Online Passive-Aggressive Algorithms

Tirgul 11
Multi-Label Classification
Multilabel Problem: Example

• Mapping Apps to smart folders:

Assign an installed app to one or more folders
Goal

• Given $A$, an installed app:

• Assign it to $\mathcal{y} \subseteq \mathcal{Y}$ the set of relevant folders:

• Where $\mathcal{Y}$ is the set of all folders:
Multilabel Classification

• A variant of the classification problem:
  • Multiple target labels must be assigned to each instance.

• Setting:
  • There are $k$ different possible labels: $Y = \{1, \ldots, k\}$.
  • Every instance $x_i$ is associated with a set of relevant labels $y_i$.

• Special case:
  • There is a single relevant label for each instance: multiclass (single-label) classification.
Multilabel Classification

• Other usage:
  • Text categorization:
    • $x_i$ represents a document.
    • $y_i$ is the set of topics which are relevant to the document
      • Chosen from a predefined collection of topics.
    • E.g.: A text might be about any of religion, politics, finance or education at the same time or none of these.
Task

Assign user’s installed applications to **one or more** smart folders in an automatic way
Task

• Training set of examples:

\[ S = \{ A_1, y_1 \}, \ldots, (A_m, y_m) \}\]

• Each example \((A_i, y_i)\) contains an application and a set of one or more smart folders:
Find a function $y = f_w(A)$ with parameters $w$ that assigns a set of folders $y$ to an installed app $A$. 
Model and Inference

\[ \hat{y} = f_w(A) = \text{argsort}_y \mathbf{w} \phi(A, y) \]
Feature maps

Google Play’s category

TF-IDF of title

TF-IDF of description

Representation of related apps
Inference

\[ \hat{y}_i = \text{argsort}_y w \phi(A_i, y) \]

- Algorithm’s output upon receiving an instance \( x_i \):
  - A score for each of the \( k \) labels in \( y \).

\[ y = \{ \text{Social, Music, Games, Photography, Shopping} \} \]

\[ \{ 0.6, 0.3, 0.7, 0.1, 0.01 \} \]
Inference

• The algorithm’s prediction is a vector in $\mathbb{R}^k$ where each element in the vector corresponds to the score assigned to the respective label.

• This form of prediction is often referred to as *label ranking*.

\[ \hat{y} = \text{argsort}_y w \phi(A, y) \]

\[ y = \{ \text{Social, Music, Games, Photography, Shopping} \} \]

\[ \{ 0.6, 0.3, 0.7, 0.1, 0.01 \} \in \mathbb{R}^k \]
Inference

• For a pair of labels $r, s \in \mathcal{Y}$:
  • If $\text{score}(r) > \text{score}(s)$:
    • Label $r$ is ranked higher than label $s$.

• Goal of Algorithm:
  • Rank every relevant label above every irrelevant label.
The Margin

• Example: \((A_i, y_i) = (\text{Games}, \text{Social})\).
• After making predictions \(\hat{y}_i\), the algorithm receives the correct set \(y_i\).
  1. Find the least probable correct folder:
     \[
     r = \arg \min_{r \in y_i} w \cdot \phi(A_i, r)
     \]
  2. Find the most probable wrong folder:
     \[
     s = \arg \max_{s \notin y_i} w \cdot \phi(A_i, s)
     \]
Iterate over examples

• Example: \((A_i, y_i) = (\text{Games}, \text{Social}).\)

• After making predictions \(\hat{y}_i\), the algorithm receives the correct set \(y_i\).

\[
r = \arg \min_{r \in y_i} w \cdot \phi(A_i, r)
\]

\[
s = \arg \max_{s \notin y_i} w \cdot \phi(A_i, s)
\]

• Update:

\[
w \leftarrow w + \tau \left( \phi(A_i, r) - \phi(A_i, s) \right)
\]
The Margin

• We define the margin attained by the algorithm on round $i$ for example $(x_i, y_i)$,

$$\gamma(w_i, (x_i, y_i)) = \min_{r \in y_i} w_i \phi(x_i, r) - \max_{s \notin y_i} w_i \phi(x_i, s).$$
The Margin

• The margin is positive if all relevant labels are ranked higher than all irrelevant labels.

• We are not satisfied with only a positive margin; we require the margin of every prediction to be at least 1.

\[ \gamma(w_i, (x_i, y_i)) = \min_{r \in y_i} w_i \phi(x_i, r) - \max_{s \notin y_i} w_i \phi(x_i, s). \]
The Loss

\[ \gamma(w_i, (x_i, y_i)) = \min_{r \in y_i} w_i \phi(x_i, r) - \max_{s \in y_i} w_i \phi(x_i, s). \]

- Define a hinge loss:

\[
\ell(w_i, (x_i, y_i)) = \begin{cases} 
0 & \gamma(w_i, (x_i, y_i)) \geq 1 \\
1 - \gamma(w_i, (x_i, y_i)) & \text{otherwise}
\end{cases}
\]
The Loss

\[ \ell(w_i, (x_i, y_i)) = \begin{cases} 
0 & \gamma(w_i, (x_i, y_i)) \geq 1 \\
1 - \gamma(w_i, (x_i, y_i)) & \text{otherwise}
\end{cases} \]

• Could also be written as follows:

\[ \ell(w_i, (x_i, y_i)) = [1 - \gamma(w_i, (x_i, y_i))]_+ \]

• where \([a]_+ = \max(0, a)\)
Learning

• First approach:

