Adaptive Two-View Online Learning for Math Topic Classification

Tam T. Nguyen[1], Kuiyu Chang[2], and Siu Cheung Hui[3]

Nanyang Technological University,
50 Nanyang Avenue, Singapore 639798
Email: [1]nguy0080@e.ntu.edu.sg, [2]askychang@ntu.edu.sg, [3]asschui@ntu.edu.sg

Abstract. Text categorization has been a popular research topic for years and has become more or less a practical technology. However, there exists little research on math topic classification. Math documents contain both textual data and math expressions. The text and math can be considered as two related but different views of a math document. The goal of online math topic classification is to automatically categorize a math document containing both mathematical expressions and textual content into an appropriate topic without the need for periodically retraining the classifier. To achieve this, it is essential to have a two-view online classification algorithm, which deals with the textual data view and the math expression view at the same time. In this paper, we propose a novel adaptive two-view online math document classifier based on the Passive Aggressive (PA) algorithm. The proposed approach is evaluated on real world math questions and answers from the Math Overflow question answering system. Compared to the baseline PA algorithm, our method's overall F-measure is improved by up to 3%. The improvement of our algorithm over the plain math expression view is almost 6%.

1 Introduction

Math documents such as math questions, scientific papers, etc., constitute a substantial portion of modern scientific literature. To organize these materials for easy retrieval, they are usually classified into pre-defined categories via automatic topic classifiers. Online documents such as blog posts, question and answer posts, emails, etc., are growing at extreme speeds. Manual classification is time consuming and expensive, to say the least. Over the years, machine-based topic classification has become matured enough to be deployed practically for this task. However, existing methods only focus on the textual data, which typically overwhelms underlying semantics like math expressions, tables, charts, diagrams, etc. In an attempt to improve the performance of classifiers on rich math content documents, we propose a novel approach for math-aware topic classification.

Clearly, it is trivial to regard math expressions as normal text data during tokenization, and treat the entire math document just like a regular text document using conventional text document classification methods. However, this approach basically ignores math expressions, which are highly structured data

containing valuable hints. As such, math expression semantics should be extracted using math-aware methods. Ideally, we should treat text and math as two distinct feature sets or views. To classify math documents based on text and math features, we need a suitable algorithm that can work on the two kinds of data at the same time without one view dominating the other. SVM-2K [11] and Two-view SVM [17] can be applied in this case. However, in dynamic systems such as online question answering sites, where new data is generated continuously, an online/incremental classification algorithm is more desirable. Such a system can predict the topic of each posting and adjust dynamically (if the automatic prediction is deemed wrong by the user). In this case, an online learning algorithm such as Perceptron [2, 21], Second-order Perceptron [4], or Passive Aggressive (PA) [8] can be considered. However, these algorithms only work on a single view and therefore cannot be applied directly to two-view data.

User postings in question answering systems such as *Cross Validated*[1], *Meta Optimize*[2], and *Math Overflow*[3], etc., typically contain many math expressions. While Cross Validated and Meta Optimize are mainly used by computer scientists, Math Overflow users include serious mathematicians. Moreover, one common characteristic of postings on all three sites is their rich math content. To automatically classify these content, we need a fast online learning algorithm that can work on two kinds of features, textual data and math expressions.

2 Related Work

Document topic classification aims to automatically categorize a given document into the appropriate topics or classes. Common classification algorithms include Naïve Bayes [15, 16], Nearest-Neighbors [6], C4.5 [22], Support Vector Machine (SVM) [5], etc. Document topic classification has been applied to many domains, e.g., emails, blogs, and news articles.

For online document topic classification, many algorithms have been proposed. The Perceptron algorithm [2, 21] is simple and fast but its classification accuracy is not good enough. To improve the performance of the Perceptron algorithm, Cesa-Bianchi et al. [4] proposed the Second-order Perceptron (SOP) algorithm, which takes advantage of second-order information; it performs better than the original Perceptron in terms of accuracy but is slower due to the added complexity required to estimate the covariance.

