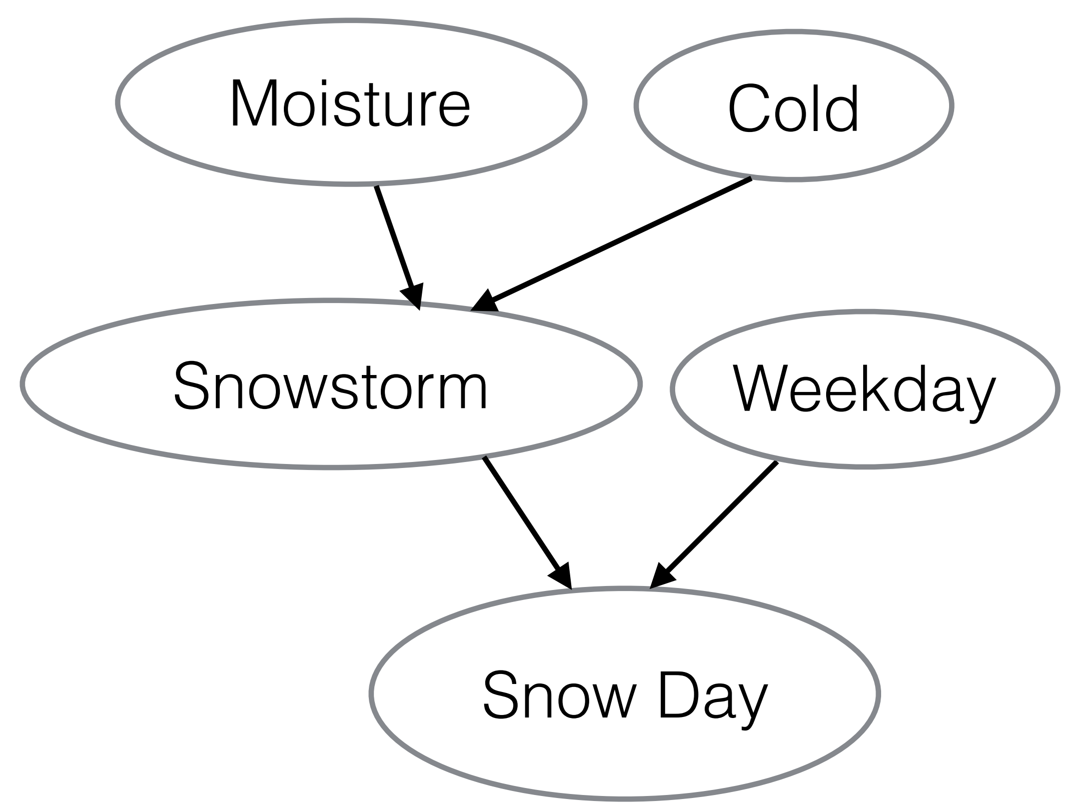
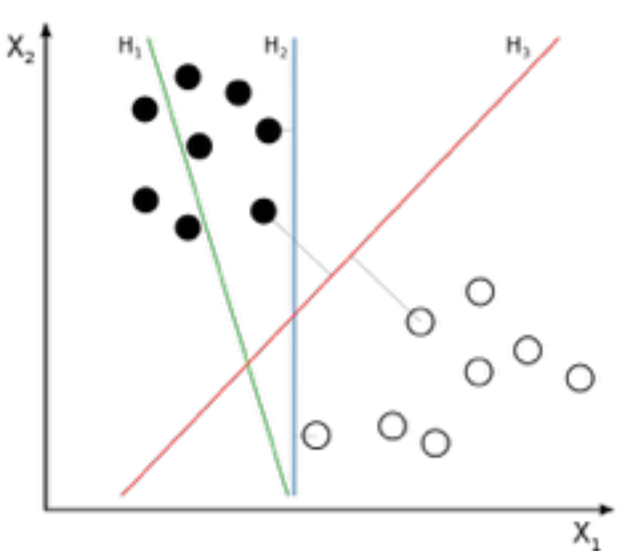




A Whirlwind Tour of ML

IAP 2017





What's the course about?

- Answer the question "What's all this buzz about?"
- Give you a flavor of machine learning: its variety, breadth and power
- Teach the basic vocabulary and concepts so you can study further
- Tools to pick the right approach for a problem

Logistics

- Dates: Jan 24th - Jan 27th
- Time: 3 - 5 pm
- Location: Room: 36 - 156
- Materials to be made available online
- Not for credit

Schedule

Session I: Introduction to ML

Manasi Vartak

Session II: Sampling & Inference

Maggie Makar

Session III: Bayesian Methods

Trevor Campbell

Session IV: Neural Networks

Carl Vondrick

Caveats

- The course topics are not exhaustive
- We are going for breadth as opposed to depth
- Lecture format as opposed to lab
- Taught by grad students; we may not know everything about everything!

Let’s get started!

- We hope you find the material useful
- We will point you to lots of resources
- Please ask questions!

Introduction to ML

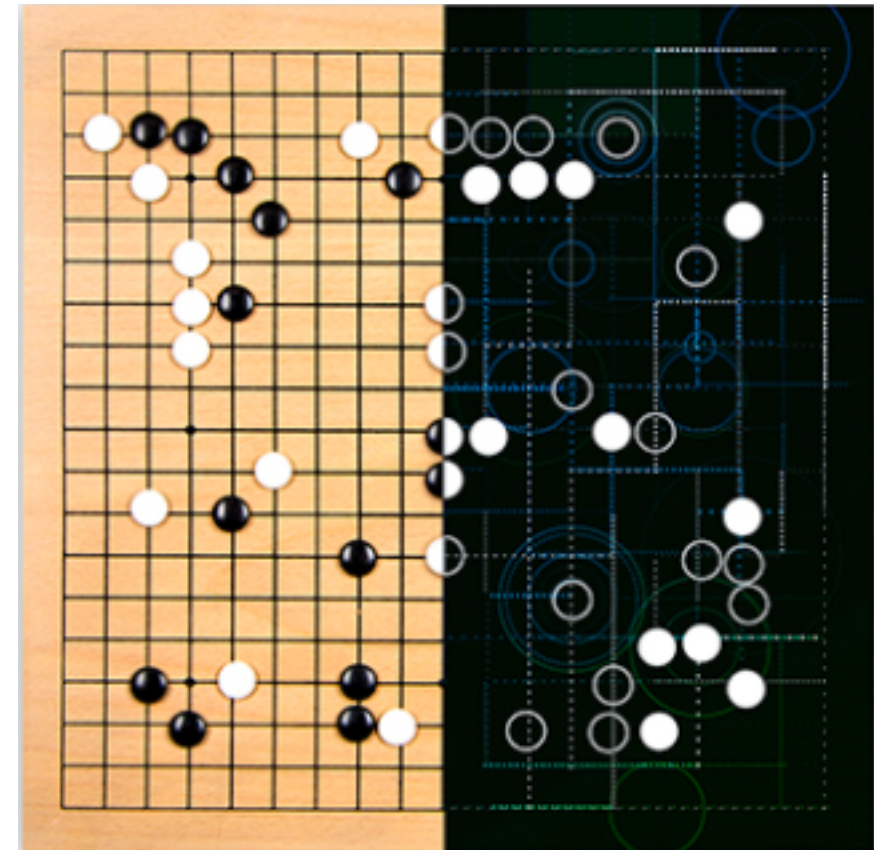
Manasi Vartak
PhD Student, MIT CSAIL
@DataCereal

What is Machine Learning?

Robotics

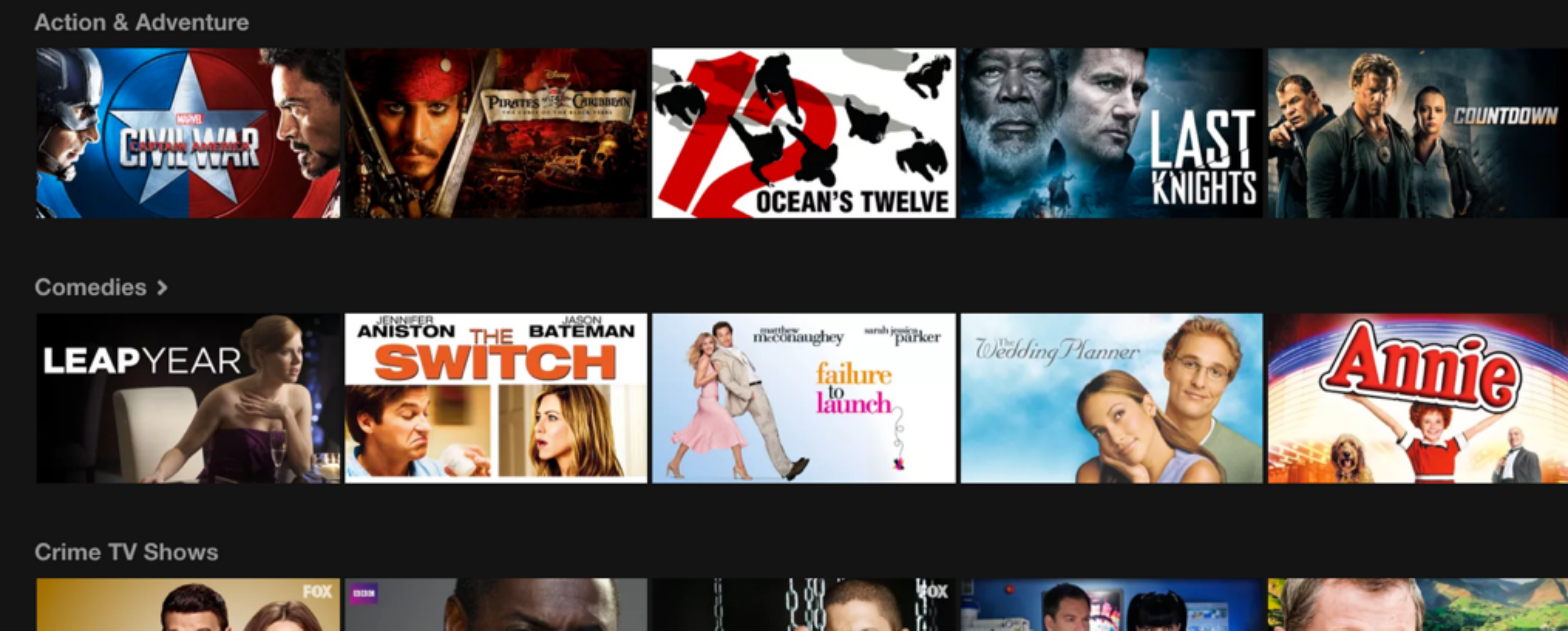
Why Poker Is a Big Deal for Artificial Intelligence

