Types of Problems in Data Science

Dr. Fayyaz Minhas

Department of Computer Science
University of Warwick
https://warwick.ac.uk/fac/sci/dcs/teaching/material/cs909/
Common Theme

• Structural Risk Minimization
  – Loss function
    • Limits training error
      – Promotes learning from training data
  – Regularization
    • Capacity Control Term
    • Controls complexity of the boundary of the classifier
    • Doesn’t allow small changes in input produce large changes in the output
Classification

- **Supervised Classification**
  - Apple or orange
  - K-NN, Perceptron, Support Vector Machines, Decision Trees, Random Forests, XGBoost

- **Multi-class Classification**
  - Apple or orange or mango
  - For a binary classifier we can use
    - **One vs. All**
      - Apple vs. (Orange, Mango)
      - Orange vs. (Apple, Mango)
      - Mango vs. (Apple, Orange)
    - **One against One**
      - Apple vs. Orange
      - Apple vs. Mango
      - Orange vs. Mango
Classification: SVM

- **Representation:** \( f(x_i) = w^T x_i + b \)
  - Can use kernels to do an implicit transformation of the data to another feature space for solving linear separability problems

- **Evaluation: Structural Risk Minimization**
  - **Regularization:** Controls the complexity of the classifier
    \( w^T w \)
  - **Empirical error** (Hinge Loss)
    \[ \sum_{i=1}^{N} \max(0, 1 - y_i(w^T x_i + b)) \]

- **Optimization**
  - **Optimization Problem**
    \[ \min_w w^T w + \frac{C}{N} \sum_{i=1}^{N} \max(0, 1 - y_i(w^T x_i + b)) \]
  - Gradient Descent
  - Quadratic Programming, etc.
Dimensionality Reduction

• Reduce the dimensionality of the data
• Representation: \( f(x_i) = w^T x_i \)

• Evaluation
  – Regularization: Maximize margin, reduce norm of weight vector \( w^T w \)
  – Empirical error (Variance Loss due to projection)
    \[ V = w^T C w \]

• Optimization
  – Closed form solution: \( Cw = \lambda w \)
    • Eigen value problem
Regression

• Predicting a dependent variable (typically continuous) based on independent variables
  
  – Predict age of a person from an image
Regression: SVR

- **Representation**
  - \( f(x) = w^T x + b \)

- **Evaluation**
  - Regularization: \( w^T w \)
  - Error: Epsilon insensitive loss
  - \( l(f(x), y) = \max(0, |f(x) - y| - \epsilon) \)

- **Optimization Problem**

\[
\min_{w,b} \frac{1}{2} w^T w + \frac{C}{N} \sum_{i=1}^{N} \max(0, |f(x_i) - y_i| - \epsilon)
\]
Other problems

- Variants of linear models for classification and regression
- Feature selection
- Dimensionality Reduction
  - Encoding
  - Manifold Learning
- One class classification (Novelty/Outlier Detection)
- Learning to Rank (Ranking)
- Recommender Systems
- Survival (Failure/Churn) Prediction
- Clustering
- Reinforcement Learning
- ...
Variations on Linear Models

• Changing the loss function changes the behaviour of the model

\[
\text{OLS: } \min_{w,b} \sum_{i=1}^{N} (f(x_i) - y_i)^2
\]

\[
\text{Ridge Regression: } \min_{w,b} \alpha \|w\|^2 + \sum_{i=1}^{N} (f(x_i) - y_i)^2
\]

\[
\text{SVR: } \min_{w,b} \frac{1}{2} \|w\|^2 + \frac{C}{N} \sum_{i=1}^{N} \max(0, |f(x_i) - y_i| - \epsilon)
\]

\[
\text{SVM: } \min_{w,b} \frac{1}{2} \|w\|^2 + \frac{C}{N} \sum_{i=1}^{N} \max(0, 1 - y_i f(x_i))
\]
General Form

• Structural Risk Minimization

$$\min_{w,b} \lambda R(w) + \sum_{i=1}^{N} l(f(x_i, y_i))$$
Changing loss function

