Foundations of Large-Scale Multimedia Information Management and Retrieval

Lecture #3 Machine Learning

Edward Y. Chang
Foundations of Large-Scale Multimedia Information Management and Retrieval

Edward Y. Chang

Foundations of Large-Scale Multimedia Information Management and Retrieval Mathematics of Perception covers knowledge representation and semantic analysis of multimedia data and scalability in signal extraction, data mining, and indexing. The book is divided into two parts: Part I - Knowledge Representation and Semantic Analysis focuses on the key components of mathematics of perception as they apply to data management and retrieval. These include feature selection/reduction, knowledge representation, semantic analysis, distance function formulation for measuring similarity, and multimodal fusion. Part II - Scalability Issues presents indexing and distributed methods for scaling up these components for high-dimensional data and Web-scale datasets. The book presents some real-world applications and remarks on future research and development directions.

The book is designed for researchers, graduate students, and practitioners in the fields of Computer Vision, Machine Learning, Large-scale Data Mining, Database, and Multimedia Information Retrieval.

Dr. Edward Y. Chang was a professor at the Department of Electrical & Computer Engineering, University of California at Santa Barbara, before he joined Google as a research director in 2006. Dr. Edward Y. Chang received his M.S. degree in Computer Science and Ph.D degree in Electrical Engineering, both from Stanford University.
Machine Learning Approaches

• Introduction
• Linear Models
  – Large D
  – D >> N
• Generative vs. Discriminative Models
• Non-Linear Models
Statistical Learning

• Program the computers to learn!
• Computers improve performance with experience at some task
• Example:
  – Task: playing checkers
  – Performance: % games it wins
  – Experience: expert players
Statistical Learning

• Task $\hat{Y} = f(U)$
  – Represented by some model(s)
  – Implies hypothesis
• Performance
  – Measured by error functions
• Experience (L)
  – Characterized by training data
• Algorithm ($\Phi$)
Supervised Learning

• X: Data
  – U: Unlabeled pool
  – L: Labeled pool

• G: Labels
  – Regression
  – Classification

• Φ: Learning algorithm

• f = Φ(L)

• Ŷ = f(U)
Learning Algorithms $\Phi$

- Linear Model
- K-NN
- Kernel Methods
- Neural Networks
- Probabilistic Graphic Models
- Decision Trees
- Etc.
Linear Model

Figure 2.1: A classification example in two dimensions. The classes are coded as a binary variable—
GREEN = 0, RED = 1—and then fit by linear regression. The line is the decision boundary defined by \( x^T\beta = 0.5 \). The red shaded region denotes that part of input space classified as RED, while the green region is classified as GREEN. [T. Hastie, etc. 2001]
Linear Model

- \( \hat{y} = w_0 + \sum X_j w_j \) (\( j = 1 \) to \( d \))
- \( X \) is an \( n \times d \) matrix
  - \( d \): data dimension
  - \( n \): number of training instances
- \( \hat{y} = Xw \)
- \( L(w, S) = \text{RSS}(w) = (y - Xw)^T(y - Xw) \)
  - \( \text{RSS}: \) Residual Sum of Square
- \( \frac{\partial L(w, S)}{\partial w} = -2X^T y + 2X^T X w = 0 \)
- \( w = (X^T X)^{-1} X^T y \)
Three Challenges

• D is too large
  – Curse of dimensionality

• D > N
  – Insufficient samples

• N is too large
  – Later...
Gene Profiling Example

N = 59 cases, D = 4026 genes
Subset Selection & Shrinkage

• Least Square often suffers from large variance
• Shrinkage sets some coefficients to zero
• Algorithms
  – Forward Stepwise Selection
  – Backward Stepwise Selection
  – Ridge Regression
Ridge Regression

\[ w = \arg\min_w \left\{ \sum_n \left( y_i - w_0 - \sum_d x_{ij} w_j \right)^2 + \lambda \sum_d w_j^2 \right\} \]