MIT TechReview

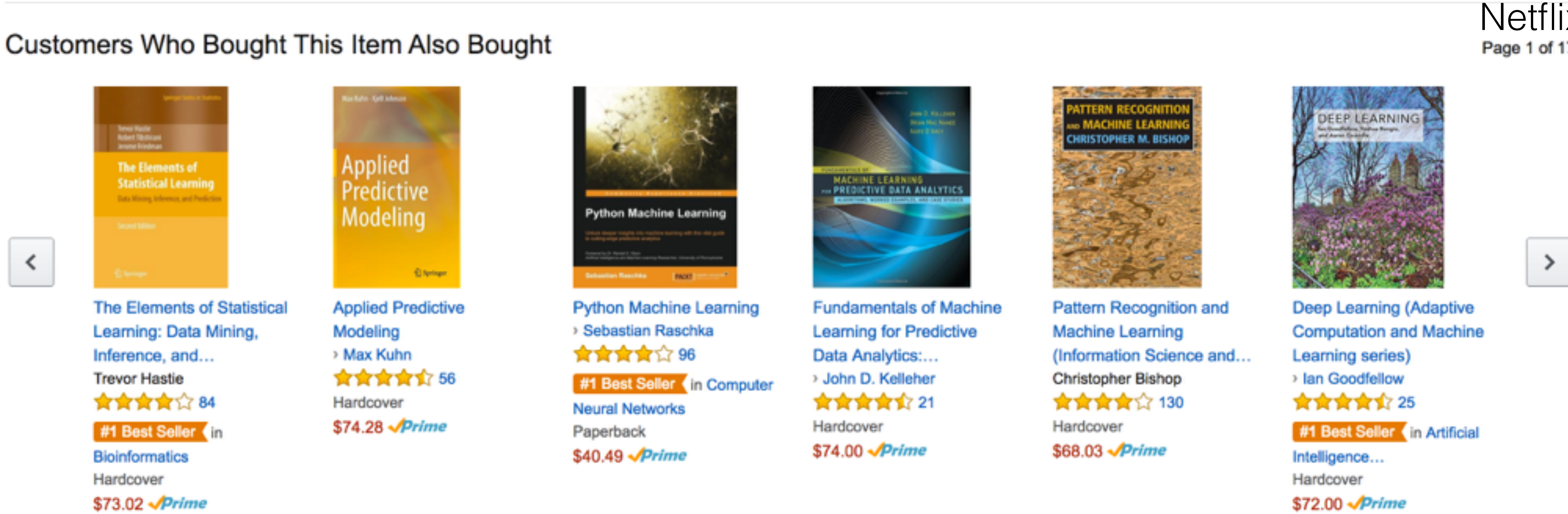


MIT TechReview

Tesla

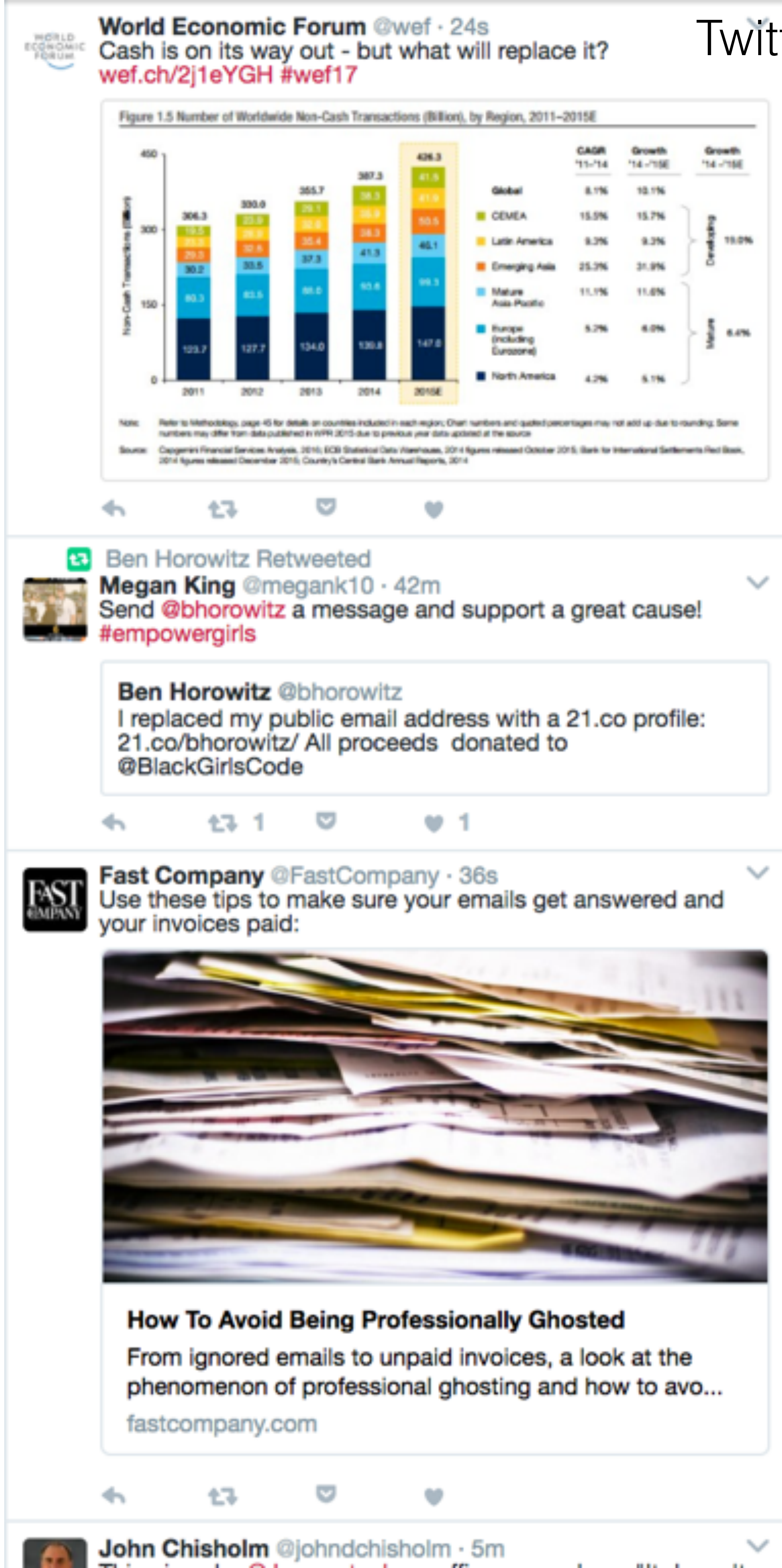


Netflix

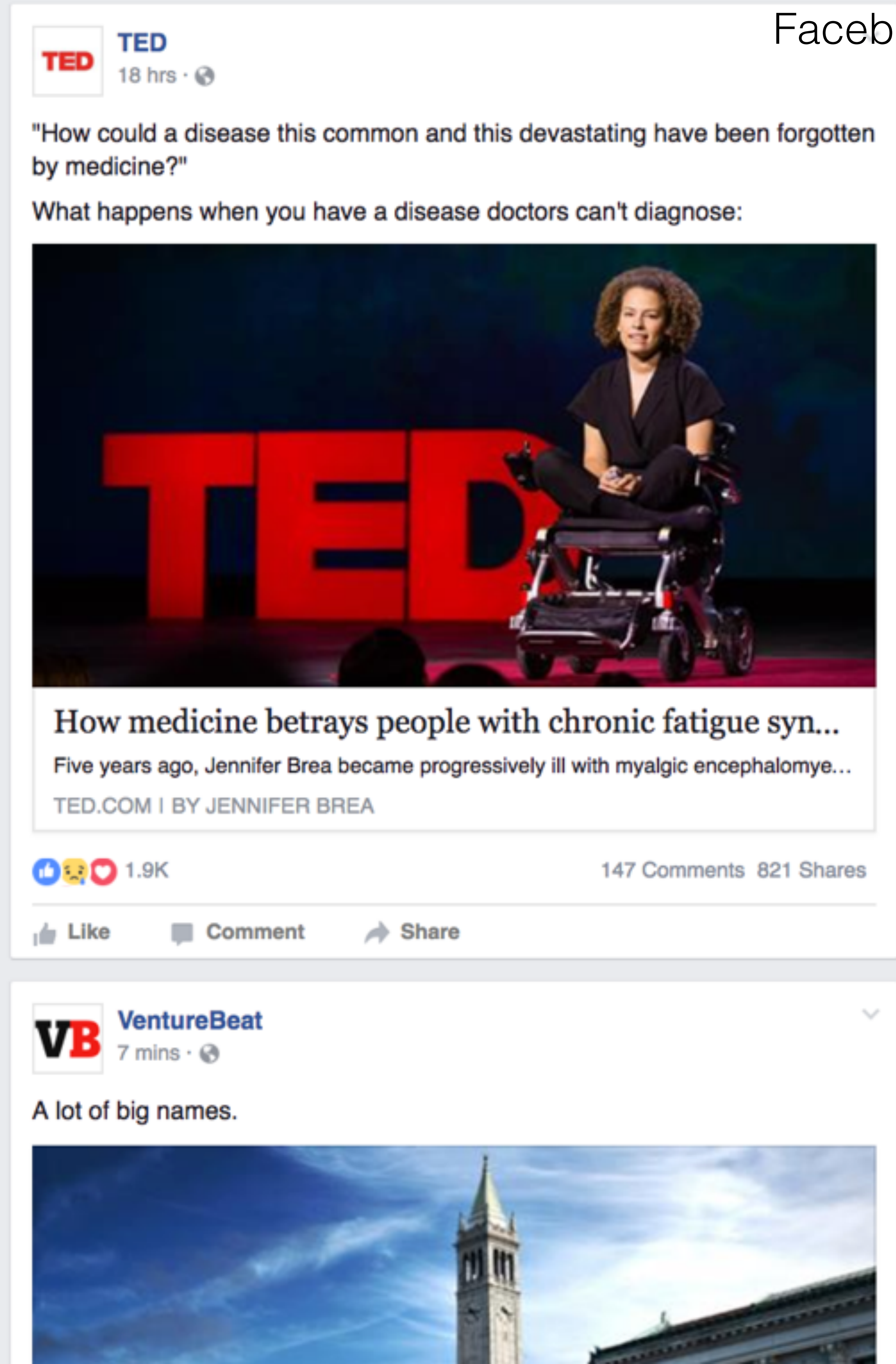


Amazon

Twitter



Facebook



Patient Risk Stratification with Time-Varying Parameters: A Multitask Learning Approach

Systematic chromatin state comparison of epigenomes associated with diverse properties including sex and tissue type

Cross-Corpora Unsupervised Learning of Trajectories in Autism Spectrum Disorders

Sequencing and comparison of yeast species to identify genes and regulatory elements