- Logistic Regression:
  \[ \min_{\mathbf{w}, b} \frac{1}{2} \| \mathbf{w} \|^2 + \frac{C}{N} \sum_{i=1}^{N} \log(\exp(-y_i f(x_i)) + 1) \]

- Uses logistic loss function
  - Used for classification
Understanding Regularization forms

• $L_p$ Regularization

• $R(w) = \|w\|_2^2 = w_1^2 + w_2^2 + \cdots + w_d^2$

• In general

  $\|w\|_p = (|w_1|^p + |w_2|^p + \cdots + |w_d|^p)^{1/p}$

  $\|w\|_1 = |w_1| + |w_2| + \cdots + |w_d|$

  $\|w\|_0 = \text{number of non-zero vector elements}$
Understanding Regularization forms

- Let’s take a vector

\[ \mathbf{w} = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} \]

- \[ \|\mathbf{w}\|_{p=2} = \sqrt{1 + 0.25} = 1.18 \]

- \[ \|\mathbf{w}\|_{p=1} = 1 + 0.5 = 1.5 \]

- \[ \|\mathbf{w}\|_{p=0} = 2 \]

- Minimizing \( \|\mathbf{w}\|_p \)
  - \( p = 2 \): pulls the point towards the origin
  - \( p = 1 \): reduces the axes coordinates individually
  - \( p = 0 \): reduces the number of non-zero components
Understanding Regularization forms

• Remember, output is a weighted combination of the input
  \[ f(x) = w^T x \]

• Minimizing \( \|w\|_p \)
  – \( p = 2 \): pulls the point towards the origin
    • Don’t allow weight elements from growing large
  – \( p = 1 \): reduces the axes coordinates individually
    • Reduce the magnitude of individual weights
    • Stronger penalty than when \( p = 2 \)
    • Smaller individual feature components
  – \( p = 0 \): reduces the number of non-zero components
    • Reduce the number of “active” features
    • Feature selection
Understanding Regularization forms

• Optimization
  – Difficult in cases other than when \( p = 2 \)
Other variants of linear models

• LASSO

\[ \min_{w,b} \alpha \|w\|_1 + \sum_{i=1}^{N} (f(x_i) - y_i)^2 \]

• Other variants

<table>
<thead>
<tr>
<th>Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ordinary Least Squares</td>
<td>( \min_w \left| Xw - y \right|^2 )</td>
</tr>
<tr>
<td>Ridge Regression</td>
<td>( \min_w \left| Xw - y \right|^2 + \alpha \left| w \right|^2 )</td>
</tr>
<tr>
<td>Lasso</td>
<td>( \min_w \frac{1}{2n_{\text{samples}}} \left| Xw - y \right|^2 + \alpha \left| w \right|_1 )</td>
</tr>
<tr>
<td>Multitask Lasso</td>
<td>( \min_w \frac{1}{2n_{\text{samples}}} \left| XW - Y \right|_{Fro}^2 + \alpha \left| W \right|_2^2 )</td>
</tr>
<tr>
<td>Elastic Net</td>
<td>( \min_w \frac{1}{2n_{\text{samples}}} \left| Xw - y \right|^2 + \alpha \rho \left| w \right|_1 + \frac{\alpha (1 - \rho)}{2} \left| w \right|_2^2 )</td>
</tr>
<tr>
<td>Multi-task Elastic Net</td>
<td>( \min_w \frac{1}{2n_{\text{samples}}} \left| XW - Y \right|_{Fro}^2 + \alpha \rho \left| W \right|<em>2^2 + \frac{\alpha (1 - \rho)}{2} \left| W \right|</em>{Fro}^2 )</td>
</tr>
<tr>
<td>LARS and LARS Lasso</td>
<td>Can get the full regularization path</td>
</tr>
<tr>
<td>Orthogonal Matching Pursuit</td>
<td>( \arg \min \left| y - X\gamma \right|_2^2 ) subject to ( \left| \gamma \right|<em>0 \leq n</em>{\text{nonzero coefs}} ) OR ( \arg \min \left| \gamma \right|_0 ) subject to ( \left| y - X\gamma \right|_2^2 \leq \text{tol} )</td>
</tr>
<tr>
<td>Logistic Regression (For classification)</td>
<td>( \min_{w,c} \frac{1}{2}w^Tw + C \sum_{i=1}^{n} \log(\exp(-y_i(X_i^Tw + c)) + 1) ).</td>
</tr>
<tr>
<td>Perceptron</td>
<td>Regularized Perceptron</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>Passive Aggressive Algorithm</td>
<td>For Robust regression</td>
</tr>
<tr>
<td>RANSAC</td>
<td></td>
</tr>
<tr>
<td>Thiel Sen</td>
<td>For robust regression</td>
</tr>
<tr>
<td>Huber Regressor</td>
<td></td>
</tr>
<tr>
<td>[ \min_{w,\sigma} \sum_{i=1}^{n} \left( \sigma + H_m \left( \frac{X_i w - y_i}{\sigma} \right) \sigma \right) + \alpha</td>
<td></td>
</tr>
<tr>
<td>[ H_m(z) = \begin{cases} z^2, &amp; \text{if }</td>
<td>z</td>
</tr>
</tbody>
</table>
How to program any model?