Later, Crammer et al. [8] proposed another Perceptron-based algorithm called the Passive Aggressive (PA) algorithm [8], which uses modern margin maximization learning. The PA algorithm performs better than both the original and Second-order Perceptrons. Nevertheless, the PA algorithm only works on single view datasets. Similar algorithms that improved upon the PA algorithm include the Passive-Aggressive Mahalanobis [19], Confidence-Weight (CW) Linear Clas-

[1] http://stats.stackexchange.com/
[2] http://metaoptimize.com/qa/
[3] http://mathoverflow.net/questions

sification [10], and CW algorithm for multi-class classification [9]. In this paper, we derive a two-view adaptive version of the PA algorithm.

3 Math Topic Classification

3.1 Math Document

Math documents contain not only textual data, but also math expressions. Math expressions embody abstract mathematical semantics via math symbols and structures. In math documents, math expressions are presented in ASCII or markup formats, e.g., LaTeX, ASCIIMath [12], OMDoc [13], OpenMath [3], and MathML [1]. Among these, the LaTeX markup language has been used by many researchers for more than 40 years. We choose LaTeX as the raw storage format for embedding math expressions in math documents, since it requires less memory compared to other markup languages. For example, the raw format of a posting from Math Overflow is given in Listing 1.1, where math expressions are enclosed between the $ symbols.

Listing 1.1. An Example Question

```
Title:   Why isn't Likelihood a Probability Density Function?
Content: I've been trying to get my head around why a likelihood isn't a
    probability density function. My understanding says that for an event
     X and a model parameter m:
        $P(X|m)$ is a probability density function
        $P(m|X)$ is not...
    It feels like it should be, and I can't find a clear explanation of
        why it's not. Does it also mean that a Likelihood can take a
        value greater than $1$?
```

Different from textual descriptions, math expressions cannot be simply encoded as an unordered sequence of tokens due to its rich math semantics, which includes functions, operators, variables, constant numbers, etc. As such, we propose a novel feature extraction method to convert math expressions into math features, as described in the following section.

3.2 Math Feature Extraction

Due to the existence of both semantic and structural information, the preprocessing step for math expressions is more complex than that of text documents. For the purpose of math topic classification, math features should be representative enough to reflect the underlying characteristics of each math topic. To do that, we perform the following steps:

- Content MathML conversion. We convert the retrieved LaTeX math expressions into Content MathML format.
- Math feature extraction. From the Content MathML data, we can extract math features by traversing the MathML tree.

To convert math expressions from LaTeX, we use the SnuggleTeX library[4]. We first convert the math expressions from LaTeX to the representation MathML format, then we use cascading stylesheets to map the representation MathML to content MathML. MathML is selected over LaTeX for its rich semantics and ease of processing via standard XML libraries. Listing 1.2 shows an example of content MathML for the math expression $(x + y)^2$.

For content MathML data, we use the XML tree traversal approach for extracting math features. In this research, we only use two kinds of features, single features and combination features. The single features are used to express constant numbers, variable names, functions names, etc. Combination features are the combinations of math operators and operands in the math expressions.

Listing 1.2. *MathML* Content Markup of $(x + y)^2$

```
<apply>                 % apply operator
   <power/>             % power operator
   <mfence>
      <apply>           % apply operator
         <plus/>        % plus operator
         <ci>x</ci>     % variable (ci) x
         <ci>y</ci>     % variable (ci) y
      </apply>
   </mfence>
   <cn>2</cn>           % constant (cn) 2
</apply>
```

Take the sub-expression $x + y$ in Listing 1.2 as an example; based on the content MathML data, we have two single features (ci)x and (ci)y, where ci stands for a variable and (ci)x stands for a variable named x. We also have one combination feature (plus)(ci)x(ci)y, where plus stands for the operator + and (plus)(ci)x(ci)y denotes the operator + applied to two operands x and y.

3.3 Supervised Key Phrase Extraction

In math document classification, key phrases play a crucial role in discriminating math documents. Take "Wishart distribution" and "topological vector space" as examples, "Wishart distribution" tends to appear in documents related to statistics. On the other hand, "topological vector space" is a very common phrase in linear algebra. Therefore, in this research, we adapt the Natural Language Toolkit (NLTK)[5] to extract noun phrases from math documents. Then, the Jensen-Shannon (JS) divergence [7] is used to weight each noun phrase as follows:

$$JS(p_1, \ldots, p_C) = -(\sum_{i=1}^{C} \pi_i p_i) \log(\sum_{i=1}^{C} \pi_i p_i) + \sum_{i=1}^{C} \pi_i p_i \log(p_i) \quad (1)$$

[4] http://www2.ph.ed.ac.uk/snuggletex
[5] http://www.nltk.org/

where p_i is the probability of the phrase appearing in class i, $\sum_i \pi_i = 1$ over all C classes, and $\pi_i > 0$. JS is zero for phrases that appear uniformly over all classes, and maximized (at 1) for phrases appearing only in one class.