Unsupervised Learning from Noisy Networks with Applications to Hi-C Data

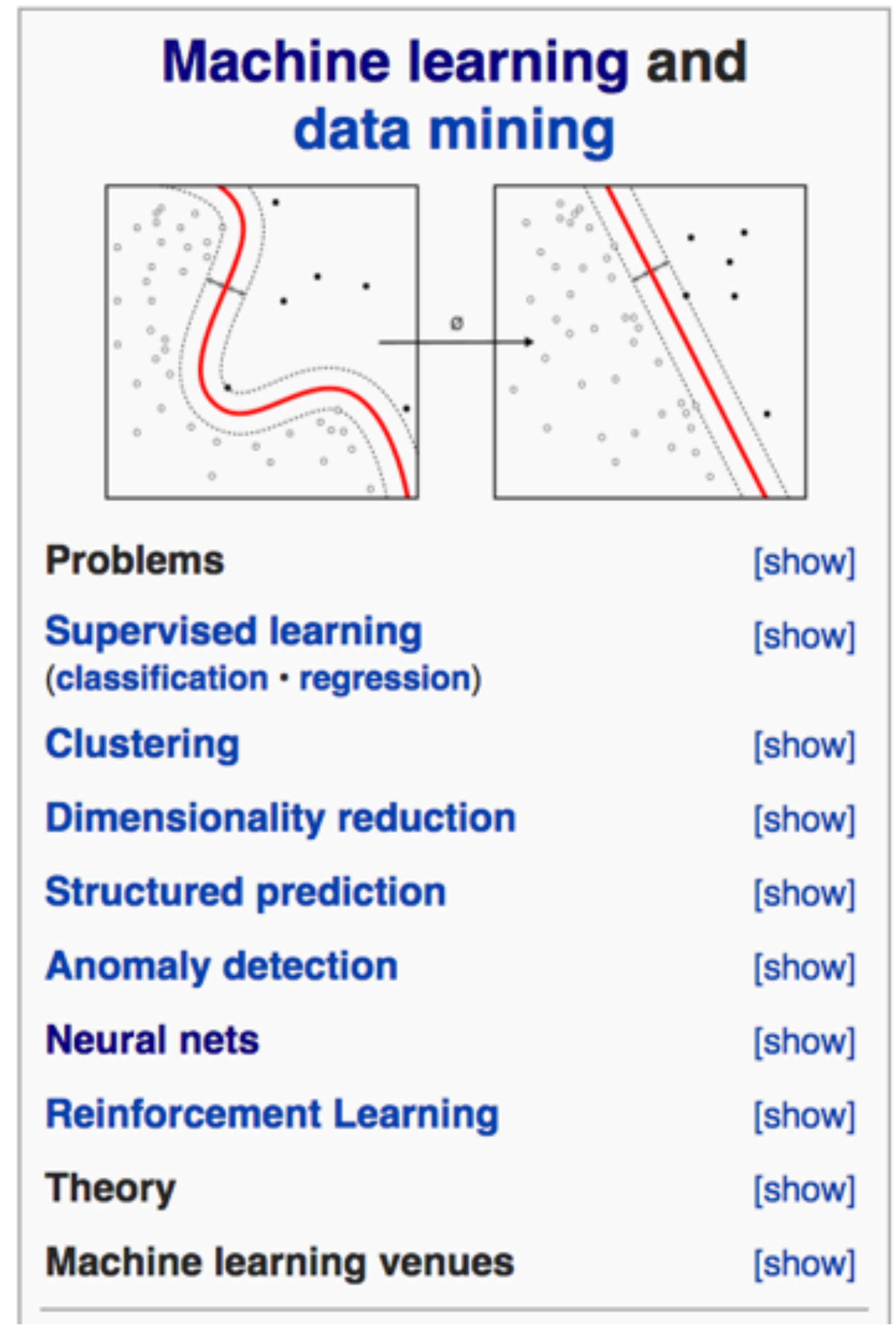
Some Descriptions

- Learning from data as opposed to explicitly programming result for every possible output

- Finding structure and patterns in data

- Learning from feedback or experience

- *Subset* of Artificial Intelligence



Wikipedia

Topics for today

- Supervised Learning
- Unsupervised Learning
- Probabilistic Graphical Models
- Practical ML (if time permits)

*Material based from courses/papers by Lorenzo Rosasco (MIT 9.520), Andrew Ng (Coursera, Intro to ML), Michael Jordan (Intro to Graphical Models). See Resources.

Supervised Learning

- Most common type of machine learning problem (e.g. ad click, news feed, detecting a disease, detecting cats)

- We are given both the input data and labels associated with it.

$$S = \{ (x_1, y_1) , (x_2, y_2) , (x_3, y_3) \ldots (x_n, y_n) \}$$

- Goal: Find function relating x's to corresponding y's

$$\mathcal{F} : \mathcal{X} \to \mathcal{Y}$$

  - Must work well for **new** x's (*generalization*)

Data Spaces

- Input space: X
- Output space: Y
  - Depending on the variable we are trying to predict:
    - Regression (y is continuous)
    - Classification (y is discrete)
- Assume (x, y) are **independently and identically sampled** from a fixed, unknown distribution

How good is our $\mathcal{F}$?

- Measures the error (or cost) of making an incorrect prediction

$$\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow [0, \inf)$$

- The expected loss (i.e. over entire data space) or **risk**

$$\mathcal{E}(f) = \mathbb{E}[\ell(y, f(x))] = \int p(x, y)\ell(y, f(x))dxdy$$

Risk Minimization

- The “best” function from X —> Y is one that works well over past as well as future data

- Problem: we don’t know the true distribution of data, can’t estimate risk accurately

- Instead, consider the empirical error

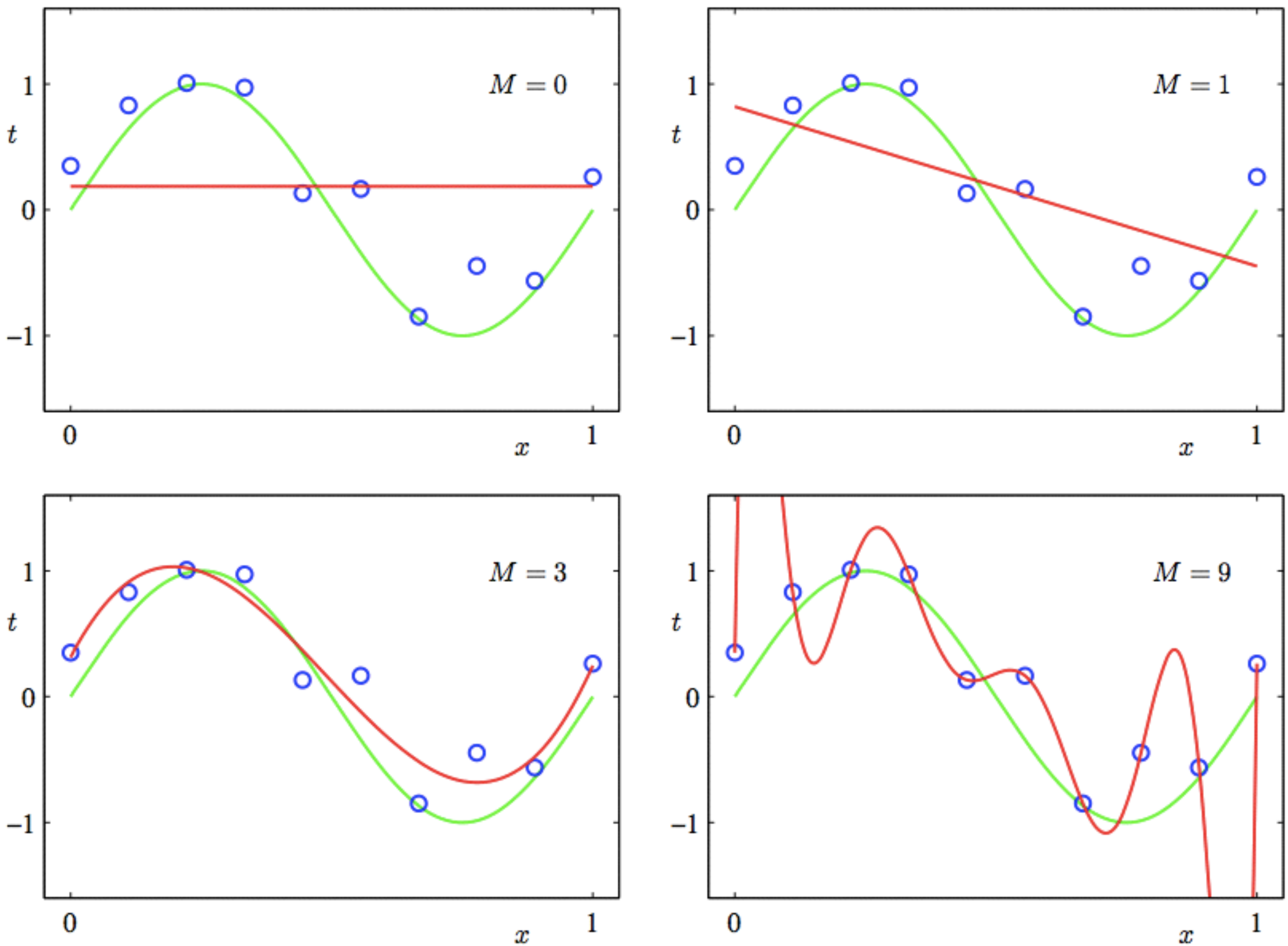
$$\hat{\mathcal{E}}(f) = \frac{1}{n}\sum_{i=1}^{n}\ell(y_i, f(x_i)),$$

Designing a Learning Algorithm

- Every learning algorithm has associated with it a "**hypothesis space**", H: a space of functions which will be explored to find a fit to data
  - E.g. linear functions, polynomials
- H should be **rich enough** to adequately capture the data, but highly complex H can lead to **overfitting**

Fitting, Generalization, Stability, Consistency

- Fitting: must adequately capture variation in the data
- Stability: must not change if the input changes a little
- Generalization: must work on previously unseen data
- Consistency: as more data is seen, the empirical risk should approach expected risk



Pattern Recognition and Machine Learning, Bishop

Regularization

- The most popular approach to preventing overfitting (others include early stopping)
- Penalizes model complexity and prefers simpler models
- E.g. Tikhonov regularization for linear models

$$\min_{w \in \mathbb{R}^d} \hat{\mathcal{E}}(f_w) + \lambda \|w\|^2$$

Linear Regression

$$\min_{w \in \mathbb{R}^d} \hat{\mathcal{E}}(f_w) + \lambda \|w\|^2, \quad \hat{\mathcal{E}}(f_w) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f_w(x_i))$$