• If you can define a loss function
• And a regularizer
• The rest can be automated
  – For any problem*!

• Example solution:
  – https://github.com/foxtrotmike/CS909/blob/master/barebones.ipynb

*Terms and conditions apply
Clustering

• Grouping objects such that:
  – Objects within the same group (cluster) are more similar to each other
  – And different from objects in other groups

• Input
  – Typically, a Data Matrix (X)
  – Unsupervised technique
Original unclustered data

Clustered data
Clustering

• Using K-Means
  – Most commonly used algorithm for clustering

  – Input: Data Matrix X
  – Hyper-parameter: Number of clusters, Initial Cluster Centers

  – Output:
    • Assignment of each example to a cluster center
k-means Clustering

Initialize $m_i, i = 1, \ldots, k$, for example, to $k$ random $x_t$

Repeat

For all $x_t \in X$

$$b_i^t \left\{ \begin{array}{ll}
1 & \text{if } \|x^t - m_i\| = \min_j \|x^t - m_j\| \\
0 & \text{otherwise}
\end{array} \right.$$ 

For all $m_i, i = 1, \ldots, k$

$$m_i \leftarrow \sum_t b_i^t x^t / \sum_t b_i^t$$

Until $m_i$ converge

• $b_i^t$ is 1 when the $i^{th}$ center is the one closest to $x^t$
k-means Clustering
k-means Clustering
k-means Clustering
k-means Clustering
K-Means Clustering

• Hidden Hyperparameters
  – Distance metric
    • You can get different clustering based on the distance you use

• Regularization?
  – Cluster assignment of an example should not change within a short distance
  – The choice of your distance metric controls regularization
Hierarchical Clustering

• Build a hierarchy of clusters
• Agglomerative Clustering
  – Bottom up Approach
• Single Dimensional Example
Linkage

• How do we define the distance between clusters
  – Min
  – Max
  – Average
Linkage
Hierarchical clustering

• Allows us to represent the findings in a tree
• We can cutoff at any height to get different number of clusters

• Hyperparameters
  – Distance metric
  – Linkage
Hierarchical Clustering

- A “Phylogenetic” tree based on **genomic distance** for the SARS-CoV-2

[Diagram of a phylogenetic tree showing the relationship between different strains of SARS-CoV-2 and other viruses.]


https://nextstrain.org/ncov/global
Support Vector Clustering

• Ben-Hur and Vapnik 2001
• No assumptions on the shape and number of clusters
• Enclose all examples in a tight sphere centered at “a” with minimum radius “R”

\[ \min_{R,a} R^2 + C \sum_{i=1}^{N} \max(0, \|\phi(x_i) - a\|^2 - R^2) \]