3.4 Online Learning Classification

Given a classification task of two classes (negative -1 and positive $+1$) and an unknown pattern represented by feature vector $\mathbf{x}$. The goal is to learn the weight $\mathbf{w}$ of a linear prediction function $f(\mathbf{x}) = \text{sign}(\mathbf{w} \cdot \mathbf{x})$. The online learning algorithm operates in rounds, as input data arrives sequentially. Let $x_t \in \mathbb{R}^n$ be an example arriving at round t. The algorithm predicts its label $\hat{y}_t \in \{-1, +1\}$, after which it receives the true label. If its prediction is correct, the learning process proceeds to the next round. Otherwise, it suffers a loss $\ell(y_t, \hat{y}_t)$, and updates its weight $\mathbf{w}$ accordingly. The loss can be modeled using the hinge-loss function, which equals to zero when the margin exceeds 1, as follows.

$$\ell(\mathbf{w}_t; (\mathbf{x}_t, y_t)) = \begin{cases} 0 & \text{if} \quad y_t(\mathbf{w}_t \cdot \mathbf{x}_t) \geq 1 \\ 1 - y_t(\mathbf{w}_t \cdot \mathbf{x}_t) & \text{otherwise} \end{cases} \tag{2}$$

Crammer et al. [8] formulated three optimization problems; one based on hard margin and two using soft margins, which are named PA, PA-I, and PA-II respectively, as follows.

$$\begin{aligned} \mathbf{w}_{t+1} &= \underset{\mathbf{w} \in \mathbb{R}^n}{\text{argmin}} \frac{1}{2} \parallel \mathbf{w} - \mathbf{w}_t \parallel^2 && \text{(PA)} \\ &\text{s.t.} \quad \ell(\mathbf{w}; (\mathbf{x}_t, y_t)) = 0. \\ \mathbf{w}_{t+1} &= \underset{\mathbf{w} \in \mathbb{R}^n}{\text{argmin}} \frac{1}{2} \parallel \mathbf{w} - \mathbf{w}_t \parallel^2 + C\xi && \text{(PA-I)} \\ &\text{s.t.} \quad \ell(\mathbf{w}; (\mathbf{x}_t, y_t)) \leq \xi; \xi \geq 0. \\ \mathbf{w}_{t+1} &= \underset{\mathbf{w} \in \mathbb{R}^n}{\text{argmin}} \frac{1}{2} \parallel \mathbf{w} - \mathbf{w}_t \parallel^2 + C\xi^2 && \text{(PA-II)} \\ &\text{s.t.} \quad \ell(\mathbf{w}; (\mathbf{x}_t, y_t)) \leq \xi. \end{aligned} \tag{3}$$

Intuitively, the new weight $\mathbf{w}_{t+1}$ should be close to the old weight $\mathbf{w}_t$ while minimizing the loss $\ell(\mathbf{w}; (\mathbf{x}_t, y_t))$. Solving the above problems, they obtained the weight update equation as follows:

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \tau_t y_t \mathbf{x}_t$$

where the coefficient τ_t has one of the following three forms:

$$\tau_t = \frac{1 - y_t(\mathbf{w}_t \cdot \mathbf{x}_t)}{\parallel \mathbf{x}_t \parallel^2} \text{ (PA)}, \; \tau_t = \min\left\{C, \frac{1 - y_t(\mathbf{w}_t \cdot \mathbf{x}_t)}{\parallel \mathbf{x}_t \parallel^2}\right\} \text{ (PA-I), and}$$
$$\tau_t = \frac{1 - y_t(\mathbf{w}_t \cdot \mathbf{x}_t)}{\parallel \mathbf{x}_t \parallel^2 + \frac{1}{2C}} \text{ (PA-II).}$$