- If y is continuous, and we choose squared loss, we get linear regression (regularized least squares)

$$\ell(y, f_w(x)) = (y - f_w(x))^2$$

- Can be solved analytically

Classification Techniques

$$\min_{w \in \mathbb{R}^d} \hat{\mathcal{E}}(f_w) + \lambda \|w\|^2, \quad \hat{\mathcal{E}}(f_w) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f_w(x_i))$$

- Different loss functions —> Different Learning Algorithms

- Ideally: 0/1 loss

- Logistic Loss: Logistic Regression

$$\ell(y, f_w(x)) = \log(1 + e^{-y f_w(x)})$$

Classification Techniques

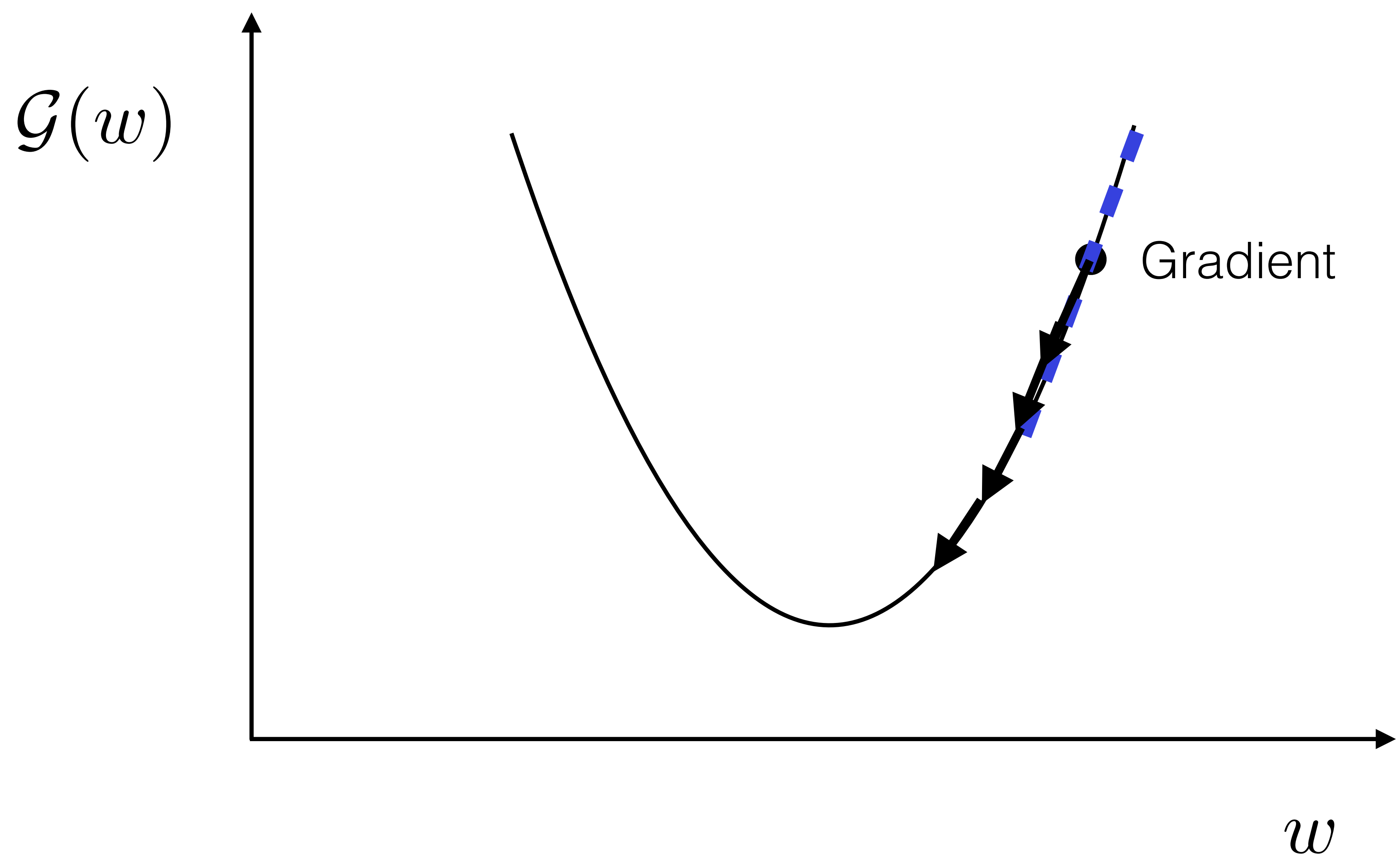
$$\min_{w \in \mathbb{R}^d} \hat{\mathcal{E}}(f_w) + \lambda \|w\|^2, \quad \hat{\mathcal{E}}(f_w) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f_w(x_i))$$

- Different loss functions —> Different Learning Algorithms
- Hinge Loss: SVM

$$\ell(y, f_w(x)) = |1 - y f_w(x)|_+$$

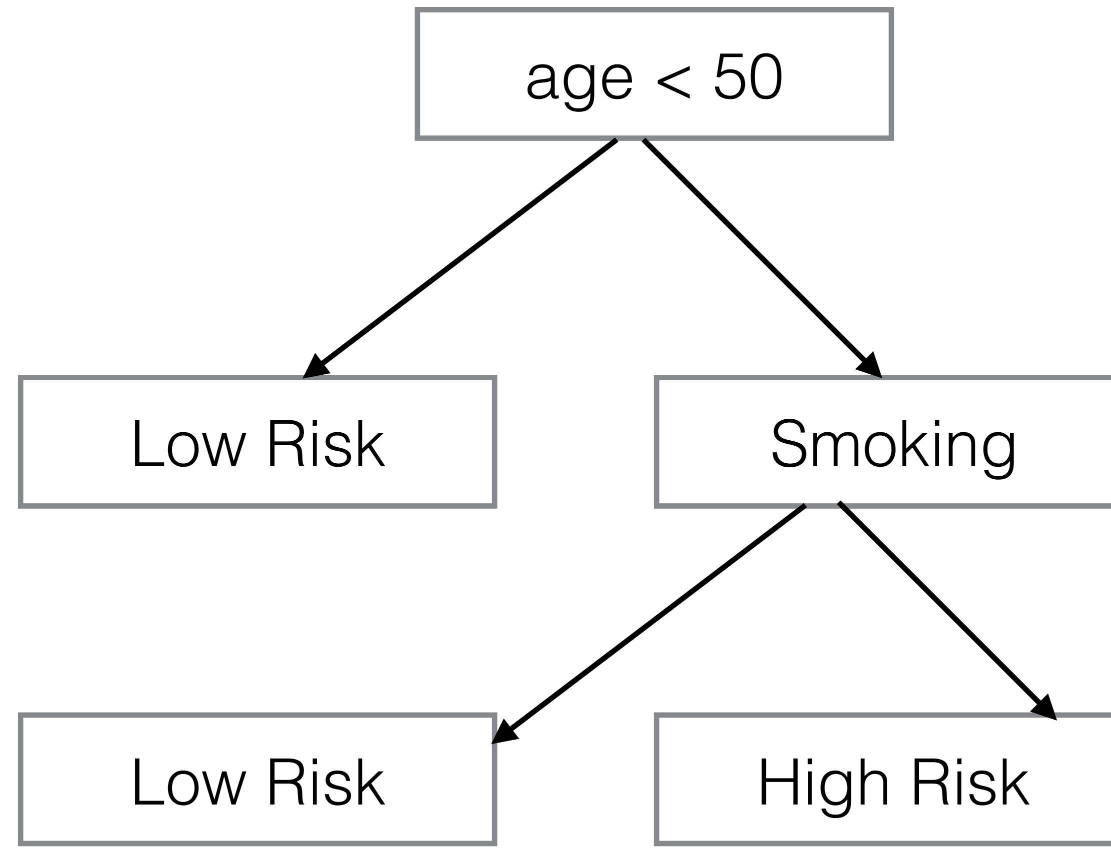
Gradient Descent

$$\min_{w \in \mathbb{R}^d} \hat{\mathcal{E}}(f_w) + \lambda \|w\|^2, \quad \hat{\mathcal{E}}(f_w) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f_w(x_i))$$



Decision Tree

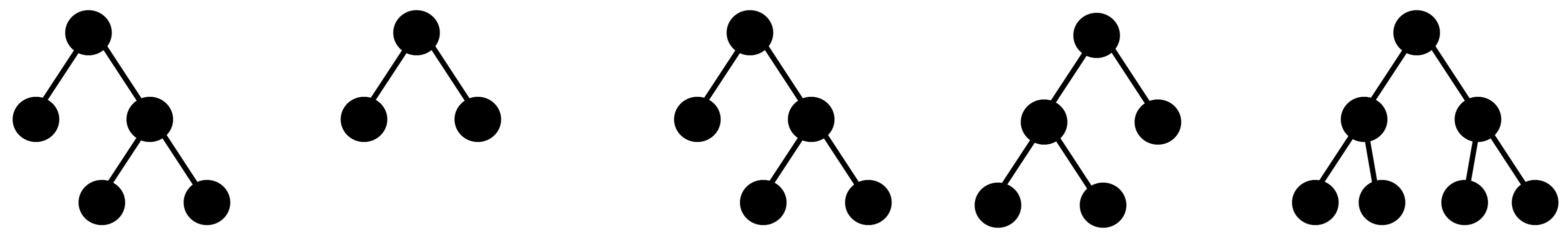
- Set of decision rules arranged in a hierarchy
- While training a decision tree, the objective is to arrive at leaves that are “pure”
- During prediction, starting at the root, a new example is sequentially tested against checks at each node. Final prediction is made at the leaf



Ensemble Methods

- Use more than one learner to make a prediction
- Often these learners are weak learners or learners learnt to make up for the errors of another learner
- Often work better than single models
- These learners can be of any type: linear models, trees, neural nets, SVMs etc.

Random Forest



- Ensemble of decision trees

- Tens to hundreds of decision trees learnt on the same data (using subsets of features and data) and predictions are made by voting

- One of the **most** popular classifiers

- Success attributed to ease of training and good performance on a variety of datasets

Random Forest Algorithm

- Every tree gets a random set of data points on which to train
- For every node in the tree, the candidate features used for splitting are chosen randomly
- Works because of non-correlated errors between individual trees
- Also look at gradient boosted trees

So far...

- Linear Regression
- Logistic Regression
- SVMs
- Decision Trees
- Random Forests

Hyperparameters

- Every learning algorithm has a set of parameters that are not learnt from data directly. They must be specified by the user
    - Number of trees
    - Depth of trees
    - Regularization parameters
- Choose via **cross validation**

Unsupervised Learning

What is it?