• Why would this help?
  – Regularization: remedying an ill-posed model
  – Correlated variables

• Data preparation
  – Normalize input
  – Centralize input (removing \( w_0 \))

• \[ w = (X^T X + \lambda I_d)^{-1} X^T y \]
Ridge Regression
Tikhonov Regularization

• \( \min L_\lambda (w S) = \min \lambda \ |w| \ |^2 \)
  \[ + \sum (y_i - f(x_i))^2 \]
  – As oppose to \( \min (y - Xw)^T(y - Xw) \)

• \( w = (X^TX + \lambda I_d)^{-1} X^Ty \)
  – As oppose to \( w = (X^TX)^{-1} X^Ty \)

• \( w = X^T\alpha \)
  – \( \alpha = (G + \lambda I_n)^{-1} y \) (\( G: n \times n \) Gram matrix)
# Regularization

<table>
<thead>
<tr>
<th>Model</th>
<th>Fit measure</th>
<th>Entropy measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIC/BIC</td>
<td>$|Y - X\beta|_2$</td>
<td>$|\beta|_0$</td>
</tr>
<tr>
<td>Basis pursuit denoising</td>
<td>$|Y - X\beta|_2$</td>
<td>$\lambda |\beta|_1$</td>
</tr>
<tr>
<td>Dantzig Selector[^4^]</td>
<td>$|X^\top(Y - X\beta)|_\infty$</td>
<td>$|\beta|_1$</td>
</tr>
<tr>
<td>Lasso[^2^]</td>
<td>$|Y - X\beta|_2$</td>
<td>$|\beta|_1$</td>
</tr>
<tr>
<td>Ridge regression</td>
<td>$|Y - X\beta|_2$</td>
<td>$|\beta|_2$</td>
</tr>
<tr>
<td>RLAD[^3^]</td>
<td>$|Y - X\beta|_1$</td>
<td>$|\beta|_1$</td>
</tr>
</tbody>
</table>
SVD Interpretation

Figure 3.8: Principal components of some input data points. The largest principal component is the direction that maximizes the variance of the projected data, and the smallest principal component minimizes that variance. Ridge regression projects y onto these components, and then shrinks the coefficients of the low-variance components more than the high-variance components.
PCR: Principal Component Regression

- SVD
- Discard components with smallest Eigen coefficients
- PCA Linear Multivariate Regression
  - Sum of univariate regressions
Limitations & Treatments

• High bias
• Low variance
Linear Model

Figure 2.1: A classification example in two dimensions. The classes are coded as a binary variable—GREEN = 0, RED = 1—and then fit by linear regression. The line is the decision boundary defined by $x^T \beta = 0.5$. The red shaded region denotes that part of input space classified as RED, while the green region is classified as GREEN. [T. Hastie, etc. 2001]
Limitations & Treatments

• High bias

• Low variance

• High-dimensional or overfitting
  – Ridge, Subset, Lasso
  – PCR, PLS
  – In general, Regularization
Generative vs. Discriminative Models

• Generative Models
  – Model entire distribution
  – One class at a time
  – Look for maximum likelihood

• Discriminative Models
  – Model class boundaries
  – Ignore distribution
  – Support Vector Machines (SVMs)
  – Perhaps better for large problems!
Maximum Likelihood View

- \( \hat{y} = w_0 + \sum w_j X_j \) (j = 1 to d)
- \( \hat{y} = Xw \)
- \( \hat{y} = Xw + \varepsilon \)
  - \( \varepsilon \) (noise signals) are independent
  - \( \varepsilon \rightarrow N (\hat{y}, \partial^2) \)
- \( P(\hat{y} \mid wx) \) has a normal dist. with
  - Mean at \( \hat{y} = wx \)
  - Variance \( \partial^2 \)
Derivation