• Can be kernelized
y_pred = KMeans(n_clusters=3).fit_predict(X)

https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html
<table>
<thead>
<tr>
<th>Method name</th>
<th>Parameters</th>
<th>Scalability</th>
<th>Usecase</th>
<th>Geometry (metric used)</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-Means</td>
<td>number of clusters</td>
<td>Very large n_samples, medium n_clusters with MiniBatch code</td>
<td>General-purpose, even cluster size, flat geometry, not too many clusters</td>
<td>Distances between points</td>
</tr>
<tr>
<td>Affinity</td>
<td>damping, sample preference</td>
<td>Not scalable with n_samples</td>
<td>Many clusters, uneven cluster size, non-flat geometry</td>
<td>Graph distance (e.g. nearest-neighbor graph)</td>
</tr>
<tr>
<td>propagation</td>
<td>bandwidth</td>
<td>Not scalable with n_samples</td>
<td>Many clusters, uneven cluster size, non-flat geometry</td>
<td>Distances between points</td>
</tr>
<tr>
<td>Mean-shift</td>
<td>bandwidth</td>
<td>Not scalable with n_samples</td>
<td>Many clusters, uneven cluster size, non-flat geometry</td>
<td>Distances between points</td>
</tr>
<tr>
<td>Spectral</td>
<td>number of clusters</td>
<td>Medium n_samples, small n_clusters</td>
<td>Few clusters, even cluster size, non-flat geometry</td>
<td>Graph distance (e.g. nearest-neighbor graph)</td>
</tr>
<tr>
<td>clustering</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ward hierarchical clustering</td>
<td>number of clusters or distance threshold</td>
<td>Large n_samples and n_clusters</td>
<td>Many clusters, possibly connectivity constraints</td>
<td>Distances between points</td>
</tr>
<tr>
<td>Agglomerative</td>
<td>number of clusters or distance</td>
<td>Large n_samples and n_clusters</td>
<td>Many clusters, possibly connectivity constraints, non Euclidean distances</td>
<td>Any pairwise distance</td>
</tr>
<tr>
<td>clustering</td>
<td>threshold, linkage type, distance</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DBSCAN</td>
<td>neighborhood size</td>
<td>Very large n_samples, medium n_clusters</td>
<td>Non-flat geometry, uneven cluster sizes</td>
<td>Distances between nearest points</td>
</tr>
<tr>
<td>OPTICS</td>
<td>minimum cluster membership</td>
<td>Very large n_samples, large n_clusters</td>
<td>Non-flat geometry, uneven cluster sizes, variable cluster density</td>
<td>Distances between points</td>
</tr>
<tr>
<td>Gaussian mixes</td>
<td>many</td>
<td>Not scalable</td>
<td>Flat geometry, good for density estimation</td>
<td>Mahalanobis distances to centers</td>
</tr>
<tr>
<td>Birch</td>
<td>branching factor, threshold, optional</td>
<td>Large n_clusters and n_samples</td>
<td>Large dataset, outlier removal, data reduction</td>
<td>Euclidean distance between points</td>
</tr>
<tr>
<td></td>
<td>global clusterer.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

One Class Classification

- Unary Classification
- Class Modelling
- Novelty Detection
- Examples for one class only (say normal)

**Identify those examples that differ from the given class**

Concept Descriptor

Boundary $f(x,w)=0$
OCC: Nearest Neighbor Data Descriptor

- Compares the distance of the test sample from its nearest neighbor and the distance of the nearest neighbor to its own nearest neighbor

\[ f_{NN^{tr}}(z) = I \left( \frac{\|z - NN^{tr}(z)\|}{\|NN^{tr}(z) - NN^{tr}(NN^{r}(z))\|} \leq 1 \right) \]