- The examples do not have labels, so there isn't a hypothesis we can fit to the data
- $S = \{ x1, x2 \ldots xn \}$
- Goal: find some structure in the dataset
- Examples: clustering, dimensionality reduction

Clustering

- Some uses:
  - Find segments of users
  - Cluster server data that is accessed together
  - Cluster genes by functions, interactions, lineage
- Algorithms: K-Means, Gaussian Mixture Models, Spectral clustering

k-means Algorithm

- Iterative Algorithm
- Steps
  - Choose centroids and assign points to centroids
  - Update the location of centroids
- Alternate between updating centroids and updating assignments

http://shabal.in/visuals/kmeans/2.html

Gaussian Mixture Models

- Clusters ~ sub-populations. Assume that data is drawn from a mixture of Gaussian distributions

- Instead of assigning a point to a **single** cluster, assign it to all the clusters but with different probabilities/weights

- Learn parameters of the distribution similar to k-means

Expectation Maximization

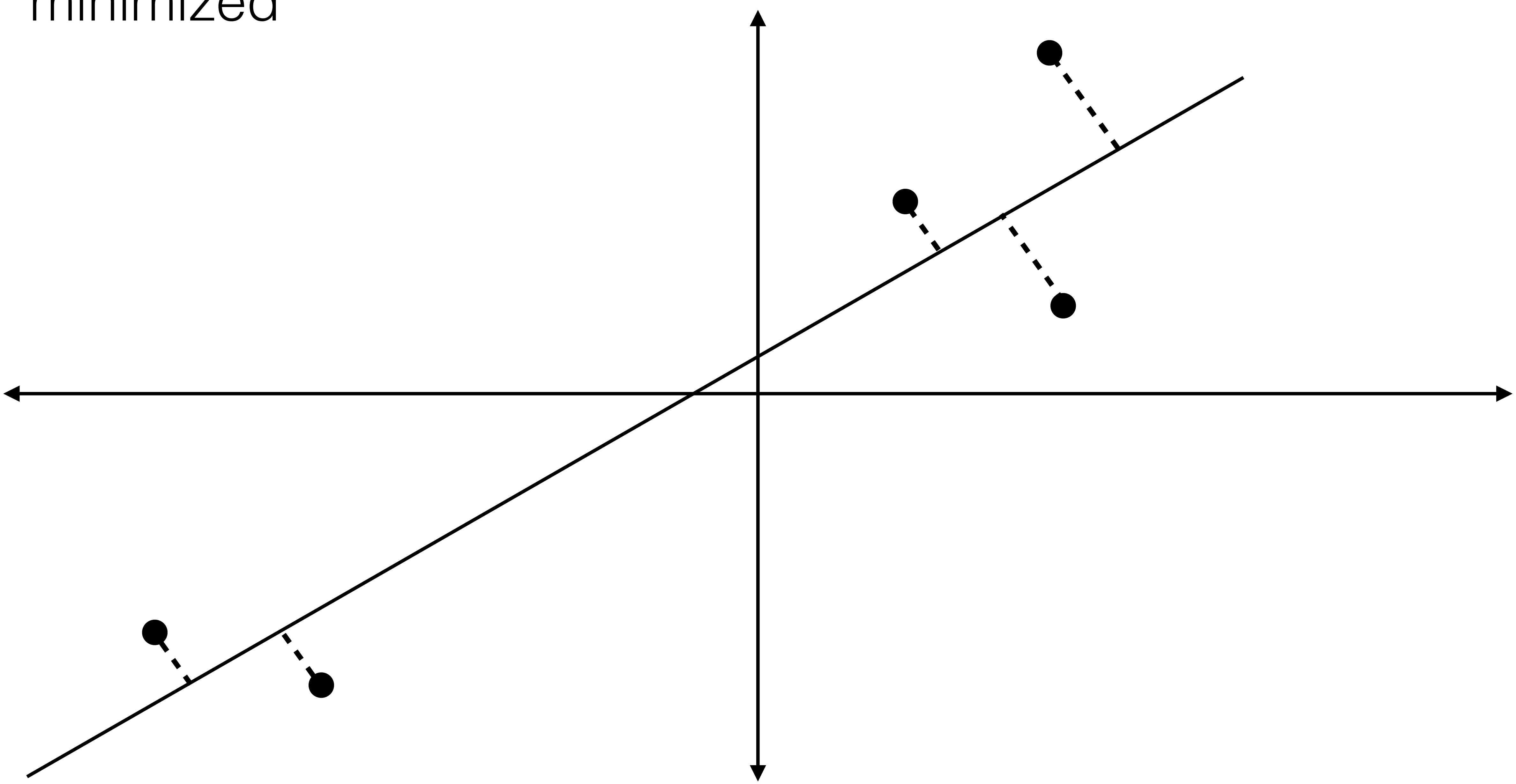
- Iterative Algorithm
- Steps
  - E-step: Compute membership weights for each point as belong to a mixture component
  - M-step: Compute the new parameter values (means, variances) for components
- Alternate between E and M steps

Dimensionality Reduction

- Data often have 1000s of dimensions, the goal is to reduce the number of dimensions to 100s
  - Data compression (storage, faster algorithms)
  - Data visualization
  - Find structure in the underlying data

Principal Component Analysis

Find a lower dimension surface such that projection error is minimized



PCA Algorithm Overview

- Compute covariance matrix: Σ
- Compute eigenvectors
  - [U, S, V] = svd(Σ)
  - Columns(U) = eigenvectors
  - Pick first *k* columns to project X into *k*-dimensional sub-space
- U_k^T * X gives projected data

Hyperparameters

- Unsupervised methods have hyperparameters too: number of clusters, dimensionality of sub-space
- However, unlike supervised methods, *no objective way* to determine which hyperparameter is better and therefore which model is better

Resources

- Coursera ML course
- MIT 6.867: (G) Machine Learning
- MIT 9.520: (G) Statistical Learning Theory
- CMU Intro to ML course

Probabilistic Graphical Models

“Graphical models are a marriage between probability theory and graph theory…

They provide a natural tool for dealing with two problems…uncertainty and complexity…”

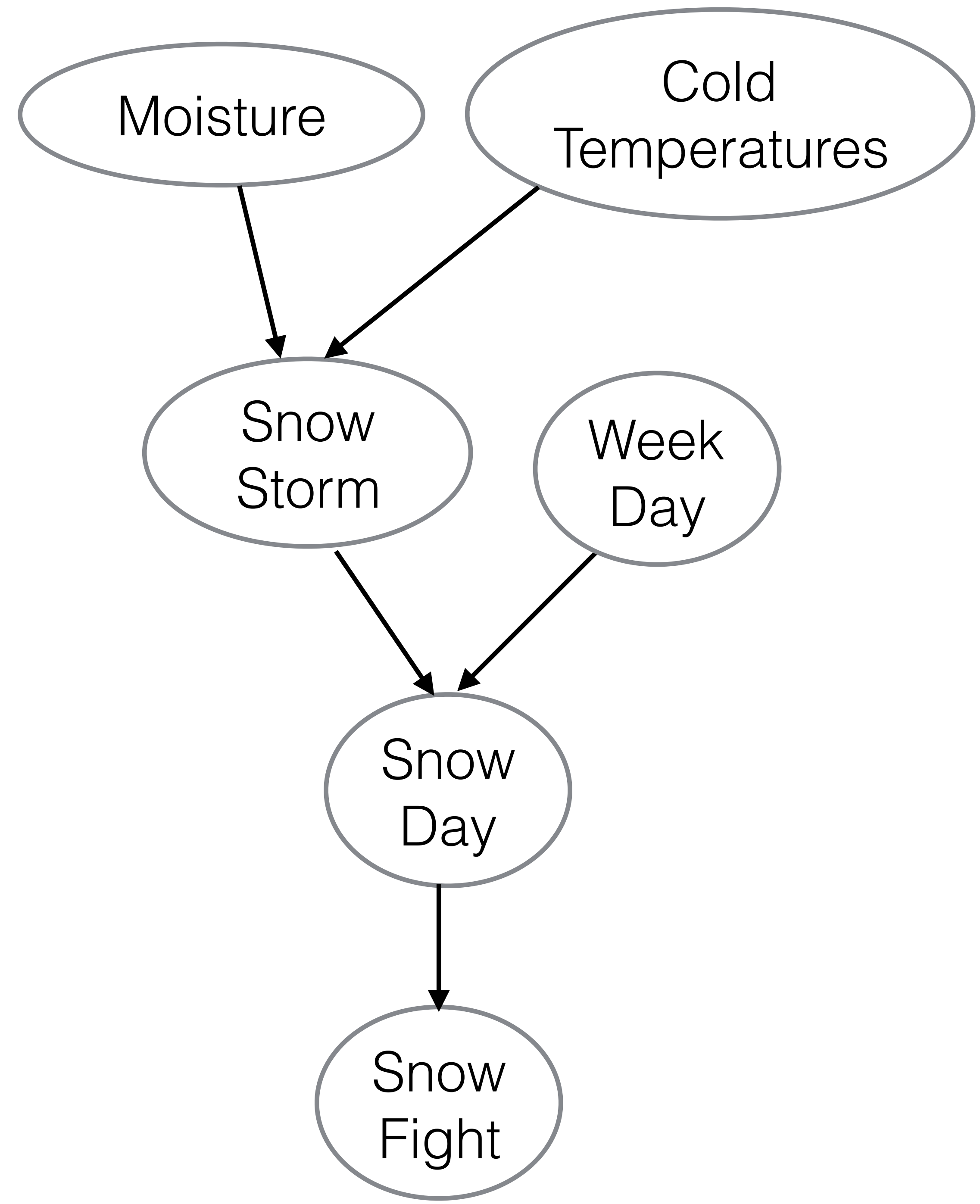
–Michael Jordan, 1998

Probability Terms

- Marginal: $p(x)$ — uncertainty in data
- Conditional: $p(y|x)$ — noise in outcome
- Independence: $p(x \wedge y) = p(x) * p(y)$
- Conditional Independence: $p(x \wedge y \mid z) = p(x \mid z) * p(y \mid z)$