• $P(\hat{y} \mid w, x) \rightarrow N(\hat{y}, \partial^2)$

• Training
  – Given $(x_1, y_1) (x_2, y_2) \ldots (x_n, y_n)$
  – Infer $w$ by training data
  – By Bayes rule, or
    Maximum Likelihood Estimate
Maximum Likelihood

- For what \( w \) is
  - \( P(y_1, y_2, \ldots y_n \mid x_1, x_2, \ldots x_n, w) \) maximized?
  - \( \prod P(y_i \mid wx_i) \) maximized?
  - \( \prod \exp(-\frac{1}{2}(y_i-wx_i/\partial)^2) \) maximized?
  - \( \sum (-\frac{1}{2}(y_i-wx_i/\partial)^2) \) maximized?
  - \( \sum (y_i-wx_i)^2 \) minimized?
Observations

• RSS = MAP

• What if n < d ?
  – Gradient Decent (Perceptron)
    • Converges only when instances are linearly separable, or behaves erratically
  – Dual Formulation
Dual View (Duality)

- **Primal**
  - \( w = (X^T X)^{-1} X^T y \) \((d \times d) \times (d \times n) \times (n \times 1)\)

- **Dual (if \((X^T X)^{-1}\) exists)**
  - \( w = X^T X (X^T X)^{-2} X^T y = X^T (X (X^T X)^{-2} X^T y) = X^T \alpha \)
  - \( w = \sum \alpha_i x_i, \ i = 1 \text{ to } n \)
  - \( \alpha = X (X^T X)^{-2} X^T y = G y \)
    - \( G: (n \times d) \times (d \times d) \times (d \times n) \rightarrow n \times n \) Gram matrix

- **When \( n < d \) \((X^T X)^{-1} \) does not exists**
  - Restrict (bias) the choice of functions
  - Regularization
Ridge Regression

\[ \min L_\lambda (w, S) = \min \lambda \left| \left| w \right| \right|^2 + \sum (y_i - f(x_i))^2 \]

- As oppose to \( \min (y - Xw)^T(y - Xw) \)

\[ w = (X^T X + \lambda I_d)^{-1} X^T y \]

- As oppose to \( w = (X^T X)^{-1} X^T y \)

\[ w = X^T \alpha \]

- \( \alpha = (G + \lambda I_n)^{-1} y \) (G: n x n Gram matrix)
## Primal vs. Dual

<table>
<thead>
<tr>
<th></th>
<th>Primal</th>
<th>Dual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training Cost</td>
<td>$O(d^3)$</td>
<td>$O(n^3)$</td>
</tr>
<tr>
<td>Classification</td>
<td>$O(d)$</td>
<td>$O(nd)$</td>
</tr>
</tbody>
</table>
### Primal vs. Dual

<table>
<thead>
<tr>
<th></th>
<th>Primal</th>
<th>Dual is the choice</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Training Cost</strong></td>
<td>$O(d^3)$</td>
<td>$O(n^3)$</td>
</tr>
<tr>
<td><strong>Classification</strong></td>
<td>$O(d)$</td>
<td>$O(nd)$</td>
</tr>
</tbody>
</table>
Models & Linearity

• Generative Models
  – Model entire distribution
  – One class at a time
  – Look for maximum likelihood

• Discriminative Models
  – Model class boundaries
  – Ignore distribution
  – Support Vector Machines (SVMs)
  – Perhaps better for large problems!
Gaussian Mixture Model
Support Vector Machine - Linear
Support Vector Machine – Nonlinear
Decision Tree

Dependent variable: PLAY

OUTLOOK?

sunny

Play 2
Don't Play 3

HUMIDITY?

<= 70

Play 2
Don't Play 0

> 70

Play 0
Don't Play 3

overcast

Play 4
Don't Play 0

rain

Play 3
Don't Play 2

WINDY?

