MLSS 2017 Lecture Notes

Bernard Schölkopf - What is ML?

- Leibniz - thought experiments about understanding laws for data
- what does it mean to generalize
  - deduction problem
- statistical learning theory - demarcation problem

Cybernetics

- 1940s, Cybernetics Norbert Wiener, Cynernetics or control and communication in the animal and the machine
  - study of control and information processing rather than energy processing in machines and animals
- macy conferences 1946-1953
- project cybersyn at allende government (chile, 1971 - 1973)
- mcculloch-pittts, formal neurons can emulate universal turing machines
- hebbs - formal neurons
- rosenblatt - the perceptron, a probabilistic model for information storage
  - first perceptron, 6 input pixels -> modifiable weights from error propagation
  - perhaps first example of learning weights
- perceptron convergence theorem (1962)
- limitations - xor problems
  - excessive learning times
- minsky and papert (1969) - perceptrons
- adam newey - CS as a principled discipline for inquiry
- symbolic AI
  - process of manipulating discrete symbols
    - john mccarthy, allen newey, herb simon, marvin minsky
- symbolic ai did lead to the birth of CS
  - led to development of high-level programming langauges (IPL and lisp)
- the end of perceptrons
- rosenblatt continued, but passed away in 1971
- defeat of neural networks ligeitmized symbolic AI
- neural network research continued at the fringe
  - kohonen, hinton, amari, grossberg
  - probabilistic reasonign in intelligent systems
- in parallel, pattern recognition studying statistical learning theory development (international of control science at russia)
  - vapnik and chervonenkis (1968 - 1982)
- expert systems / knowledge representations were made probabilistic

  - judea pearl (1988)
  - gave birth to bayesian networks - probabilistic graphical models
  - how to connect probabilities
- backpropagation in 1980s
- perceptrons 2nd edition
  - backprop simply form of calculating gradients
  - leads to solutions every time
  - just hill climbing
- solomonoff (1950s) - probabilistic AI
- vapnik - generalized portrait algorithm (mid 60s in his thesis)
  - some kind of optimal marginal for perceptron rule/algorithm
  - notion of positive definitive kernel (1904, hilbert)
- CS is a discipline centered around programs
- program can be written iff we have a pricise model of what it shoudl do
- human comes up with models (induction), computer does the rest (deduction)
- nick bostrom - superintelligence
- Vapnik paper, generalization of Glivenko-Cantelli
  - dudley: "shocking"

Shai Ben-David - Understanding Machine Learning, A Theory Perspective

- key ingredients
- data distribution D
- $f : x \to y$
- minimize probability of $p_h(H(x) \neq f(x))$
- natural measure: empirical error of h $\#S = |i : h(x_i) \neq f(x_i)|$
- pigeon superstition (Skinner 1948)
  - aim to replicate human behavior in animals
  - pigeon experiment - replicate superstition
- no free lunch
  - no learning is possible without prior knowledge
- PAC Learnability - if there is a function $m_h : (0,1)^2 \to N$ and a learnign algoirthm A, such that for every distribution D over X, ever ϵ, $\delta > 0$, and every f in H, for samples S of size $m > m_H(\epsilon, \delta)$ generated by D and labeled by f,
  - $Pr[L_D((A(S)) > \epsilon] < \delta$
- independent of unknown distribution D
  - in statistics, often make assumptions on distribution D first
  - in ML, we are using arbitrary distribution D, bound still holds as it is
- the rule depends on the classes
- relaxing the realizability assumption
  - wish to model scenarios in which the learner does not have prior knowledge of a class to which the true classifer belongs