For the two-view online learning setting, training data are triplets $(\mathbf{x}_t^A, \mathbf{x}_t^B, y_t) \in \mathbb{R}^n \times \mathbb{R}^m \times [-1, +1]$, which arrive in sequence where $\mathbf{x}_t^A \in \mathbb{R}^n$ is the first view

vector, $\mathbf{x}_t^B \in \mathbb{R}^m$ is the second view vector, and y_t is their common label. Since we don't know which view is more important than the other, the coupled weights $(\mathbf{w}_t^A, \mathbf{w}_t^B)$ should be learnt based on the weighted *hybrid model* [20] as follows:

$$f(\mathbf{x}_t^A, \mathbf{x}_t^B) = \text{sign}\Big(\eta \mathbf{w}_t^A \cdot \mathbf{x}_t^A + (1-\eta)\mathbf{w}_t^B \cdot \mathbf{x}_t^B\Big)$$

where $\eta \in (0,1)$ is used to adjust the importance of the two views.

Let $g(\mathbf{x}_t^A, \mathbf{x}_t^B) = \eta \mathbf{w}_t^A \cdot \mathbf{x}_t^A + (1-\eta)\mathbf{w}_t^B \cdot \mathbf{x}_t^B$. To incorporate the new model into the algorithm, we define the loss function as follows:

$$\ell((\mathbf{w}_t^A, \mathbf{w}_t^B); (\mathbf{x}_t^A, \mathbf{x}_t^B, y_t)) = \begin{cases} 0 & \text{if} \quad y_t g(\mathbf{x}_t^A, \mathbf{x}_t^B) \geq 1 \\ 1 - y_t g(\mathbf{x}_t^A, \mathbf{x}_t^B) & \text{otherwise} \end{cases} \tag{4}$$

Relationship between Views To determine the relatedness between the two views, we define a disagreement factor as follows:

$$|\eta \mathbf{w}_t^A \cdot \mathbf{x}_t^A - (1-\eta)\mathbf{w}_t^B \cdot \mathbf{x}_t^B| \tag{5}$$

where $|\cdot|$ denotes the absolute function and η, similar to the hybrid model, is used to trade off the disagreement between the two views. The objective is to minimize the disagreement between the two views.

Adaptive Two-view Passive Aggressive Algorithm The ideal objective function should include both the new loss function in (4) and the view relatedness factor in (5). Similar to the PA algorithm, the new weights of the two-view learning algorithm are updated based on the optimization problem as follows:

$$\begin{aligned}(\mathbf{w}_{t+1}^A, \mathbf{w}_{t+1}^B) \quad = \underset{(\mathbf{w}^A, \mathbf{w}^B) \in \mathbb{R}^n \times \mathbb{R}^m}{\text{argmin}} & \frac{1}{2} \parallel \mathbf{w}^A - \mathbf{w}_t^A \parallel^2 + \frac{1}{2} \parallel \mathbf{w}^B - \mathbf{w}_t^B \parallel^2 \\ & + \gamma |\eta y_t \mathbf{w}^A \cdot \mathbf{x}_t^A - (1-\eta) y_t \mathbf{w}^B \cdot \mathbf{x}_t^B| + C\xi \end{aligned}$$

$$\text{s.t.} \quad 1 - y_t g(\mathbf{x}_t^A, \mathbf{x}_t^B) \leq \xi; \quad \xi \geq 0\,.$$

where γ and C are positive agreement and aggressiveness parameters respectively. While γ is used to adjust the importance of the agreement between the two views, C is used to control the aggressiveness property of the PA algorithm. Note that the multiplier y_t in the agreement is there to simplify subsequent derivations.

For the absolute function, we have

$$\begin{aligned}|\eta y_t \mathbf{w}^A \cdot \mathbf{x}_t^A - (1-\eta) y_t \mathbf{w}^B \cdot \mathbf{x}_t^B| = \max \Big(& \eta y_t \mathbf{w}^A \cdot \mathbf{x}_t^A - (1-\eta) y_t \mathbf{w}^B \cdot \mathbf{x}_t^B, \\ & (1-\eta) y_t \mathbf{w}^B \cdot \mathbf{x}_t^B - \eta y_t \mathbf{w}^A \cdot \mathbf{x}_t^A \Big)\end{aligned}$$