- The distances of k-nearest neighbors can be averaged

---

OCC: Support Vector Data Descriptors

- Finds a hyper-sphere with center at ‘a’ and radius $R$ so that the target class examples lie within the hyper-sphere
  - Error function: Penalize training examples if they lie outside the hypersphere

$$\min_{a,R} R^2 + \frac{C}{N} \sum_{i=1}^{N} \max(0, \|x - a\|^2 - R^2)$$
One-Class SVM (Scholkopf 2001)

- Separate points of the target class from the origin with maximum margin
- Linear Separability Case $f(x) = w^T x + b$
  - All given (target) class examples should have $f(x) \geq 0$
  - Consider that the outliers are mapped to the origin $f(0) = b < 0$

- Loss function
  - Error when
    - A point of the target class produces $f(x) < 0$
      $f(x) = w^T x + b < 0$
    - At Origin, $f(0) > 0$
      $f(0) = w^T 0 + b = b > 0$
One-Class SVM (Scholkopf 2001)

• Loss function thus becomes
  \[ \max(0, -f(x)) + b \]

• The resulting Structural Risk Minimization can be written as
  \[
  \min_{w, b} \frac{1}{2} w^T w + \frac{C}{N} \sum_{i=1}^{N} \max(0, -(w^T x_i + b)) + b
  \]

• Can be kernelized
Abnormal Beat Detection in ECG

• Problem
  – ECG based automated detection of abnormal beats has low generalization across individuals

• Solution
  – Use OCC to train on the normal beats for each individual

• Advantages
  – Expected to improve accuracy
  – Do not have to give any abnormal beats
Proposed Idea

• Use a person-specific classifier

• Problems?
  – A typical classifier requires both positive (abnormal) and negative (normal) examples for training
  – Number of positive examples is small and we do not know when the next abnormal beat is going to occur

• Solution
  – Use One-class classifiers or novelty detectors
Details about the Data

- 46 Records from MIT-BIH Database with lead MLII
- 73,258 normal (~69.0%) and about 32,827 (~31%) abnormal beats
- Evaluation Metrics: AUC
Performance: Over first N beats

![Performance Over First N Beats](image-url)
Performance: Over Randomly selected N beats
Learning to Rank

• Assign a rank to an input example

• Apple grading
  – U.S. Extra Fancy
  – U.S. Fancy
  – U.S. No. 1
  – U.S. No. 1 Hail
  – U.S. Utility
Learning to Rank

- Cancer Grading

<table>
<thead>
<tr>
<th>Grade 1</th>
<th>Grade 2</th>
<th>Grade 3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Glandular/Tubular Differentiation:</strong> &gt;75% of tumor forms glands</td>
<td><strong>Glandular/Tubular Differentiation:</strong> 10% to 75% of tumor forms glands</td>
<td><strong>Glandular/Tubular Differentiation:</strong> &lt;10% of tumor forms glands</td>
</tr>
<tr>
<td><strong>Nuclear Pleomorphism:</strong> Uniform cells with small nuclei similar in size to normal breast epithelial cells</td>
<td><strong>Nuclear Pleomorphism:</strong> Cells larger than normal with open vesicular nuclei, visible nucleoli, and moderate variability in size and shape</td>
<td><strong>Nuclear Pleomorphism:</strong> Cells with vesicular nuclei, prominent nucleoli, marked variation in size and shape</td>
</tr>
<tr>
<td><strong>Mitotic Count:</strong> &lt; 7 mitoses per 10 high power fields</td>
<td><strong>Mitotic Count:</strong> 8-15 mitoses per 10 high power fields</td>
<td><strong>Mitotic Count:</strong> &gt; 16 mitoses per 10 high power fields</td>
</tr>
</tbody>
</table>
Classification, Regression, Ranking

• Classification
  – Assign into classes
    • Typically no semantic relationship between classes (you cannot define greater than or less than in apples vs. oranges)

• Regression
  – Assign continuous variables
    • Age: 21 is greater than 18 (a relationship exists)

• Ranking
  – Assigning ranks
    • This is better than that
Ranking Error

• If each of the $m$ instances need to be ranked higher than $n$ other instances, then the average ranking error will be
  
  – Number of pairs that are not ranked appropriately

\[
\text{er}_S(f) = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} 1(f(x_i^+) < f(x_j^-))
\]
Ranking Error

• What will be the classification and ranking errors of the following ranking functions?