- furthermore, often the labels are not determined by the instance attributes (not deterministic)
- general loss: $\ell : H \times Z \to \mathbb{R}$
  - loss tells you how bad the model is given a point
  - $L_P(H) = \mathbb{E}_{X\ P}(\ell(h, z))$
  - general loss tells you expected loss under given sample point
- Agnostic PAC Learner
  - H is agnostic PAC lernable if there is a function $m_H : (0, 1)^2 \to N$ and a learning algorithm A, such that for every distribution P over $X \times Y$ and every $\epsilon, \delta > 0$, for samples S of size $m > m_H(\epsilon, \delta)$ generated by P,
    - $Pr[L_P(A(S)) > Inf_[h \in H]L_P(h) + \epsilon] < \delta$
  - instead of making absolute statement that is guaranteed only under certain assumptions (like realizability), making a weaker, relative guarantee that is not much worse than the best in the class, and is guaranteed to always hold
- **uniform convergence property** :
- If H is finite, then it has the uniform convergence property
- any finite H, is agnostically PAC-learnable.
- *proof*: hoeffding inequality implies uniform convergence property for single h's and then teh union bound handles the full class
- can we not restrict to a class H, i.e., use a universal learner
  - no-free lunch theorem says no universal learner
  - Let A be any learnign algorithm over some domain set X
  - Let m be < |X| / 2 then there is a distribution P over $X \times 0, 1 and f : X \to 0, 1$ such that
    1. $L_P(f) = 0$
    2. For P-samples S of size m with probability $> 1/7$ $L_P(A(S)) > 1/8$

Distinguishing between learnable and not learnable

- some infinite classes are learnable
  - eg:
    1. initial segments of the real line
    2. class of singletons over any domain set
- a combinatorial characterization of PAC learnable classes
- a class H shatters a domain subset A if for every susbet B of A there is some h_B in H so that for all x in A $h_B(x) = 1$ if and only fi x is in B.
- VC dimension:
  - largest set such that H shatters A
  - $VC_{dim_H} = \sup |A| : H shatters A$
- The fundamental theorem: the following statemetns are equivalent
  1. H has the uniform convergence property
  2. ERM is an agonstic PAC learner for H
  3. H is agnostic PAC learnable

4. H is PAC learnable
5. VC_{dim_H} is finite

Part III

Quantitative version of the fundamental theorem

- H has **uniform convergence property** with $C_1(d + \log(1/\delta))\epsilon^2 < m_H^{uc}(\epsilon, \delta) < C_2(d + \log(1/\delta))/\epsilon^2$
- H is **agnostic PAC learnable** with $C_1(d + \log(1/\delta))\epsilon^2 < m_H(\epsilon, \delta) < C_2(d + \log(1/\delta))/\epsilon^2$
- H is **PAC learnable** (*realizable case*) with $C_1(d + \log(1/\delta))\epsilon < m_H(\epsilon, \delta) < C_2(d + \log(1/\delta))/\epsilon$

Example Neural networks, VC dimension is about $|E| \times \log|E|$, where E are number of edges/weights. Rearranging, $C_1(d + \log(1/\delta))m_H^{uc}(\epsilon, \delta) < \epsilon^2 < C_2(d + \log(1/\delta))/m_H^{uc}(\epsilon, \delta)$, or roughly d/m, where d is sample-size and m is VC-dimension, or edges.

So if edges is order of magnitude the same or larger than training size, ϵ will be ≥ 1, no guarantees.

Hence, for guarantees need sample sizes with training examples that are order of magnitude larger than edges. This is worst-case theory.

Two missing components

- classes are learnable only if they have finite VC dimensions: this might be too restricted
  - fixing the class with finite VC dimension is sometimes too limited
- computational complexity
  - ERM's computational complexity in many cases is NP hard.

Relaxing the notion of learnability – non-uniform learnability

- A class H is non-uniformly learnable if there is a function $m_H : H_x(0,1)^2 \to \mathbb{N}$ and a learning algorithm A, such that for every distribution P over $X \times Y$ and every $\epsilon, \delta > 0$, for every h in H for samples S of size $m > m_H(h, \epsilon, \delta)$ generated i.i.d. by P,

$$Pr[L_P(A(S)) > L_p(H) + \epsilon] < \delta$$

- no longer uniform in m_H, different number of necessary samples depending on the h.