Suppose $z = |\eta y_t \mathbf{w}^A \cdot \mathbf{x}_t^A - (1-\eta) y_t \mathbf{w}^B \cdot \mathbf{x}_t^B|$, the above optimization problem can be expressed as follows:

$$(\mathbf{w}_{t+1}^A, \mathbf{w}_{t+1}^B) = \underset{(\mathbf{w}^A, \mathbf{w}^B) \in \mathbb{R}^n \times \mathbb{R}^m}{\operatorname{argmin}} \frac{1}{2} \parallel \mathbf{w}^A - \mathbf{w}_t^A \parallel^2 + \frac{1}{2} \parallel \mathbf{w}^B - \mathbf{w}_t^B \parallel^2 + \gamma z + C\xi$$

$$\begin{aligned} \text{s.t.} \qquad & 1 - y_t g(\mathbf{x}_t^A, \mathbf{x}_t^B) \leq \xi; \quad \xi \geq 0; \\ & z \geq \eta y_t \mathbf{w}^A \cdot \mathbf{x}_t^A - (1-\eta) y_t \mathbf{w}^B \cdot \mathbf{x}_t^B; \\ & z \geq (1-\eta) y_t \mathbf{w}^B \cdot \mathbf{x}_t^B - \eta y_t \mathbf{w}^A \cdot \mathbf{x}_t^A. \end{aligned} \tag{6}$$

Remark When the value of the objective function is small, the disagreement factor z of the two views is forced to be small. If we set $\gamma = 0$, the problem reduces to learning two independent linear models; if we set $\gamma = 1$, we aggressively penalize any view disagreements. By adjusting the value of $0 < \gamma < 1$, we can control the amount of collaboration between the two views.

Proposition 1 *The optimization problem (6) has the following close form solution:*

$$\mathbf{w}^A = \mathbf{w}_t^A - \eta(\alpha - \beta - \tau) y_t \mathbf{x}_t^A$$

and

$$\mathbf{w}^B = \mathbf{w}_t^B - (1-\eta)(\beta - \alpha - \tau) y_t \mathbf{x}_t^B$$

where

$$\tau = \min \Big\{ C, \frac{(\alpha - \beta)\Big(\eta^2 \parallel \mathbf{x}_t^A \parallel^2 - (1-\eta)^2 \parallel \mathbf{x}_t^B \parallel^2 \Big) + \ell_t}{\eta^2 \parallel \mathbf{x}_t^A \parallel^2 + (1-\eta)^2 \parallel \mathbf{x}_t^B \parallel^2} \Big\},$$

$$\alpha = \min \Big\{ \gamma, \frac{1}{2}\Big(\gamma + \frac{1}{\eta} \frac{y_t \mathbf{w}_t^A \cdot \mathbf{x}_t^A}{\parallel \mathbf{x}_t^A \parallel^2} - \frac{1}{1-\eta} \frac{y_t \mathbf{w}_t^B \cdot \mathbf{x}_t^B}{\parallel \mathbf{x}_t^B \parallel^2} \Big) \Big\}, \text{ and}$$

$$\beta = \min \Big\{ \gamma, \frac{1}{2}\Big(\gamma - \frac{1}{\eta} \frac{y_t \mathbf{w}_t^A \cdot \mathbf{x}_t^A}{\parallel \mathbf{x}_t^A \parallel^2} + \frac{1}{1-\eta} \frac{y_t \mathbf{w}_t^B \cdot \mathbf{x}_t^B}{\parallel \mathbf{x}_t^B \parallel^2} \Big) \Big\}.$$

Finally, we obtain our Adaptive Two-view Passive Aggressive formulation as shown in Algorithm 1.

Getting Rid of Parameter η One limitation of the Two-view PA algorithm in [20] is that its view parameter η must be chosen beforehand. In practice, however, choosing a suitable value for this parameter can be tedious. In addition, the optimal value may change with time, thereby affecting the performance. Therefore, we propose an adaptive variant of the Two-view PA algorithm that automatically determines the best value of η. The idea is to modify the objective function of the optimization problem (6) by adding a new regularization factor $\frac{\zeta}{2}(\eta - \eta_t)^2$. The new optimization problem has no close form expression for η since α, β, and τ all depend on η. Without loss of generality, we assume that these variables only depend on the *previous* value of η, i.e., η_t.

