CIS 520 Machine Learning
Summary

Lyle Ungar

What we covered
What we didn’t cover

Review Session: Friday
Final: Wednesday 3:00

Piazza; finish quizzes, worksheets
Course goals

◆ Be familiar with all major ML methods
  ● Regression (linear, logistic), regularization, feature selection
  ● K-NN, Decision trees, Random Forests, SVMs
  ● PCA, K-means, GMM, Autoencoders
  ● Naive Bayes, Bayes Nets, Markov Nets, LDA, HMMs
  ● Boosting, perceptrons, LMS
  ● Deep learning (CNNs, GANs)
  ● Reinforcement Learning (MDP, Q-learning)

◆ Know their strengths and weaknesses
  ● know jargon, concepts, theory
  ● be able to modify and code algorithms
  ● be able to read current literature

We did all of these!
Components of ML

- **Representation**
  - Feature set
  - Model form

- **Loss function**
  - And regularization penalty

- **Optimization method**
  - For parameter estimation
  - For model selection and hyperparameter tuning
Representations

◆ **Linear models**
  - Hyperplane as a separator
  - Kernel methods

◆ **Decision Trees**
  - Random forests, gradient tree boosting

◆ **Neural nets**
  - CNNs, (GANs and Conditional GANs)
  - Recurrent Nets/LSTMs

◆ **Structured X, y:** Graphs & Trees (not covered)
Representations

**Linear** (parametric)
- OLS
- Logistic regression
- HMM
- MDP

**Nonlinear** (semi-parametric)

**Nonlinear** (non parametric)
- K-NN
- Trees, Forests
## Representations

<table>
<thead>
<tr>
<th>Loss function</th>
<th>Bayesian (MLE/MAP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>OLS</td>
<td>OLS</td>
</tr>
<tr>
<td>Logistic regression</td>
<td>Logistic regression</td>
</tr>
<tr>
<td>PCA</td>
<td>mixture models: GMM, LDA</td>
</tr>
<tr>
<td>K-means</td>
<td>Belief Nets (NB, HMM, …)</td>
</tr>
</tbody>
</table>
Representations: Primal/Dual

**Primal**: feature space

- $X^TX$
- Covariance
- OLS

**Dual**: observation space

- $XX^T$
- Kernel matrix
- SVM
Invariances

Translational invariance

In Space: CNN, data augmentation

In time: RNN, CNN, RL
What loss functions have we used?

- L0, L1, L2
- Log-likelihood (MLE, MAP)
- Hinge
- Logistic
- Exponential
- Cross-Entropy; KL-divergence

**Boosting**: $\exp(-y_if_\alpha(x_i))$  
**Logistic**: $\log(1 + \exp(-y_if_w(x_i)))$
Loss Functions

- $L_0$
- Hinge
- Logistic
- Exponential (adaboost)
Regularization priors

\[ \text{Argmin}_w \| y - w \cdot x \|_2^2 + \lambda \| w \|_p^p \]

- **L₂** \( \| w \|_2^2 \)
  - Gaussian prior: \( p(w) \sim \exp(-\| w \|_2^2/\sigma^2) \)

- **L₁** \( \| w \|_1 \)
  - Laplace prior: roughly \( p(w) \sim \exp(-\| w \|_1/\sigma^2) \)

- **L₀** \( \| w \|_0 \)
  - Spike and slab prior

\[
\log P(D_X, D_Y, \theta) = \log P(D_X, D_Y \mid \theta) + \log P(\theta) = -\text{loss}(\theta) + \text{regularizer}(\theta)
\]
L₀, L₁ and L₂ Penalties