TRUE

Play 0
Don't Play 2

FALSE

Play 3
Don't Play 0
Decision Tree Output
Boosted Decision Tree

• Multiple weak classifiers
  – Strength in numbers

• Emphasize mistakes
  – Put resources at hard cases

• Provable
  – Strong classifier
  – Converges
AdaBoost Example
Machine Learning Approaches

• Introduction
• Linear Models
  – Large D
  – D >> N
• Generative vs. Discriminative Models
• Non-Linear Models
Classical Model

• N: Number of training instances
• N⁺, N⁻
• D: Dimensionality
• N >> D  N → ∞
  – E.g., PAC learnability
• N⁻ ≈ N⁺
Emerging MM Applications

- $N < D$
- $N^+ << N^-$
- Examples
  - Information Retrieval with relevance feedback
  - Surveillance event detection
IR \rightarrow A Classification Problem
Apple Search
Relevance Feedback
Fruit
Step #1: Solicit Labels
Step #2: Compute Boundary
Step #3: Identify Useful Samples
Step #4: Solicit More Feedback
Step #5: Refine Boundary
Step #6: Ranking
Observations

- Identify good samples
- Collect diversified samples
- Is linear model sufficient?
IR $\rightarrow$ A Classification Problem
Non-Linear Boundary

(a) Input Space

(b) Projected Space
Separating Hyperplane
Separating Hyperplane
Maximum Margin Hyperplane
Linear Model Fits All Data?
Linear Model Fits All?
How about Joining the Dots?

- \( \hat{y}(x) = \frac{1}{k} \sum y_i, \)
  
  \(- x_i \in N_k(x) \)

- \( K = 1 \)
 NN with $k = 1$

Figure 2.3: The same classification example in two dimensions as in Figure 2.1. The classes are coded as a binary variable ($\text{GREEN} = 0, \text{RED} = 1$), and then predicted by 1-nearest-neighbor classification. [T. Hastie, etc. 2001]
Nearest Neighbor

• Four Things Make a NN Memory-Based Learner
  – A distance function
  – K: number of neighbors to consider?
  – A weighted function (optional)
  – How to fit with the local points?
Problems

• Fitting Noise
• Jagged Boundaries
Solutions

• Fitting Noise
  – Pick a Larger K?

• Jagged Boundaries
  – Introducing Kernel as a weighting function
NN with $k = 15$

Figure 2.2: The same classification example in two dimensions as in Figure 2.1. The classes are coded as a binary variable (GREEN = 0, RED = 1) and then fit by 15-nearest-neighbor averaging as in (2.8). The predicted class is hence chosen by majority vote amongst the 15-nearest neighbors. [T. Hastie, etc. 2001]
NN
Solutions

• Fitting Noise
  – Pick a larger K?

• Jagged Boundaries
  – Introducing Kernel as a weighting function
Nearest Neighbor -> Kernel Method

• Four Things Make a Memory Based Learner
  – A distance function
  – K: number of neighbors to consider? All
  – A weighted function: RBF kernels
  – How to fit with the local points? Predict weights
Kernel Method

• RBF Weighted Function
  – Kernel width holds the key
  – Use cross validation to find the “optimal” width

• Fitting with the Local Points
  – Where NN meets Linear Model
LM vs. NN

• Linear Model
  – $f(x)$ is approximated by a global linear function
  – More stable, less flexible

• Nearest Neighbor
  – K-NN assumes $f(x)$ is well approximated by a locally constant function
  – Less stable, more flexible

• Between LM and NN
  – The other models...
Where Are We and Where Am I Heading To?