Proposition 2 *Suppose that α, β, and τ are independent of η, the new optimization problem has an approximated close form solution as follows.*

$$\eta = \eta_t - \frac{1}{\zeta}\Big((\alpha - \beta - \tau) y_t \mathbf{w}^A \cdot \mathbf{x}_t^A - (\beta - \alpha - \tau) y_t \mathbf{w}^B \cdot \mathbf{x}_t^B \Big) \tag{7}$$

Initially the two views are treated equally, i.e., $\eta = 0.5$, and will be updated based on Equation (7) thereafter. This is thus an adaptive version of the Two-view PA algorithm, which we call the Adaptive Two-view PA algorithm.

Algorithm 1 Adaptive Two-view Passive Aggressive Algorithm

Input:
C = positive aggressiveness parameter
γ = positive agreement parameter
Output:
None
Process:
Initialize $\mathbf{w}_1^A \leftarrow \mathbf{0}$; $\mathbf{w}_1^B \leftarrow \mathbf{0}$; $\eta = 0.5$;
for $t = 1, 2, \ldots$ **do**
Receive instances $\mathbf{x}_t^A \in \mathbb{R}^n$ and $\mathbf{x}_t^B \in \mathbb{R}^m$
Predict $\hat{y}_t = \text{sign}(\eta \mathbf{w}_t^A \cdot \mathbf{x}_t^A + (1-\eta)\mathbf{w}_t^B \cdot \mathbf{x}_t^B)$
Receive correct label $y_t \in \{-1, +1\}$
Suffer loss
$\ell_t \leftarrow \max\left\{0, 1 - \eta y_t \mathbf{w}_t^A \cdot \mathbf{x}_t^A - (1-\eta) y_t \mathbf{w}_t^B \cdot \mathbf{x}_t^B\right\}$
if $\ell_t > 0$ **then**
Update $\mathbf{w}_t^A$ and $\mathbf{w}_t^B$ per Proposition 1
Update η per Proposition 2
end if
end for

Table 1. Summary of Datasets in Our Experiments

	View		Sample Count		
	Name	#Dim	#Pos	#Neg	#Total
Ads	img & dest	929	459	2820	3279
	alt & base	602			
Product Review	lexical	2759	1000	1000	2000
	formal	5			
WebKB	page	3000	230	821	1051
	link	1840			

4 Performance Evaluation

In this section, we evaluate the online classification performance of our proposed Adaptive Two-view PA algorithm on three benchmark datasets, Ads [14], Product Review [17], and WebKB [23], and a math dataset taken from the Math Overflow site. The *single-view* PA algorithm serves as the baseline. We employ a different PA model for each view, denoted as *PA View 1* and *PA View 2* for

each view, respectively. We also report the results from a simple concatenation of the input feature vectors from each view to form a larger feature set, denoted as *PA Cat*. We compare the performance of all algorithms based on F-measure instead of accuracy because most of the datasets are highly (class) imbalanced.

4.1 Two-view Learning Evaluation

In this section, we evaluate the proposed algorithm on three benchmark datasets. The dataset characteristics are listed in Table 1. We use cross validation to select the optimal value for all parameters C, γ, and η, so as to make the comparison fair and meaningful. Here, η is learnt using the Adaptive Two-view PA algorithm.

Table 2. F-Measure on 3 benchmark datasets

Dataset	PA View 1	PA View 2	PA Cat	Adaptive Two-view PA
Ads	83.41 ± 2.76	76.26 ± 2.21	81.95 ± 2.55	$\mathbf{84.96} \pm 2.41$
Product Review	86.69 ± 1.39	70.80 ± 1.57	86.68 ± 1.63	$\mathbf{88.72} \pm 1.80$
WebKB	93.76 ± 2.30	90.68 ± 2.70	95.56 ± 0.98	$\mathbf{98.02} \pm 2.14$

The Ads dataset was first used by Kushmerick [14] to automatically filter advertisement images from web pages. In our experiments, we used just four views, namely *image URL view*, *destination URL view*, *base URL view*, and *alt view*. Since we are limited to handling two views for each task, the first and second views were concatenated into View 1 and the remaining two views were concatenated into View 2. This dataset has 3279 examples, including 459 positive examples (ads), with the remaining samples negative (non-ads). Experimental learning results on the Ads dataset are shown in Table 2, which shows the proposed algorithm to have the best F-measure score, performing more than 1% better than the runner-up, PA View 1. Interestingly, PA Cat fared worse than PA View 1, which could be due to a noisy decision boundary in the space of PA View 2. This can also be seen by the marginal improvement of 1% of the adaptive Two-view PA results.