Example 1

\[ f_1 \]  
\[ \text{Classification error } = \frac{1}{4} \]  
\[ \text{Ranking error } = \frac{1}{4} \]

\[ f_2 \]  
\[ \text{Classification error } = \frac{1}{4} \]  
\[ \text{Ranking error } = \frac{1}{2} \]

Example 2

\[ f \]  
\[ \text{Classification error } = \frac{1}{100} \]  
\[ \text{Ranking error } = \frac{1}{2} \]
Ranking in Information Retrieval

warwick

Google Search  I'm Feeling Lucky
Ranking in Drug Discovery

• Given: A large number of molecules from a chemical library
• Output: Which one is the most promising as a drug against a virus?
Generalized Instance Ranking Problem

• Given
  – Input:
    • Training samples as pairs such that one $x_i$ is to be ranked higher than the other $x'_i$ by an amount $r_i$
    • $S = \{(x_1, x'_1, r_1), \ldots, (x_m, x'_m, r_m)\} \in (X^2 \times R^+)^m$
      – Pairs of examples $(x_i, x'_i)$
      – Rank difference $r_i$
    • For example:
      – A pair of webpages and the difference between their ranks
  – Output
    • Ranking function: $f : X \rightarrow R$
      $f(x_i)$ is the rank for example $x_i$
      We want: $f(x_i) - f(x'_i) \geq r_i$
Generalized Instance Ranking Problem

• Misclassification vs. mis-ranking
  – Mis-classification: assign wrong classification label
  – Mis-ranking: one example should have been ranked higher than the other but is not

• Generalized ranking loss:

\[
l(x_i, x'_i, r_i; f) = \max \left( 0, r_i - (f(x_i) - f(x'_i)) \right)
\]

\[
\min_{w,b} \frac{1}{2} w^T w + \frac{C}{N} \sum_{i=1}^{N} l(x_i, x'_i, r_i; f)
\]
Predicting anti-CRISPR proteins

• Identify if a protein in a set of proteins is an Anti-CRISPR protein
  – Given: sets of sets of proteins (proteomes) in which at least one protein is an anti-CRISPR protein
    • Only 20 examples
  – Required: Rank the known anti-CRISPR protein higher than non-anti-CRISPR proteins in the proteome

• Used ranking constraints
• Able to identify new anti-CRISPR proteins in new species

Recommendation Systems

• Task
  – Predict the rating or preference a user would give an item

• Given:
  – Training data
    • Items rated by users

• Other names
  – Matrix Completion Problem
  – Information Filtering Problem

## Examples

<table>
<thead>
<tr>
<th>MOVIE / USERS</th>
<th>Alice (1)</th>
<th>Bob (2)</th>
<th>Carol (3)</th>
<th>Dave (4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Love at last</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Romance Forever</td>
<td>5</td>
<td>?</td>
<td>?</td>
<td>0</td>
</tr>
<tr>
<td>Cute puppies of love</td>
<td>?</td>
<td>4</td>
<td>0</td>
<td>?</td>
</tr>
<tr>
<td>Nonstop car chases</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Swords vs. Karate</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>?</td>
</tr>
</tbody>
</table>
Applications

• Youtube
• Facebook
• Amazon
• Last.fm
• Netflix
  – Competition: 2006-2009 $1, 100M movie ratings, 107 algorithms
• Pandora

https://en.wikipedia.org/wiki/Recommender_system
How would you solve this problem?