- LM and NN
- Kernel Method of Three Views
  - LM view
  - NN view
  - Geometric view
Linear Model View

• \( Y = \beta_0 + \sum \beta X \)

• Separating Hyperplane
  – \( \text{Max}_{||\beta||=1} C \)
  – Subject to \( y_i f(x_i) \geq C \), or
  – \( y_i (\beta_0 + \beta x_i) \geq C \)
Maximum Margin Hyperplane
Classifier Margin

• Margin
  – Defined as with of the boundary before hitting a data object

• Maximum Margin
  – Tends to minimize classification variance
  – No formal theory for this yet
Separating Hyperplane

+1 Zone

-1 Zone

M
M’s Mathematical Representation

- **Plus-plane**
  - \( \{x: wx+b = +1\} \)

- **Minus-plane**
  - \( \{x: wx+b = -1\} \)

- \( w \perp \) Plus-plane
  - \( w(u - v) = 0, \) if \( u \) and \( v \) on plus-plane

- \( w \perp \) Minus-plane
Separating Hyperplane

+1 Zone

-1 Zone

X+

M

X−
M

• Let \( x^- \) be any point on minus-plane
• Let \( x^+ \) be the closest plus-plane-point to \( x^- \)
• \( x^+ = x^- + \lambda w \), why
  – The line \((x^+x^-) \perp \) minus-plane
• \( M = |x^+ - x^-| \)
1. \( wx^- + b = -1 \)
2. \( wx^+ + b = 1 \)
3. \( x^+ = x^- + \lambda w \)
4. \( M = |x^+-x^-| \)
5. \( w(x^- + \lambda w) + b = 1 \) (from 2 & 3)
6. \( wx^- + b + \lambda ww = 1 \)
7. \( \lambda ww = 2 \)
1. $\lambda ww = 2$
2. $\lambda = 2/ww$
3. $M = |x^+ - x^-| = |\lambda w| = \lambda |w| = 2/|w|$

4. Max $M$
   - Gradient decent, simulated annealing, EM, Newton’s method?
Max M

• Max $M = 2/|w|$
• Min $|w|/2$
• Min $|w|^2/2$
  – subject to $y_i(x_iw+b) \geq 1$
  – $i = 1,...,N$
• Quadratic criterion with linear inequality constraints
Max M

- Min $|w|^2/2$
  - subject to $y_i(x_iw+b) \geq 1$
  - $i = 1, \ldots, N$

- $L_p = \min_{w,b} |w|^2/2 + \sum_{i=1..N} \alpha_i[y_i(x_iw+b)-1]$

- $w = \sum_{i=1..N} \alpha_iy_ix_i$

- $0 = \sum_{i=1..N} \alpha_iy_i$
Wolfe Dual

- \( \text{Ld} = \sum_{i=1..N} \alpha - \frac{1}{2} \sum \sum_{i,j=1..N} \alpha_i \alpha_j y_i y_j x_i x_j \)

- Subject to
  - \( \alpha_i \geq 0 \)
  - \( \alpha_i [y_i(x_i w + b) - 1] = 0 \)
  - KKT conditions
    - \( \alpha_i > 0, y_i(x_i w + b) = 1 \) (Support Vectors)
    - \( \alpha_i = 0, y_i(x_i w + b) > 1 \)
Class Prediction

• \( y_q = w \cdot x_q + b \)
• \( w = \sum_{i=1..N} \alpha_i y_i x_i \)
• \( y_q = \text{sign}(\sum_{i=1..N} \alpha_i y_i (x_i \cdot x_q) + b) \)
Non-seperatable Classes

- Soft Margin Hyperplane
- Basis Expansion
Non-separating Case
Soft Margin SVMs

• Min $|w|^2/2$
  
  – subject to $y_i(x_iw+b) \geq 1$
  
  – $i = 1,\ldots,N$

• Min $|w|^2/2 + C \sum \varepsilon_i$
  
  – $x_iw+b \geq 1 - \varepsilon_i$ if $y_i = 1$
  
  – $x_iw+b \leq -1 + \varepsilon_i$ if $y_i = -1$
  
  – $\varepsilon_i \geq 0$
Non-separating Case
Wolfe Dual

• $L_d = \sum_{i=1..N} \alpha - \frac{1}{2} \sum_{i,j=1..N} \alpha_i \alpha_j y_i y_j x_i x_j$