• Goal:

\[ w^* = \arg\min_w \mathbb{E} [\delta [f_w(A) \neq y]] \]

\[ w^* = \arg\min_w \frac{1}{m} \sum_{i=1}^{m} \left[ 1 - \min_{r \in y_i} w \cdot \phi(A_i, r) + \max_{s \notin y_i} w \cdot \phi(A_i, s) \right] + \frac{\lambda}{2} \|w\|^2 \]
Multilabel PA Optimization Problem

• An alternative approach:

\[ r_i = \text{argmin}_{r \in y_i} w_i \phi(x_i, r) \quad \text{and} \quad s_i = \text{argmax}_{s \not\in y_i} w_i \phi(x_i, s) \]

\[ w_{i+1} = \text{argmin}_w \frac{1}{2} \| w - w_i \|^2 \]

\[ \text{s.t.} \quad \ell(w, (x_i, y_i)) = \left[ 1 - \gamma(w, (x_i, y_i)) \right]^+ = 0 \]

\[ \gamma(w_i, (x_i, y_i)) = w_i \phi(x_i, r_i) - w_i \phi(x_i, s_i) \]

i.e., the margin is greater than 1.
Passive Aggressive (PA)

• The algorithm is passive whenever the hinge-loss is zero:
  • $\ell_i = 0 \rightarrow w_{i+1} = w_i$

• When the loss is positive, the algorithm aggressively forces $w_{i+1}$ to satisfy the constraint $\ell(w_i, (x_i, y_i)) = 0$, regardless of the step size required.
Passive Aggressive

• Solving with Lagrange Multipliers, we get the following update rule:

\[ w_{i+1} = w_i + \tau_i (\phi(x_i, r_i) - \phi(x_i, s_i)) \]

\[ \tau_i = \frac{\ell_i}{\|\phi(x_i, r_i) - \phi(x_i, s_i)\|^2} \]
Passive Aggressive

• The updated vector $w_{i+1}$ will classify example $x_i$ with $\ell(w, (x_i, y_i)) = 0$. 
Ranking
Ranking Problem: Example

• A Prediction Bar:

Predict the apps the user is most likely to use at any given time and location.
assume the user is at a context \( \mathbf{x} = (x_1, \ldots, x_n) \)

\[
\begin{align*}
    x_1 &= 12:47 \\
    x_2 &= \text{at the office} \\
    x_3 &= \text{today is Wednesday} \\
    \vdots & \quad \\
    \hat{A}^1 &= \text{the most likely app} \\
    \vdots & \quad \\
    \hat{A}^4 &= \text{the 4th likely app}
\end{align*}
\]
The goal is to find a function $f$ that gets as input the current context $\mathbf{x}$ and predicts the apps the user is likely to click at a given context.

$$\hat{A}^1, \hat{A}^2, \hat{A}^3, \hat{A}^4 = f(\mathbf{x})$$
Features

- **time of day**
  \[ \sigma_{\text{time-of-day}} = 10 \text{ min} \]

- **day of week**
  \[ \sigma_{\text{day-of-week}} = 1 \text{ day} \]

- **location**
  \[ \sigma_{\text{location}} = 100 \text{ meters} \]
Discriminative Model

- Prediction:
  \[ \hat{A}^1 = \arg \max_A f_w(x, A) \]
  \[ \hat{A}^2 = \arg \max_{A \neq \hat{A}^1} f_w(x, A) \]
  \[ \vdots \]
New Evaluation

• The performance of the prediction system is measured by Receiver operating Characteristics (ROC) curve.

true positive rate = \[
\frac{\text{app } A \text{ was in the prediction bar and was clicked}}{\text{total contexts where } A \text{ was clicked}}
\]

false positive rate = \[
\frac{\text{app } A \text{ was in the prediction bar and wasn’t clicked}}{\text{total contexts where } A \text{ wasn’t clicked}}
\]
Maximizing AUC

• By definition of the AUC (Bamber, 1975; Hanley and McNeil, 1982):

$$AUC = \mathbb{P} \left[ f(x^+, A) > f(x^-, A) \right]$$

$$f(x, A) = w^A \cdot x$$
Pairwise Dataset

• For an application $A$ define two sets of context:

$$\mathcal{X}_A^+$$

$$\mathcal{X}_A^-$$
Maximizing AUC

• By definition of the AUC (Bamber, 1975; Hanley and McNeil, 1982):

$$A = \mathbb{P} \left[ f(x^+, A) > f(x^-, A) \right]$$

$$w^* = \arg \max_w \mathbb{P} \left[ f_w(x^+, A) > f_w(x^-, A) \right]$$
$$= \arg \max_w \mathbb{E} \left[ \delta \left[ f_w(x^+, A) > f_w(x^-, A) \right] \right]$$
\[ w^* = \arg\max_w \mathbb{E} \left[ \delta \left[ f_w(x^+, A) > f_w(x^-, A) \right] \right] \]

\[ w^* = \arg\min_w \frac{1}{m} \sum_{i=1}^{m} \delta \left[ f_w(x_i^+, A_i) < f_w(x_i^-, A_i) \right] + \frac{\lambda}{2} \|w\|^2 \]

\[ w^* = \arg\min_w \frac{1}{m} \sum_{i=1}^{m} \left[ 1 - f_w(x_i^+, A_i) + f_w(x_i^-, A_i) \right] + \frac{\lambda}{2} \|w\|^2 \]
Solve using PA

• Set the next $w_i$ to be the minimizer of the following optimization problem

$$\min_{w \in \mathbb{R}^d, \xi \geq 0} \frac{1}{2} \| w - w_{i-1} \|^2$$

$$s.t. \quad f_w(x_i^+, A_i) - f_w(x_i^-, A_i) \geq 1$$
Implementation

• Online algorithm to solve the optimization problem efficiency on huge data.

• Theoretical guarantees of convergence.