- Not all users will have rated all movies
- You can get new users
- You can get new movies

- User Features
- Movie Features
Modelling the problem

• Represent each movie by its features: \( x_i \)
• Represent the output of your ML model function for a user \( j \) as: \( w_j^T x_i \)
• Whenever a user \( j \) rates a movie \( i \), we set a flag \( s_{ij} \) to 1 otherwise 0
• The rating for the movie \( i \) by user \( j \) is \( y_{ij} \)
• Error function
  – Prediction error for all labelled movies
    \[
    E = \frac{1}{2} \sum_{j=1}^{u} \sum_{i=1}^{m} \sum_{s_{ij}=1} (w_j^T x_i - y_{ij})^2
    \]
• How to find user parameters \( w_j \) and movie features \( x_i \)
Collaborative Filtering

• Start with small random or user set values for both $w_j$ and $x_i$

• Given the movie features (and the movie ratings)
  – We can estimate the user parameters (weights)

• Given the user-parameters (and the movie ratings)
  – We can estimate the movie features

• We can, ask the users for initial preferences and then estimate the features and then update the preferences and keep going until convergence

$w \rightarrow x \rightarrow w \rightarrow x \rightarrow \cdots$
Collaborative Filtering

• Given $x_i, i = 1 \ldots m$ estimate $w_j, j = 1 \ldots u$

\[
\min_{w_{j=1\ldots u}} \frac{1}{2} \sum_{j=1}^{u} \sum_{i=1:s_{ij}=1}^{m} (w_j^T x^i - y_{ij})^2 + \frac{\lambda_w}{2} \sum_{j=1}^{u} \|w_j\|^2
\]

• Given $w_j, j = 1 \ldots u$ estimate $x_i, i = 1 \ldots m$

\[
\min_{x_{i=1\ldots m}} \frac{1}{2} \sum_{j=1}^{u} \sum_{i=1:s_{ij}=1}^{m} (w_j^T x^i - y_{ij})^2 + \frac{\lambda_x}{2} \sum_{i=1}^{m} \|x_i\|^2
\]
What can we do with this?

• Why is it called collaborative filtering?
• We can rank the movies that were not ranked by a user
• We can also identify similar movies
  • Nearest neighbors over $x^i$
• Or similar users
  • Nearest neighbors over $w^j$
• Or identify popular trends of movies
  – Average movie ratings across all users

Slides adapted from Dr. Andrew Ng’s videos on Recommender Systems
https://www.youtube.com/playlist?list=PL_npY1DYXHPT-3dorG7Em6d18P4JRFDvH
Required Reading

• Slides adapted from Dr. Andrew Ng’s videos on Recommender Systems
  
  • https://www.youtube.com/playlist?list=PL_npY1DYXHPT-3dorG7Em6d18P4JRFDvH

• Spark-MLIB
• Others
Semi-Supervised Classification

- Use unlabeled data set in conjunction with labeled data
- Useful in cases when the number of labeled examples is small
  - Difficult to obtain labeled examples
- Semi-Supervised SVM
- Can use indirect labeling constraints over unlabeled data as well when available
  - Let’s say we know that a pair of example has the same (but unknown) or different labels
Transductive SVM

- Use unlabeled data set in conjunction with labeled data to specifically label the unlabeled data only
- Transductive SVM
Assign a rank to an input example

Apple grading
- U.S. Extra Fancy
- U.S. Fancy
- U.S. No. 1
- U.S. No. 1 Hail
- U.S. Utility

If $i$ is ‘US Extra fancy’ and $j$ is any other category then we want the score of $i$ to be higher than that of $j$, i.e.,:

$$f(x_i) \geq f(x_i) + 1 - \xi_{ij}$$

These are the constraints in a ranking SVM. The objective function minimizes the sum of slacks and maximizes the margin.
Multi-task learning

- Simultaneously predicting related tasks
  - Predict Apple Grade and Apple Variety Simultaneously
- Tasks are related
- If the tasks are related to each other then it might be useful to do multi-task learning instead of training independent classifiers
Transfer learning

- Use information from one task to learn to classify another

Task-1
- Apple
- Apple
- Orange
- Orange

Task-2
- Pear
- Pear
- Mango
- Mango
Multi-label learning

- In certain problems, an example can belong to more than one class
  - Example
    - Does a picture have:
      - Apples
      - Oranges
      - Mangoes
Structured output learning