- If the x’s have been standardized (mean zero, variance 1) then we can visualize the shrinkage:

\[ L₂ = \text{Ridge} \]
sum of squares

\[ L₁ = \text{Lasso} \]
sum of abs value

\[ L₀ = \text{“stepwise regression”} \]
Number of features

Shrunk w → w

\[ \text{Shrunk w} \]

\[ \text{Shrunk w} \]

\[ \text{Shrunk w} \]
Bias-Variance Trade-off

\[ \mathbb{E}_{x,y,D}[(h(x; D) - y)^2] = \]

\[ \underbrace{\mathbb{E}_{x,D}[(h(x; D) - \bar{h}(x))^2]}_{\text{Variance}} + \underbrace{\mathbb{E}_x[(\bar{h}(x) - \bar{y}(x))^2]}_{\text{Bias}^2} + \underbrace{\mathbb{E}_{x,y}[(\bar{y}(x) - y)^2]}_{\text{Noise}} \]
Optimization methods

- **Gradient descent**: Stochastic, minibatch
- **Streaming/Online methods**: LMS, Perceptron
- **Closed form** (e.g. \( w = (X^TX)^{-1}X^Ty \))
- **Search**: streamwise, stepwise, stagewise
- **Power method** (for eigenvectors, SVD)
- **Lagrange Multipliers** (constrained optimization)
  - not covered
Optimization methods - alternating

- **EM** (alternating gradient descent in likelihood)
  - **E**: expected value of hidden values
  - **M**: MLE or MAP estimate of parameters

- **Other alternating methods**
  - **X \sim SW^T** for ICA, NNMF (non-negative matrix factorization)
  - **RL**: V or Q and policy
  - **Response surface**: Model and optimum
Hyperparameter Optimization

◆ Search
  ● e.g., $L_1$, $L_2$, penalties
  ● Neural network structure, regularization

◆ Auto-SKlearn
  ● Initialize hyperparameters from model predicting accuracy as a function of problem description and hyperparameter values

◆ Auto-ML
  ● Use reinforcement learning to learn a ‘design policy’
Distance and similarity

◆ Distances from norms
  - $\|x_1 - x_2\|_0$  $\|x_1 - x_2\|_1$  $\|x_1 - x_2\|_2$  ...

◆ Similarities from kernels
  - $k(x_1, x_2)$

◆ Probability-based divergence
  - $D_{KL}(p\|q) = \sum_k p_k \log(p_k/q_k)$  - KL-divergence
  - $H(p, q) = H(p) + D_{KL}(p\|q)$  - cross-entropy
    - $= - \sum_k p_k \log(q_k)$

  - $p$ is the true distribution, $q$ is the approximation
Cross entropy and log-likelihood

◆ Cross-entropy
  - $H(p,q) = - \sum_k p_k \log(q_k)$ summed over labels $k$
  - $- \sum_i \sum_k \delta_{ik} \log(p(y_i=k|x=x_i))$ \hspace{1cm} $\delta_{ik} = 1 \text{ iff } y_i=k$
  - Sum of the estimated log probabilities of the true answers
  - $\log \prod_i p(y_i|x_i) = \sum_i \log p(y_i|x_i)$ log-likelihood
KL-Divergence

- $D_{KL}(p||q) = \sum_k p_k \log(p_k/q_k)$

- **Mutual information**
  - $\text{MI}(X,Y) = D_{KL}(P(X,Y) || P(X)P(Y))$

- **Information gain**
  - $\text{IG}(Y|X_j) = D_{KL}(P(Y|X_j) || P(Y)) = H(Y) - H(Y|X_j)$
  - Which feature $X_j$ will maximize the information gain?

- **Bayesian Experimental Design**
  - For which $x$ will the label $y$ (in expectation) most change $p(w)$

https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence
Types of Learning

- **Supervised** \(X, y\)
  - Given an observation \(x\), what is the best label \(y\)?
- **Unsupervised** \(X\)
  - Given a set of \(x\)’s, cluster or summarize them
- **Reinforcement**
  - Given a sequence of states \(x\) and possible actions \(a\),
    learn which actions maximize reward.

