Machine Learning - Part I
Supervised Learning

Dongwoo Kim
What is Machine Learning?
Statistical Learning / Predictive Analytics

“Machine learning (ML) is the study of computer algorithms that improve automatically through experience”

by Tom Mitchell

• Can be categorized broadly into three categories based on problem settings
  • Supervised Learning
  • Unsupervised Learning
  • Reinforcement Learning (by Prof Ok, next week)
Supervised Learning
Supervised Learning
Problem settings

- Given *training data*
  
  - A finite set of examples \( x_i \in \mathcal{X} \) and their associated labels \( y_i \in \mathcal{Y} \)
  
  - \( \mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \)

- We want to find the ‘best’ estimator modeling the relationship between the \( x_i \) and the associated label \( y_i \), i.e. the best function

\[
f : \mathcal{X} \rightarrow \mathcal{Y}
\]

- Space of \( \mathcal{X} \) can be arbitrarily large (otherwise we can just memorize training data)
Supervised Learning
Probabilistic Framework

• Assumption: all examples are generated by some unknown (hidden) probabilistic source!

• Formally, $P(x, y)$ is an unknown probability distribution over $\mathcal{X} \times \mathcal{Y}$

• Training data $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ is i.i.d from $P$
  • Independent and identically distributed

• Estimating density is difficult!
Supervised Learning
General approach

\[ f : \mathcal{X} \rightarrow \mathcal{Y} \]

1. Define a class of functions (\(\textbf{models}\)) \(f\)
   - In other words, restrict possible functions (i.e. hyperplane)
2. Quantify ‘best’ as the optimum of some computable objective function
   - Based on \(\textbf{loss (risk)}\) between predicted and true value
3. Evaluate prediction performance on new \(\textbf{test data}\)
Loss, Risk
To quantify predictive performance

• **Loss:** The error for a particular example. $\ell(f(x_i), y_i)$

  • Example: 0/1 loss for classification: loss is 1 if $f(x_i) \neq y_i$ otherwise 0

• **Risk:** The expected loss for all data including unseen examples

  $$R(f) = \int_{x \times y} \ell(f(x), y) P(x, y) dx dy$$

• **Empirical Risk:** The average loss on training data

  $$R_{\text{emp}}(f) = \frac{1}{|D|} \sum_{(x_i, y_i) \in \mathcal{D}} \ell(f(x_i), y_i)$$
Classification & Regression

And their losses for $f : \mathcal{X} \rightarrow \mathbb{R}$

- **(Binary) Classification** (discrete output space): $\mathcal{Y} = \{-1, +1\}$
  - 0/1 loss: $\ell_{01}(f(x_i), y_i) = \begin{cases} 1 & \text{if } f(x_i) \cdot y_i < 0 \\ 0 & \text{otherwise} \end{cases}$
  - Margin-based hinge loss: $\ell_{\gamma}(f(x_i), y_i) = \max(0, -y_i \cdot f(x_i) + \gamma)$
    - Interpret magnitude of $f$ as a confidence

- **Regression** (continuous output space): $\mathcal{Y} = \mathbb{R}$
  - Squared loss: $\ell(f(x_i), y_i) = \frac{1}{2}(f(x_i) - y_i)^2$
Simple Regression Task
Get your hands dirty!

• Given $\mathcal{D} = \{(0,0), (1,3), (2,5), (3,6)\}$

1. Assume we have a \textit{finite} function $\mathcal{F} = \{f_1(x) = x, f_2(x) = 2x, f_3(x) = 3x\}$

2. (unnormalized) Empirical risk of each function with respect to the squared error is

   • $R(f_1) = (f_1(0) - 0)^2 + (f_1(1) - 3)^2 + (f_1(2) - 5)^2 + (f_1(3) - 6)^2 = 22$