• Subject to
  
  \begin{align*}
  - C & \geq \alpha_i \geq 0 \\
  - \sum \alpha_i y_i & = 0 \\
  - \text{KKT conditions}
  \end{align*}

• $y_q = \text{sign} (\sum_{i=1..N} \alpha_i y_i (x_i \cdot x_q) + b)$
Basis Function

\[ x = 0 \]
Harder 1D Example

\[ x = 0 \]
Basis Function

- $\Phi(X) = (x, x^2)$
Harder 1D Example
Some Basis Functions

• $\Phi(X) = \sum \gamma_m h_m(X)$
  – $h_m(X) \mathbb{R}^p \rightarrow \mathbb{R}$

• Common Functions
  – Polynomial
  – Radial basis functions
  – Sigmoid functions
Wolfe Dual

- $L_d = \sum_{i=1..N} \alpha - \frac{1}{2} \sum\sum_{i,j=1..N} \alpha_i \alpha_j y_i y_j \Phi(x_i) \Phi(x_j)$
- Subject to
  - $C \geq \alpha_i \geq 0$
  - $\sum \alpha_i y_i = 0$
  - KKT conditions

- $y_q = \text{sign} \left( \sum_{i=1..N} \alpha_i y_i (\Phi(x_i) \cdot \Phi(x_q)) + b \right)$

- $K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$
  - Kernel function!
Nearest Neighbor View

• Z, a set of zero mean jointly Gaussian random variables,
  – Each $Z_i$ corresponds to one example $X_i$
  – $\text{Cov}(z_i, z_j) = K(x_i, x_j)$

• $y_i$, the label of $z_i$, +1 or -1
  – $P(y_i | z_i) = \sigma(y_i, z_i)$
Training Data
General Kernel Classifier [Jaakkola, etc. 99]

- **MAP Classification for** \( x_t \)
  
  \[ y_t = \text{sign} \left( \sum \alpha_i y_i K(x_t, x_i) \right) \]
  
  \[ K(x_i, x_j) = \text{Cov} (z_i, z_j) \text{ (some similarity function)} \]

- **Supervised Training: Compute** \( \alpha_i \)
  
  - Given \( X \) and \( y \), and
  
  - An error function such as
  
  \[ J(\alpha) = -\frac{1}{2} \sum \alpha_i \alpha_j y_i y_j K(x_i, x_j) + \sum F(\alpha_i) \]
Leave One Out
SVMs

• $y_t = \text{sign} \left( \sum \alpha_i y_i K(x_t, x_i) \right)$

• $(y_i x_i)$ training data, $\alpha_i$ nonnegative, and kernel $K$ positive definite

• $\alpha_i$ is obtained by minimizing

\[- J(\alpha) = - \frac{1}{2} \sum \alpha_i \alpha_j y_i y_j K(x_i, x_j) + \sum F(\alpha_i) \]

\[- F(\alpha_i) = \alpha_i \]

\[- \alpha_i \geq 0, \sum y_i \alpha_i = 0 \]
SVMs
Important Insight

• $K(x_i, x_j) = \text{Cov}(z_i, z_j)$

• To design of a kernel is to design a similarity function that produces a positive definite covariance matrix on the training instances
Basis Function Selection

• Three General Approaches
  – Restriction methods
    • Limit the class of functions
  – Selection methods
    • Scan the dictionary adaptively (Boosting)
  – Regularization methods
    • Use the entire dictionary but restrict coefficients (Ridge Regression)
Overfitting?

• Probably Not
• Because
  – N free parameters (not D)
  – Maximizing margin
Summary of ML

• Introduction
• Linear Models
  – Large D
  – $D \gg N$
• Generative vs. Discriminative Models
• Nearest Neighbors
• Non-Linear Models

• Chapters 10, 11, 12: Large N
Reading

  - Chapter #3 Query-Concept Learning
  - Chapter #9 Imbalanced Data Learning