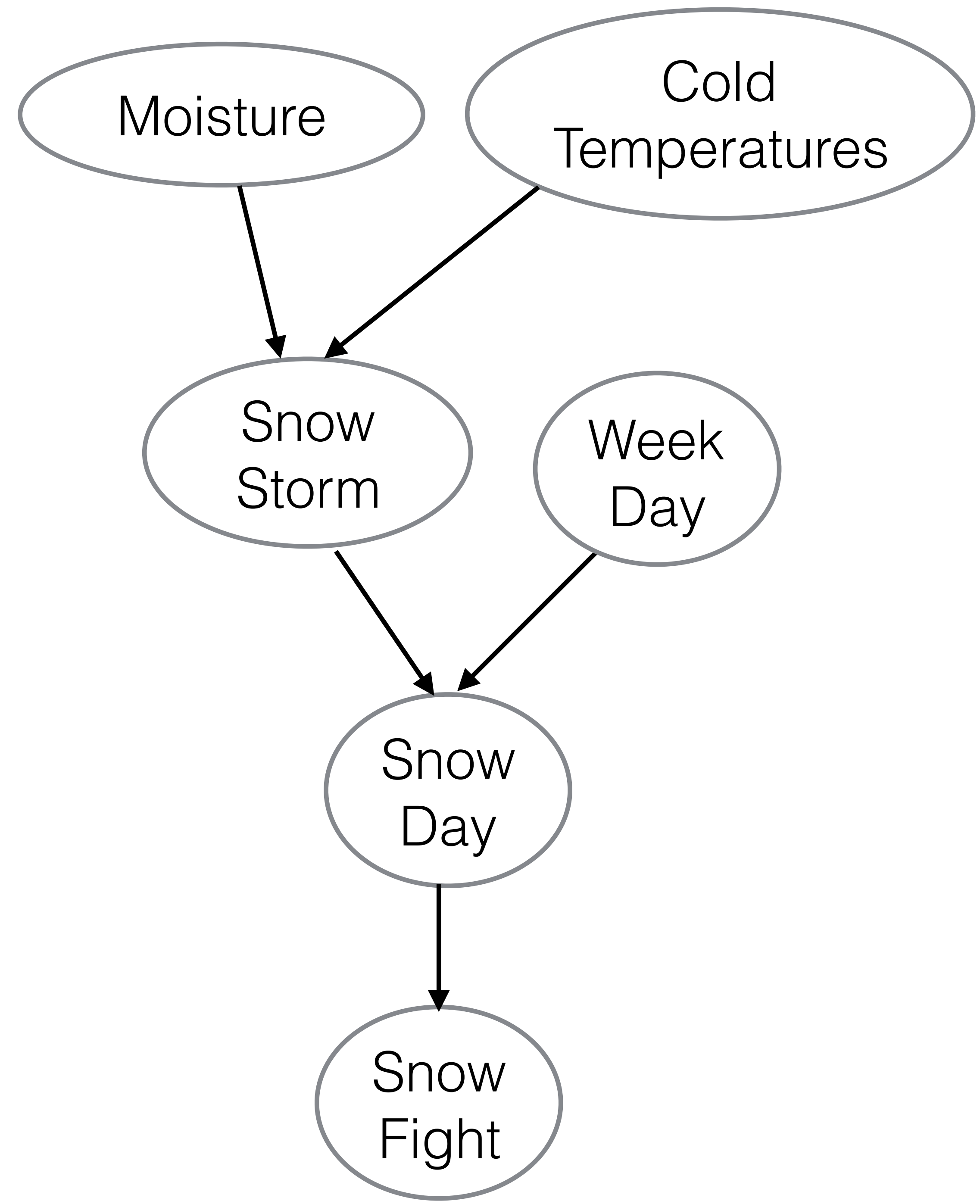
What are they?

- *Graphs*
  - Nodes: Random Variables
  - Edges: Dependences — conditional dependences — between variables
- Concise representations of complex probability distributions



What are they?

- If we knew nothing about relationships between variables, how many parameters would we have to learn?
    - $2^6 = 64$
- Given conditional independences, how many parameters?
    - 10



What are they?

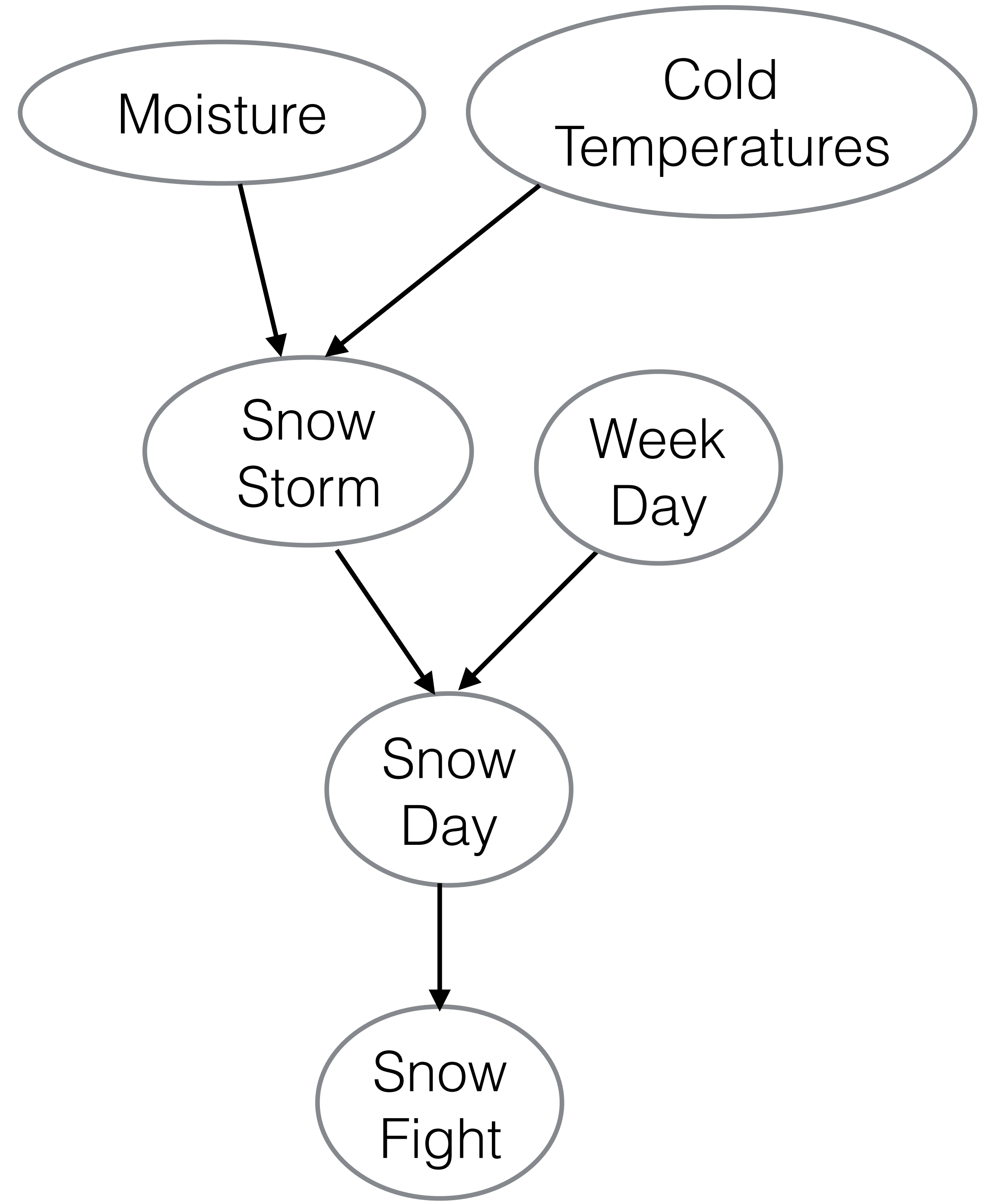
- Since edges represent conditional dependences (i.e. no edge = conditional independence), joint distribution is much *simplified*

- Joint probability distribution:

$$\mathcal{G}(\mathcal{V},\mathcal{E})$$

$$p(x_{\mathcal{V}}) = \prod_{v\in\mathcal{V}} k(x_v \,|\, x_{\pi_v}).$$

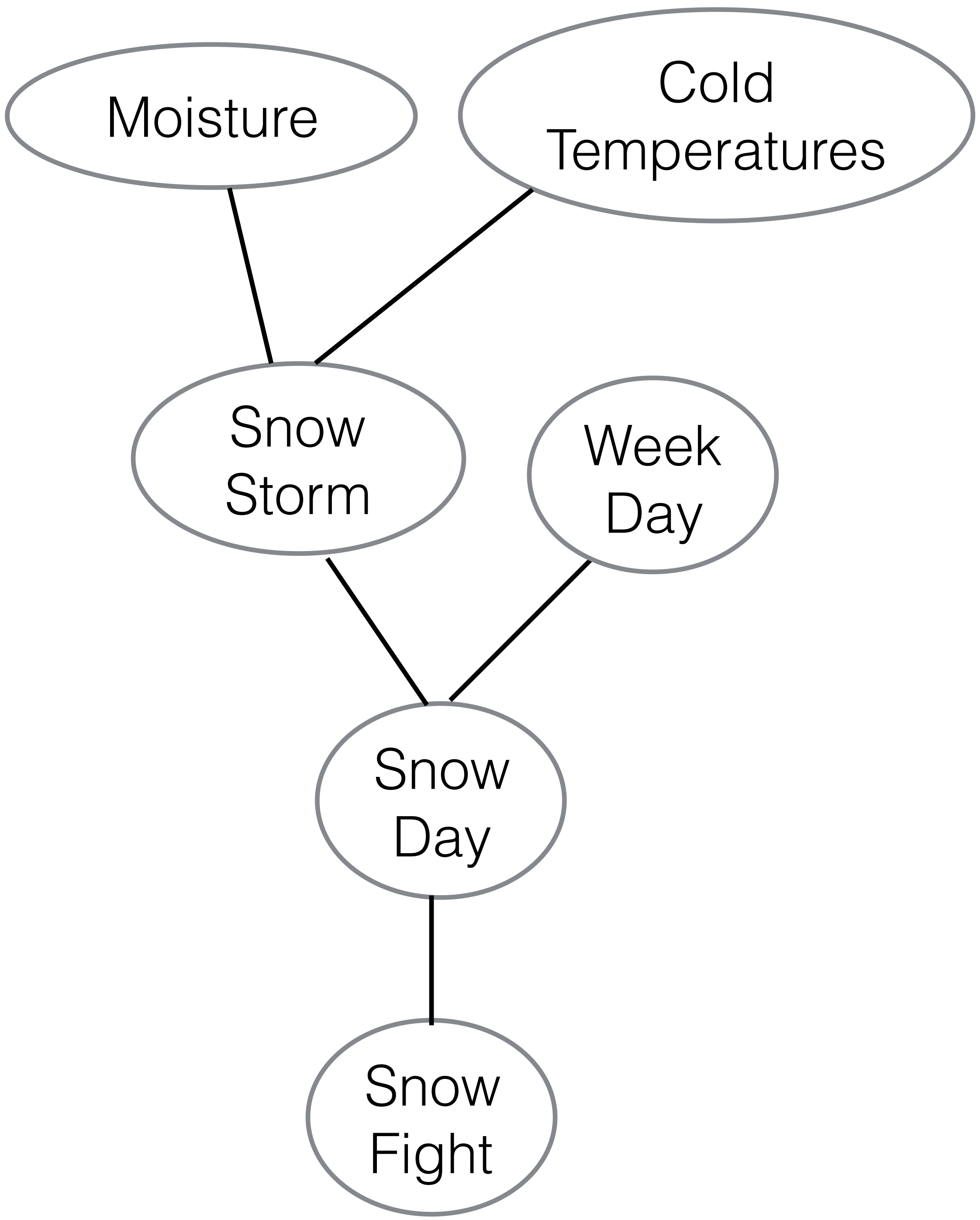
- PGMs represent a family of distributions