   • $R(f_2) = 2$

   • $R(f_3) = 10$

3. We can evaluate the empirical risk to find the best $f^*$
Infinite Function Class
How can we handle this?

- Let $\mathcal{F} = \{ f_\alpha(x) = \alpha \cdot x \mid \alpha \in \mathbb{R} \}$
  - A function class with *infinitely many* functions
  - We cannot compute the empirical risk of every function in $\mathcal{F}$
- In this case, we can analytically compute empirical risk of $f_\alpha$ given $\alpha$
- For example, with the previous dataset

$$R(f_\alpha) = (f_\alpha(0) - 0)^2 + (f_\alpha(1) - 3)^2 + (f_\alpha(2) - 5)^2 + (f_\alpha(3) - 6)^2$$

$$= (\alpha - 3)^2 + (2\alpha - 5)^2 + (3\alpha - 6)^2$$

$$= 14\alpha^2 - 28\alpha + 70$$
Optimization (Training)

How to find best $\alpha$

- $R(f_\alpha) = 14\alpha^2 - 28\alpha + 70$

- It would be the best if we can find $\alpha$ which makes $R(f_\alpha) = 0$.

- However, unfortunately, there is no such $\alpha \in \mathbb{R}$

- We can still find $\alpha$ which minimize $R(f_\alpha)$
  - via gradient descent
Gradient descent
Path to minimum

- We can compute the gradient of $R(f_\alpha)$ with respect to $\alpha$.

- If we change $\alpha$ slightly to the opposite direction of gradient, then we can reduce $R(f_\alpha)$.

$$\alpha_{\text{new}} = \alpha - \lambda \frac{d R(f_\alpha)}{d\alpha}$$

- This is the idea behind gradient descent.

$R(f_\alpha) = 14\alpha^2 - 28\alpha + 70$
Optimization

• Gradient descent is just one way to find the best model parameter.
  • We can only apply when the gradient of empirical risk is computable.
• For a classification, 0/1 loss is theoretically sounding.
  • However, we cannot compute the gradient of 0/1 loss.
• We use surrogate losses (similar to 0/1 loss but more tractable) such as hinge loss.
Empirical Risk Landscape
Convex vs Non-convex

- Convex (e.g., SVMs): guarantee to find global minimizer
- Non-convex (e.g., Neural Nets): hard to find global minimizer but more flexible
Models with more than 1 parameter
Empirical Risk vs True Risk
Statistical learning theory

• But how can we ensure that empirical risk minimize true risk?

• There are at least four quantities of interest

\[ R(f^*), \quad f^* = \arg \min_f \int_{\mathcal{X} \times \mathcal{Y}} \ell(f(x_i), y_i) P(x, y) dx dy \]

\[ R_{\text{emp}}(f^*) = \sum_{(x_i, y_i) \in \mathcal{D}} \ell(f^*(x_i), y_i) \]

\[ R_{\text{emp}}(f'), \quad f' = \arg \min_f \sum_{(x_i, y_i) \in \mathcal{D}} \ell(f(x_i), y_i) \]

\[ R(f') = \int_{\mathcal{X} \times \mathcal{Y}} \ell(f'(x_i), y_i) P(x, y) dx dy \]
Some interesting questions

• Can we show that $|R_{\text{emp}}(f') - R(f')| < \epsilon$
  
  • for some class of models $\mathcal{F}$ with some loss $\ell$
  
  • under an arbitrary data distribution $P$
  
  • How many samples do we need to achieve this?
  
• Can we show that $R(f') - R(f^*) < \epsilon$
Probably Approximately Correct
With finite function classes

Let $\mathcal{F}$ be a finite model class, $\hat{f}_n = \arg\min_f R^n_{\text{emp}} f$ (empirical risk minimizer with $n$ observations). Then

$$R(\hat{f}_n) - \min_{f \in \mathcal{F}} R(f) \leq C \sqrt{\frac{\log |\mathcal{F}|}{n}},$$

with probability $\geq 1 - \delta$ under certain assumptions* ($C < \infty$).