- If H shatters an infinite set, then it is not even non-uniform learnable

- in particular, the class of ALL functiosn over any finite domain is not non-uniform learnable.
- it shatters $\mathbb{N}$

Three Missing Topics

1. Safety - our ERM results relied on statistical guarantees. Some use-cases can not tolerate ϵ errors. Here we can'
2. Fairness - examining predictions based on past data.
3. Interpretability - understanding an interpreting model results.

Bernard Schölkopf - Causality

- Storks delivers our babies
- Reichenbach - Common cause principle
  - book: the direction of time
  1. if X and Y are statistically dependent, then there exists Z causally influencing both of them
  2. Z screens X and Y from each other, (given Z, X and Y become independent)
- SCM - structural causal model
- $A := N_A$
- $T := f_T(A, N_T)$
  - where N_T independent of N_A
- allows identification of the causal graph under suitable restrictions on the functional form of f_T.
- Structural causal model (Pearl et. al)
- directed acyclical graph with vertices
- semantics: vertices = observables, arrows = direct causations
- $X_i := f_i(PA_i, U_i)$ with indepdent RV $U_1, \ldots, U_n$, where U stands for unexplained random variabels
  - also called a nonlinear structural equation model

D. Janzing - Causality

- Causal structure formalized by DAG G with random variables $X_1, \ldots, X_n$ as nodes
- Causal markov condition states that the density $p(x_1, \ldots, x_n)$ then factorizes into

$$p(x_1, \ldots, x_n) = \prod_j^n = p(x_j | pa_j)$$

- Pearl's do-notation, distribution of Y given that X is set to x:
  - $p(Y|do\,X = x)$ or $p(Y|do\,x)$
- Computing $p(X_1, \ldots, X_n|do\,x_i)$ from $p(X_1, \ldots, X_n)$ and G
  - start with causal factorization
  - replace conditionals for intervention variables by Kronecker delta
    * i.e., replace $p(X_i|PA_i)$ with δ_{X_i,x_i}

Inferring the DAG

- Key postulate: causal markov condition
- Essential concept: d-separation
- Describing conditional independencies using paths and blocks along paths
  - d-separation provides the descriptive notion of conditional independence
- Berkson's paradox (1946): independence variables, but correlated through confounding
- (Reichenbach 1956): asymmetry under inverting arrows

Ben Schölkopf - Causality Part II

Max Welling - Marrying Graphical Models & Deep Learning

- Main actor of today's story:

$$\mathbb{E}_{Q(V)}[\log P(X|V)] - KL[Q(V)||P(V)]$$

- $P(V)$ is complexity penalty

ML as Computational Statistics

- There are perspectives from statistics that you cannot get from an optimization perspective
- Maximize log-likelihood

$$\max_{\Theta} \log P(X_1, \ldots, X_n|\Theta)$$

for unsupervised.

For supervised:

$$\max_{\Theta} \log P(Y_1, \ldots, Y_n|X_1, \ldots, X_n|\Theta)$$

and minimization of loss:

$$\min_{\Theta} \sum_i Loss(Y_i, \hat{Y}(X_i, \Theta))$$

Bias-Variance Decomposition

- Examining $Y = f(x) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon)$
- $Err(x) = \mathbb{E}[(Y - \hat{f}(x))^2]$
- $Err(x) = (\mathbb{E}[\hat{f}(x)] - f(x))^2 + \mathbb{E}[(\hat{(}f)(x) - \mathbb{E}[\hat{(}f)(x)])^2] + \sigma_\epsilon^2$
- First term is variance, second is bias, third is irreducible error

Graphical Models

- Concisely represent conditional independence relations between variables
- One-to-one correspondence between the dependencies implied by the graph and the probabilistic model
- Can do calculations just by marginalizing out the graph

Bayes Ball Algorithm

- mechanically/mechnanistically if variables are marginally independent
- An undirected path is active if a Bayes ball traveligna long it never encounters the "stop" symbol
- If there are no active paths from X to Y when $Z_1, \ldots, Z_k$ are shared, then $X \perp Y$.