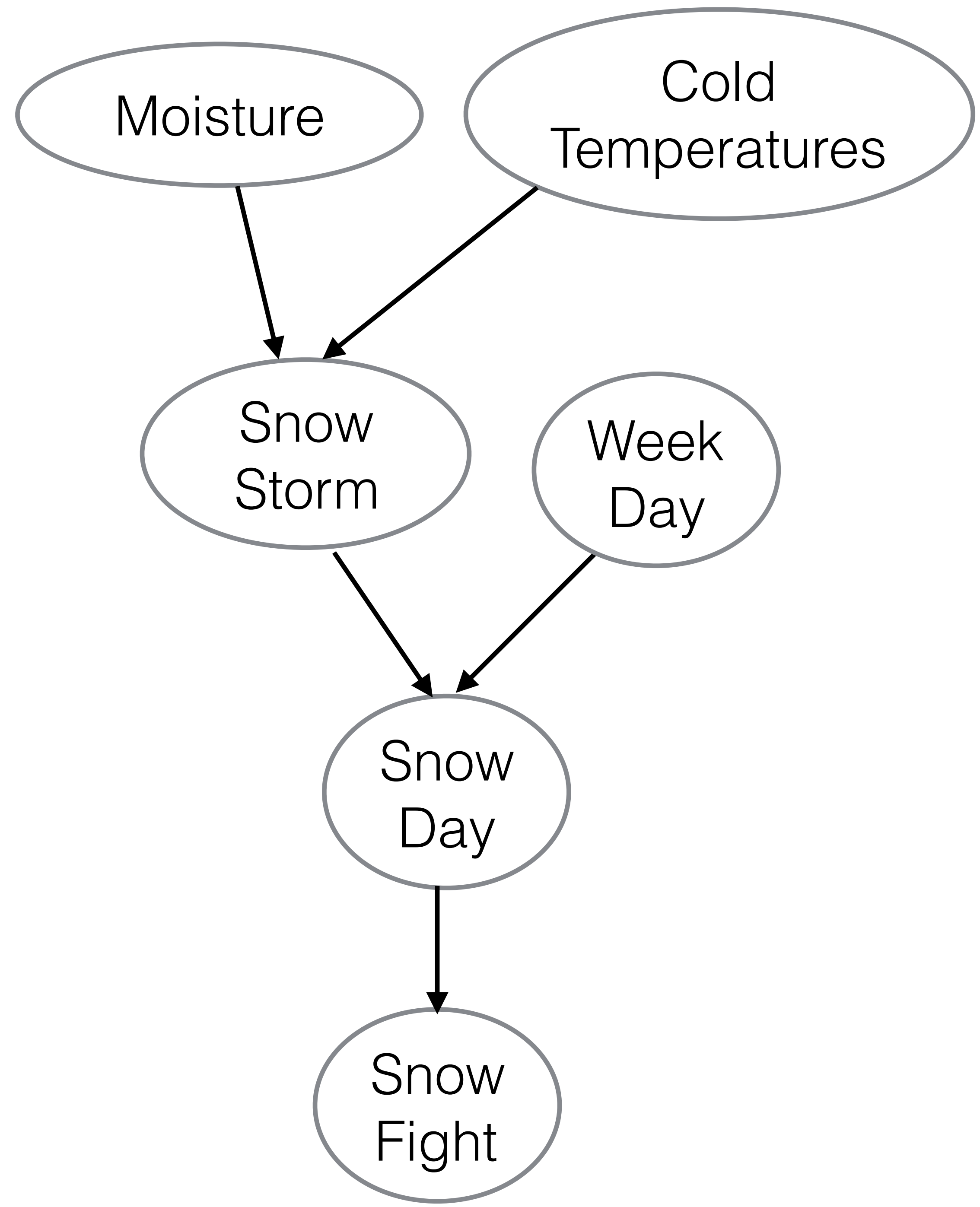
Why use PGMs?

- Real-world applications have many 1000s of variables that interact in complex ways
  - General framework to reason about them
  - Compact, intuitive structure representation
- Efficient reasoning
  - Exponential to ~polynomial number of parameters
  - Control computational cost

Two Kinds



Undirected Network

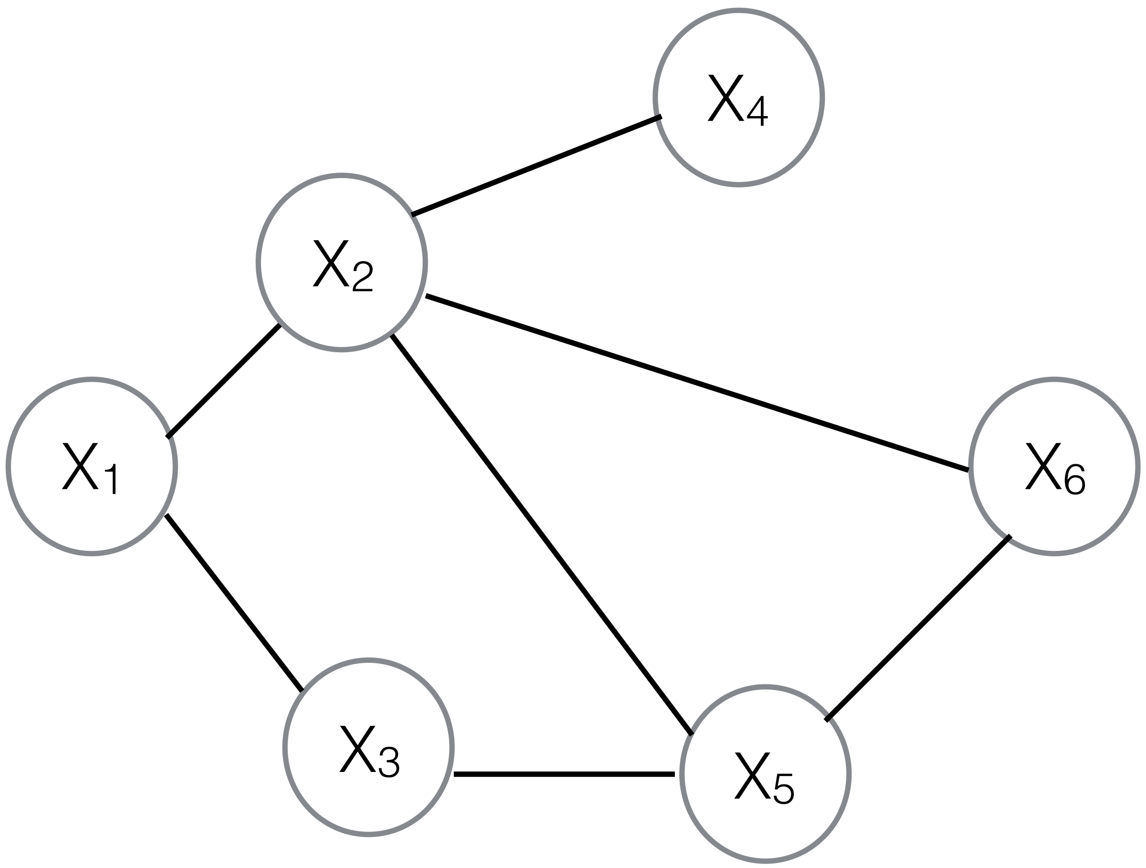


Bayesian Network

What can we do with PGMs?

- Answer queries about probabilities (*inference*): conditionals or marginals
  - E.g. if there was a snow fight, what were the chances that there had been a snow storm?
  - E.g. given that there were cold temperatures, what were the chances of getting a snow day?
- Inference algorithms: **exact**, sampling, variational

Undirected Graphs

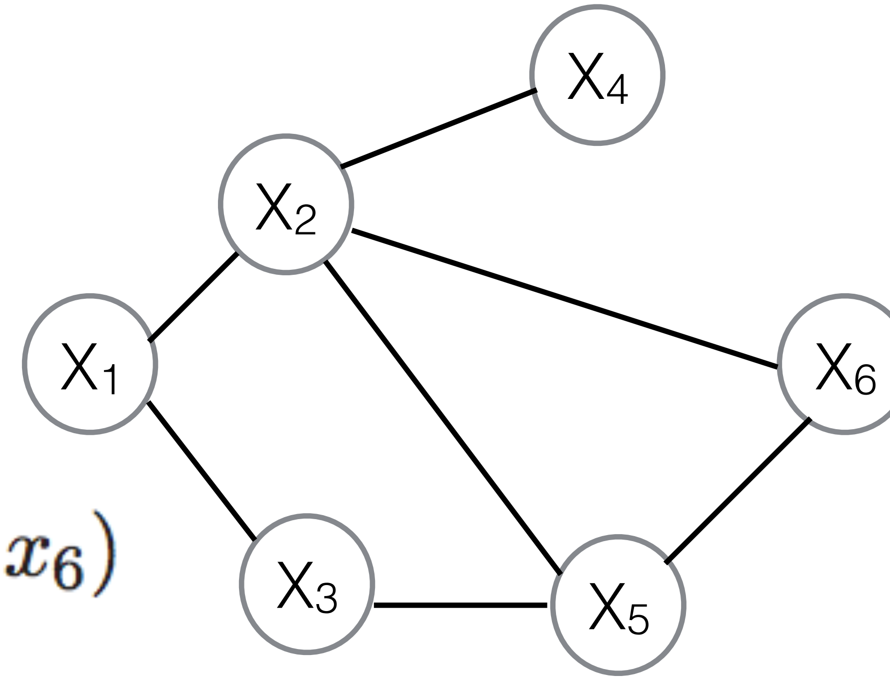


- Undirected Graph
- Joint Distribution

$$p(x_{\mathcal{V}}) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(x_C),$$

$$p(x_{\mathcal{V}}) \quad = $$
$$\frac{1}{Z}\psi(x_1, x_2)\psi(x_1, x_3)\psi(x_2, x_4)\psi(x_3, x_5)\psi(x_2, x_5, x_6)$$

Variable Elimination



$$p(x_{\mathcal{V}}) \quad = \quad \frac{1}{Z}\psi(x_1, x_2)\psi(x_1, x_3)\psi(x_2, x_4)\psi(x_3, x_5)\psi(x_2, x_5, x_6)$$

$$p(x_1) = \sum_{x_2}\sum_{x_3}\sum_{x_4}\sum_{x_5}\sum_{x_6}\frac{1}{Z}\psi(x_1, x_2)\psi(x_1, x_3)\psi(x_2, x_4)\psi(x_3, x_5)\psi(x_2, x_5, x_6).$$

$$= \quad \frac{1}{Z}\sum_{x_2}\psi(x_1, x_2)\sum_{x_3}\psi(x_1, x_3)\sum_{x_4}\psi(x_2, x_4)\sum_{x_5}\psi(x_3, x_5)\sum_{x_6}\psi(x_2, x_5, x_6)$$