*The assumption is quite strong.
Generalized Problems
Learning to Rank

• Ranking (ordering) is more important than some score in some applications
  • Movie recommendation:
    • Predicting a rating of movie is unnecessary for recommendation
    • Predicting a preference is more important
  • Information retrieval (document search):
    • The most relevant documents need to be on top of the search list
Pair-wise Loss for Ranking

Raking SVM

• Assume scoring function $f$ that gives relatively high score for more relevant items.

• For example, if $x_1$ is ranked higher than $x_2$, then $f$ needs to satisfy: $f(x_1) > f(x_2)$

• From above we can define pairwise ranking loss

$$L = \sum_{i,j} \max(f(x_i) - f(x_j) + \xi,0),$$

where ranking of $x_i$ is lower than $x_j$

• By learning $f$ that minimize $L$, we can learn a ranking model.
Active Learning
Human in the loop

• Creating training data requires labelling of data set.
• Cost of labelling is expensive.
• Let’s not label the entire data.
• Aim: build a model that performs well with a small number of labelled data.
Which one generalize better?

Label near decision boundary

* From Active Learning Literature Survey by B. Settles
Query strategy
Which sample will be labelled?

- Least Confident
  - Selects the instance for which it has the least confident in its most likely label (x1)

- Margin Sampling
  - Selects the instance that has the smallest difference between the first and the second most probable labels (x1)

- Entropy Sampling
  - Selects the instance that has the highest entropy $E(x) = \sum - p(x) \log p(x)$

<table>
<thead>
<tr>
<th>Instance</th>
<th>Class A</th>
<th>Class B</th>
<th>Class C</th>
</tr>
</thead>
<tbody>
<tr>
<td>x1</td>
<td>0.5</td>
<td>0.3</td>
<td>0.2</td>
</tr>
<tr>
<td>x2</td>
<td>0.245</td>
<td>0.51</td>
<td>0.245</td>
</tr>
</tbody>
</table>
Semi-Supervised Learning
Do you want to waste your unlabelled data?

• Use labelled data and unlabelled data together to train classifier.
• Goal: learn a better prediction rule than based on labelled data alone.
• Known algorithms
  • Self-training
  • Generative models
  • Graph based methods
  • Co-Training
  • Semi-supervised SVM ...
Self training
The most naive algorithm

- Input: labelled data $L = \{x_i, y_i\}_{i=1}^{L}$, unlabelled data $U = \{x_j\}_{j=L+1}^{L+1}$
- Approach
  1. Train $f$ from $L$ using supervised learning.
  2. Apply $f$ to the unlabelled instances in $U$
  3. Remove a subset $S$ from $U$; add $\{(x, f(x)) \mid x \in S\}$ to $L$
- Limitation
  - Mistakes may introduce self-reinforcement
Good Example

(a) Iteration 1

(b) Iteration 25

(c) Iteration 74

(d) Final labeling of all instances
Bad Example
Structured Prediction
Predict a set of structured outcomes

• Sometimes, we need to predict multiple outcomes together.
• Need to consider the relation between outputs.
• Examples:
  • Sequence tagging (Part of speech)
  • Machine translation
  • Multi-objects recognition
Part of Speech Tagging

- Tagging words with the correct part of speech, such as noun, verb, pronoun, ...
- POS tags have a structure related to the syntax of sentence.
Conditional Random Field (CRF)

Chain structured CRF

\[
P(\overline{y}|\overline{x}; w) = \frac{\exp\left(\sum_i \sum_j w_j f_j(y_{i-1}, y_i, \overline{x}, i)\right)}{\sum_{\overline{y}' \in Y} \exp\left(\sum_i \sum_j w_j f_j(y_{i-1}', y_i', \overline{x}, i)\right)}
\]

Dependency between variables

Image source: http://www.davidsbatista.net/
Conclusion

• We formalize supervised learning framework with
  • Data, model, risk, optimization

• Statistical learning aims to discover how well our classifier work to compare with the optimal classifier.

• Generalized supervised learning incorporates
  • Learning to rank
  • Active learning
  • Semi-supervised learning
  • Structured prediction