- When the output is a structured object
  - Single label or real value
  - Vector
  - Directed Acyclic Graph
  - Relationships between output variables

- Structured SVM
- Most generic
  - Can be used for
    - Multi-class
    - Multi-label
    - ...
Structured output learning

- Machine translation
Structured output learning

- Sequence labeling as structured output learning
  - Given a sequence, predict the labels
  - Example:
    - Finding what keys were pressed using audio recording of keyboard emnations
      - Uses a hidden Markov model
      - Can use a structured SVM here

Text recognized by the HMM classifier, with cepstrum features (underlined words are wrong):

the big money fight has drawn the shoporo od doses of companies in the entertainment industry as well as attorneys general on states, who fear the field shading software will encourage illegal acivitt, stem the growth of small arista and lead to lost cobs and diminished sales tax revenue.

Text after spelling correction using trigram decoding:

the big money fight has drawn the support of dozens of companies in the entertainment industry as well as attorneys general in states, who fear the file sharing software will encourage illegal activity, stem the growth of small artists and lead to lost jobs and finished sales tax revenue.

Original text. Notice that it actually contains two typographical errors, one of which is fixed by our spelling corrector:
Active Learning

- **Active Learning**
  - Special case of semi-supervised learning in which a learning algorithm is able to interactively query the user (or some other information source) to obtain the desired outputs at new data points

- **Useful when**
  - Manual labeling is expensive
  - Small set of labeled examples
  - Large set of unlabeled examples
  - Can ask about the labels of some examples in the unlabeled set from an oracle
  - Don’t ask too much and the oracle can also make errors: Proactive learning

The Oracle (The Matrix, 1999)

Is that an orange?

It’s the one!
Online Learning

- Online Learning
  - Learn and unlearn concepts
  - Concepts can change over time
  - Example: Price of apples vs. their grade
  - Online learning allows incremental learning over time so we don’t have to retrain the classifier every time

The illiterate of the 21st century will not be those who cannot read and write, but those who cannot learn, unlearn, and relearn.
- Alvin Toffler

Online learning is a method of machine learning in which data becomes available in a sequential order and is used to update our best predictor for future data at each step, as opposed to batch learning techniques which generate the best predictor by learning on the entire training data set at once.
Representation Learning

• Learn how to represent data for different machine learning tasks
  – Replacement of feature engineering or hand crafted features

• Deep Learning

• Sparse representations

Google’s Artificial Brain Learns to Find Cat Videos
Self-Taught Learning

- Self-taught learning
  - Learn features automatically using completely unrelated images

Apple  Orange  Snowflake  Dog
Reinforcement Learning

- Reinforcement learning
  - Learn to fly a plane or play Tic-Tac-Toe by yourself based only on rewards or penalties
  - If a function that gives rewards or penalties based on how the machine behaves, you can develop a reinforcement algorithm for such a problem
    - Maximize rewards!

https://www.youtube.com/watch?v=N20h6vpR13Y
Survival Prediction

• Predict survival/time to event
  – Time before a patient dies
  – Time before a machine fails
  – ...
  – Time before the lecture ends 😊

• Input
  – Features of an example

• Output
  – Time to event

Issues

• Censoring
  – You may have event data for some at the current time
  – However, for some the event may not have yet occurred
    • Right Censored
  – Different examples may have different start times
    • Left censored
How would you solve it?

- Other problems
  - Ordinal regression
  - Robust regression
  - Survival prediction
  - Segmentation
  - Object detection and localization
  - Object counting
  - Generating Images / Data
  - Biclustering
  - Learning to learn
  - Learning to optimize
  - Feature Selection
  - Ranking
  - Structured-output prediction
  - Multiple instance learning
  - Multi-task learning
  - Multi-label learning
  - Semi-Supervised learning
  - Transductive learning
  - Self-taught learning
  - Online learning
  - Active learning
  - Curriculum learning
  - Transfer learning
  - Contrastive learning
“Nearly everything is really interesting if you go into it deeply enough.”

(Feynman)