What kind of learning is missing here?
Unsupervised methods

- PCA, ICA, NNMF
  - $X \sim S V^T$
- K-means, GMM, LDA
- Auto-encoders
  - Information bottleneck
  - Denoising
  - Variational

Most of these minimize reconstruction error subject to some constraints
Bayesian Belief Nets

- **NB**
  - Binary or real-valued X’s;
- **Belief Net**
- **GMM**
  - Different model forms
- **LDA**
- **HMM/MDP**
Reinforcement learning

◆ Model-based
  ● MDP

◆ Model-free
  ● Shallow: TD(0) vs. Deep: Monte-Carlo
  ● Value: V(s) vs. Q-learning Q(s,a)

◆ On policy (ε-greedy) vs. off-policy
  ● Trade off exploration and exploitation
Summary

Model-based

Response to all possible actions

Model-free

Response to one possible actions

One-step ahead

Search to end

From David Silver UCL Course on RL: http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching.html
For any MDP, given infinite exploration time and a partly-random policy, Q-learning will find an optimal policy: one that maximizes the expected value of the total reward over all successive steps.
Deep Q-Learning (DQL)

\[ \text{Argmin}_\theta \left[ Q(s, a; \theta) - \left( r(s, a) + \gamma \max_a Q(s', a; \theta) \right) \right]^2 \]

Update this To be closer to new value estimate

Represent \( Q \) with a neural net

\( s, a \) can be one-hot or real valued
Model Interpretation

- **Global: Explain the model**
  - Variable importance
    - E.g. how much does accuracy suffer if you remove it?

- **Global: Explain the world**
  - Variable importance
    - E.g. how does this variable affect the outcome?

- **Local: Why did you make this decision for this x?**
  - Decision tree: path taken
  - LIME (Local Interpretable Model-Agnostic Explanations)
What to use when? - Supervised

<table>
<thead>
<tr>
<th>n</th>
<th>p</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>10,000</td>
<td>PCR, Ridge, SVM, semi-supervised</td>
</tr>
<tr>
<td>10,000</td>
<td>10,000</td>
<td>SKlearn; Nnet with pretraining</td>
</tr>
<tr>
<td>100,000</td>
<td>100</td>
<td>SKlearn, Nnet</td>
</tr>
<tr>
<td>1,000,000</td>
<td>10,000</td>
<td>Nnet</td>
</tr>
</tbody>
</table>

Ask: What structure do I expect to see?
### What to use when? - Unsupervised

<table>
<thead>
<tr>
<th>n</th>
<th>p</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>100,000</td>
<td>K-means or PCA</td>
</tr>
<tr>
<td>10,000</td>
<td>10,000</td>
<td>K-means or PCA</td>
</tr>
<tr>
<td>100,000</td>
<td>10,000</td>
<td>K-means? GMM? Autoencoder?</td>
</tr>
<tr>
<td>10,000,000</td>
<td>images</td>
<td>Autoencoder</td>
</tr>
<tr>
<td>100,000,000</td>
<td>words</td>
<td>LDA, RNN/Transformer</td>
</tr>
</tbody>
</table>
Feature selection

- **Regression**
  - Do you expect very few, a moderate number of, or most features?

- **Random forests, gradient tree boosting**
  - Feature selection is ‘built in’

- **Neural nets**
  - Generally no feature selection
  - Screen features before you build the net
The biggest open problem: Transfer learning

◆ **Embeddings**
  - Images, words, products, people
  - Multimodal transformers …

◆ **Multitask learning**
  - Shared embeddings for many tasks

◆ **One shot learning**
  - Generalize from one label on one image …
The biggest open problem: Transfer learning

◆ Learn “deep structure”
  ● Reusable modules

◆ Build in “deep structure”
  ● Physics
  ● Ties to symbolic reasoning?
  ● Causal models

◆ Use external data
  ● Wikipedia, databases, …
The future ...

- Every company is an AI company
- The singularity??
See some of you for the review session, Friday class time
See all of you for the final, Wednesday 3:00 - online
Stay in touch & let me know how you use ML …

Thank you!!!