Sum-Product Algorithm

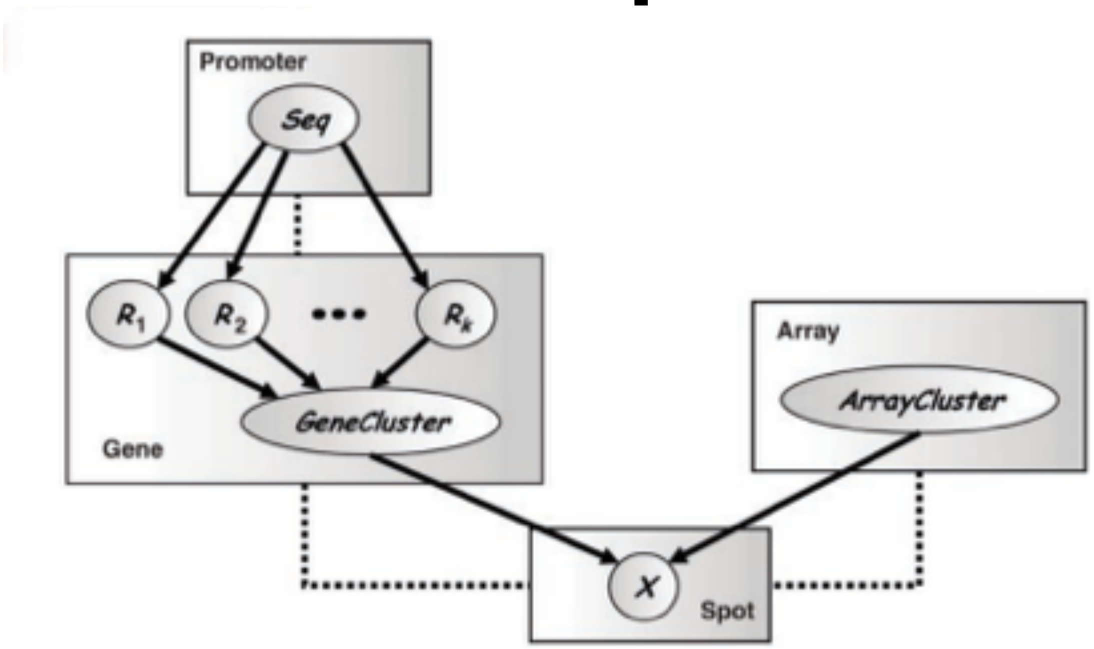
- Variable elimination is for a **single** marginal. What we we want to compute them all?

- For graphs that are *trees*, variable elimination intermediates have this form:

$$m_{ji}(x_i) = \sum_{x_j} \left(\psi(x_j)\psi(x_i, x_j) \prod_{k \in \mathcal{N}(j) \backslash i} m_{kj}(x_j) \right),$$

- If intermediate results of elimination are cached, they can be re-used (via message passing)

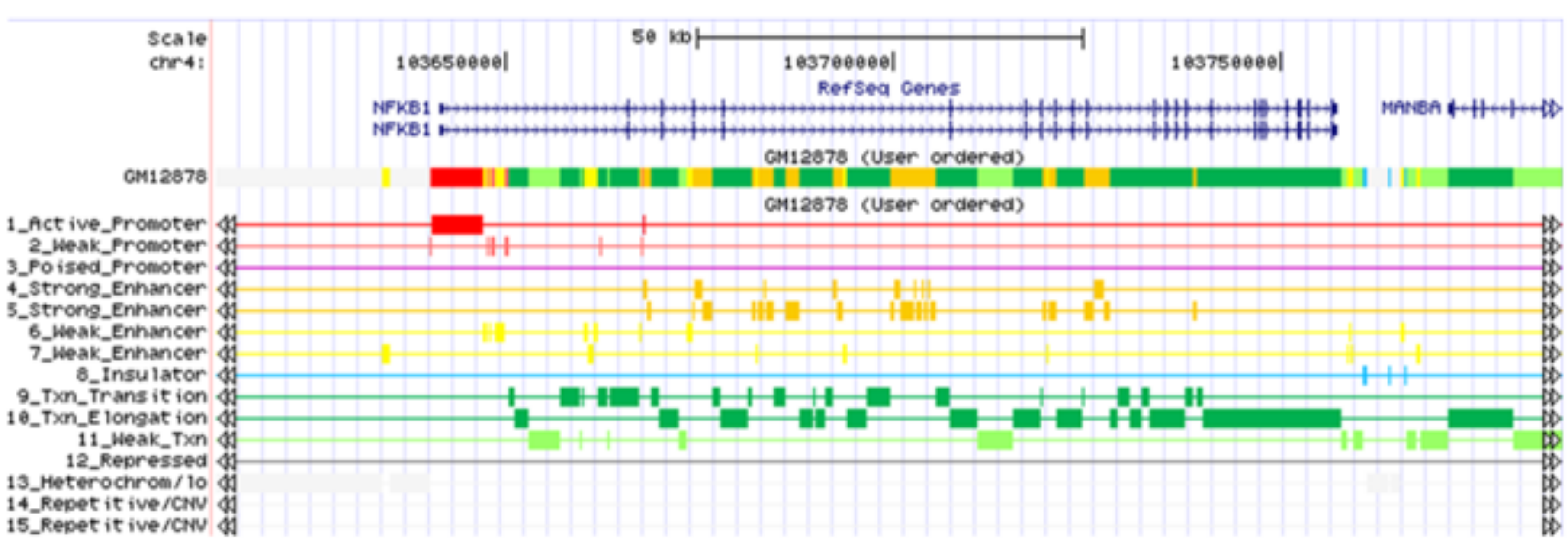
Examples & Applications



Inferring Cellular Networks (BN)

"Arts"	"Budgets"	"Children"	"Education"
NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI

Topic Modeling (LDA)



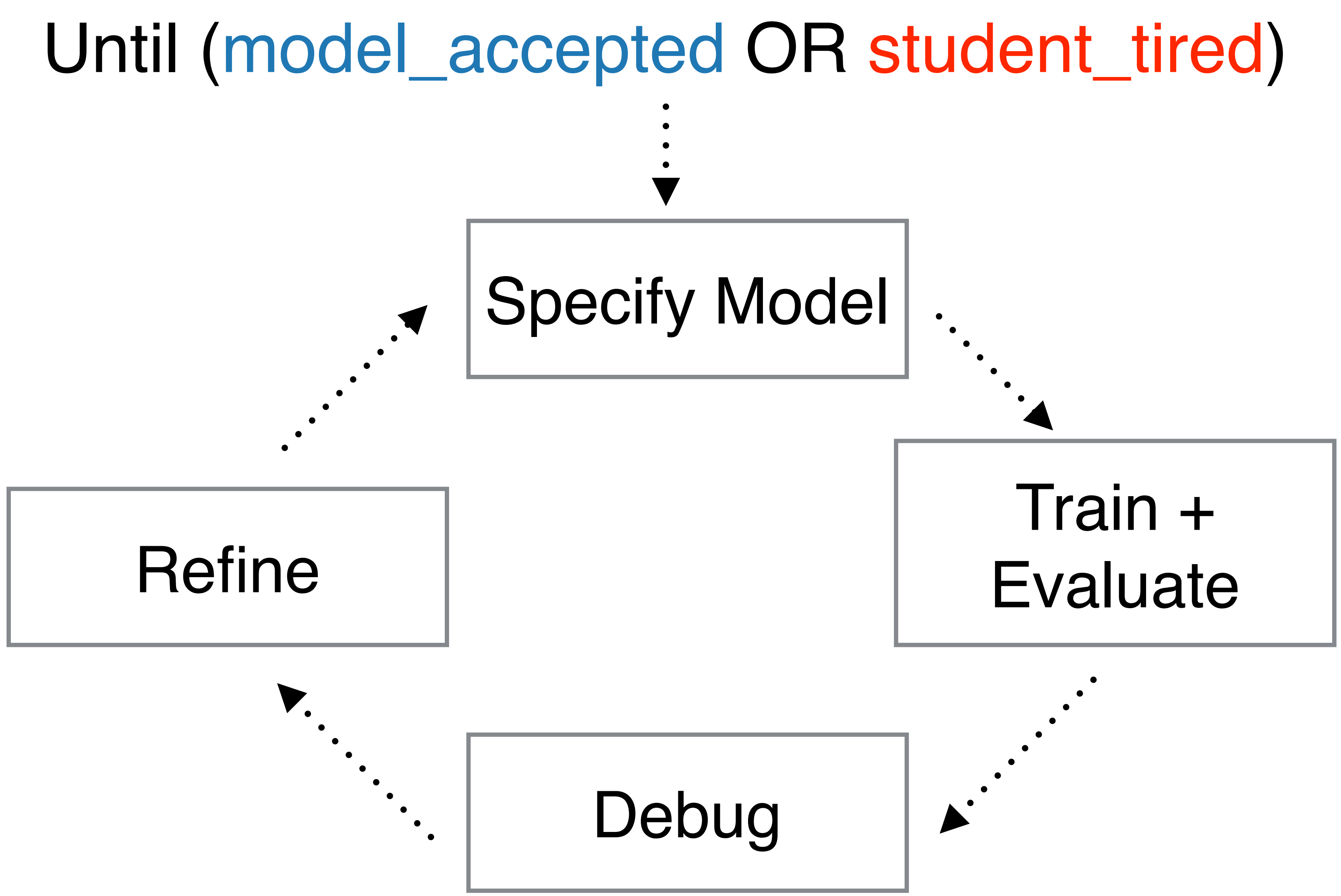
Annotating the genome (HMM)

Resources: graphical Models

- MIT 6.867: (G) Machine Learning
- Article by Mike Jordan
- Coursera course on PGMs
- David Blei's Columbia course

Practical ML

What does it take to build a good model?



Systems & Tools

- Traditional ML:
  - Python (scikit-learn, gensim)
  - R (glmnet, gbms, …)
  - Matlab, Julia, Octave …
- Deep Learning:
  - Tensorflow, Torch, MXNet
- Large-scale: Spark, H20

Some Tips and Tricks

- Look at your data!
- Try the simple stuff first
- For most algorithms, normalize and scale your data
- Make your algorithm work on a small dataset first (where you know what the right answer should be)
- Always use train, test and validation set (or CV)
- Regularize your models
- Sweep over hyperparameters

Wrap-up

- Hope you enjoyed Session I!
- Tomorrow: Sampling and Inference (Maggie)
- Feedback:

  https://goo.gl/forms/sUURVW4lcaJRVmE93

- Contact: mvartak@csail.mit.edu | @DataCereal