Markov Random Fields

- Probability distribution as maximal clique:

$$P(X) = \frac{\prod_c \Phi_c(X_c)}{Z}$$

Latent Variable Models

- Introduction latent (unobserved) variables (perhaps confounders, if taking a causal perspective) will dramatically increase the capacity of the model:

$$P(X) = \sum_Z P(X|Z)P(Z)$$

- Fundamental degrees of freedom of what you're trying to model
- Problem: $P(Z|X)$ is intractable for most nontrivial models
  - for learning/inference, you need $P(Z|X)$ (unobserved nodes given observed nodes), so this can be tricky

Mainstream Ways of Handling Intractable Inference

Variational Inference

- Want to estimate some complex probability distribution p
- Restrict yourself to a family of simple distributions Q
- **Advantages**:
  - deterministic
  - easy to assess convergence
- **Disadvantages**:
  - biased
    * Never get actual P, since you're biased
  - Local minima

Sampling - MCMC

- **Advantages**:
  - Unbiased
- **Disadvantages**:
  - Stochastic (sample error)
    * suffering from variance, not bias this time
  - hard to mix between modes
  - Hard to assess convergence

Independence Samplers and MCMC

- Genearting independent samples: sample from g and suppress samples with low $p(\theta|X)$, e.g., rejection sampling, or importance sampling
  - does not scale to high dimensions
  - too much variance
- MCMC
  - make steps by perturbing previous sample
  - probability of visiting a state is equal to $P(\theta|X)$
- Sampling 101: Metropolis-Hastings
  - propose new step with Gaussian movements
  - satisfies detailed balance: is probability flow in either transition balanced
  - is it easy to come back to the current state?
  - is the new state more probable
  - Burn-in is unnecessarily slow

- This algorithm is $\mathcal{O}(N)$

Variational Inference

- Choose tractable family of distributions
- Minimize $Q : KL[Q(Z|X)||P(Z||X)]$
- Equivalent to maximize of Φ:

$$\sum_{Z} Q(Z|X,\Phi)(\log P(X|Z,\Theta)|P(Z) - \log(Q(Z|X,\Phi)))$$

- in learning, maximize the probability of observed data given parameters
- KL provides notion of bound $B : KL[Q(Z|X)||P(Z||X)]$
- E-M:
  1. E-Step: $\arg\max_{\Phi} B(\Theta, \Phi)$ [variational infernece]
  2. M-step: $\arg\max_{\Theta} B(\Theta, |\Phi)$ [approximate learning]
- when no gap, then EM, otherwise variational inference
- coordinate ascend on bound

Amortized Inference

- Encoder: $q_{\phi}(z|x|)$
- decoder: $z \sim p_{\theta}(z)$
- parameters ϕ are shared across all data points

Relations between graphical models and deep learning

- Start with some interest in an object $P(Y|X)$
  - could be as complicated as we want
  - say a deep neural network
    * just a glorified conditional distribution in a graphical model

Deepify Operator

- Sam Roweis: “Much better to invent an operator, than a new model. Model: 1 paper, Operator: long string of operators”
- “Deepify operator” - pick a graphical model with conditional distributions and replace those with a deep neural network
- Logits: deep NN
- Deep survival analysis: replace Cox’s proportional hazard function with a deep network

Deep Genrative Model: The Variational Auto-Encoder

- Hemholtz machine (80s)
- read old Geoffrey Hinton's papers and reinvent them
- we can now reintroduce his ideas
- deterministic NN node -> unobserved stochastic node -> observed stochastic node

Wake-Sleep Algorithm

- Stochastic variational Bayesian inference

-