The Product Review dataset is crawled from popular online Chinese cellphone forums [17]. The dataset has 1000 true reviews and 1000 spam reviews. It comprises two sets of features: one based on review content (*lexical view*) and the other based on extracted characteristics of the review sentences (*formal view*). The experimental results on this dataset are shown in Table 2. Again, our Two-view adaptive PA performs better than the rest, beating the runner-up (PA View 1) by 2%. Here PA Cat performed better than either view alone, which is typically the case.

The WebKB course dataset has been frequently used in the empirical study of multi-view learning. It comprises 1051 web pages collected from the computer science departments of four universities. Each page has a class label, course or non-course. The two views of each page are the textual content of a web page

(*page view*) and the words that occur in the hyperlinks of other web pages pointing to it (*link view*), respectively. We used a processed version of the WebKB course dataset [23] in our experiment. The performance of PA Cat here is also better than the best single-view PA. And the Adaptive Two-view PA performed more than 2% better than PA Cat, and 4% better than the best individual view PA. Compared to the non-adaptive (equal weightage of both views) Two-view approach of [20], our adaptive Two-view approach performed similarly or better.

4.2 Math Topic Classification

We downloaded more than 30,000 math questions and answers belonging to 20 math categories such as algebraic geometry, number theory, algebraic topology, combinatorics, group theory, probability, etc., from the Math Overflow site, a popular math question answering system. Each math question or answer is treated as one math document, which may contain both text content and math expressions. To evaluate math topic classification, we choose two major categories (algebraic geometry and number theory), which comprises more than 7,200 math documents. After preprocessing, we obtain the following datasets.

- Text only. All math expressions are removed from math documents. The remaining text-only documents are transformed into a vector format using $tf \times idf$ weighting [18].
- Math only. To evaluate whether math expressions are useful for math topic classification, we extract all math expressions from each math document. Then we apply the math feature extraction method of Section 3.2 to generate the *math only* dataset.
- Raw. It is trivial to treat math expressions as normal text data. One can use the latest text preprocessing techniques to extract textual features from both math and text.
- Math and text. We store the *math only* and *text only* datasets as two datasets (views), called the *math & text* dataset.
- Key phrase. Math key phrases extracted using the method of Section 3.3.

PA Only After preprocessing, we then run the PA algorithm on all datasets using 5-fold cross validation. The experimental results are shown in column 2 of Table 3. Note that the PA model trained on Math & Text operates on a concatenation of the two views. We see that the PA algorithm performed the worst on the *math only* dataset and best on the *text only* dataset. Clearly, there is much room for improvement in our math expression extraction process.

For the *raw text* and *math & text* datasets, the F-measure of the PA algorithm is not high, although both math and text data are taken into consideration. In fact, the PA algorithm did very poorly on just the *raw text*. Its performance is only better than its *math only* dataset results. Compared with the *math only* and *raw text* datasets, the *math & text* dataset can improve the performance of the PA algorithm. However, its performance on this dataset is actually worse than the *text only* dataset.

The Missing View In practice, math documents do not always contain both text data and math expressions. So what happens if either text data or math expressions are available, but not both? Can Two-view PA work in this case? To find out, we trained the Two-view PA on the *text & math* dataset and tested it on the *text only* and *math only* datasets. It means that we trained the Two-view PA on both views to have two weight vectors and then used them to predict the labels for documents in individual views. While testing on one view, we will ignore the other view.

Table 3. F-Measure Comparison on Math Overflow Datasets (* trained on the Math & Text Dataset, with results for individual views shown)

Dataset	PA	Adaptive Two-view PA*
Text Only	72.73 ± 2.97	$\mathbf{73.85} \pm 2.90$
Math Only	56.31 ± 7.47	$\mathbf{64.25} \pm 6.05$
Raw	61.02 ± 0.12	-
Math & Text	68.91 ± 5.03	$\mathbf{75.70} \pm 3.37$
Key Phrase	76.78 ± 0.90	$\mathbf{78.15} \pm 1.27$

We also ran the Two-view PA algorithm on the *math & text* dataset (treating each as one view), whose 75.70% F-measure score is more than 6% better than the PA algorithm (68.91%), which was trained on the combined view. Moreover, compared with the PA on *text only* dataset (72.73%), the two view performance (75.70%) is improved by nearly 3%.

This means when user posts a math question containing math expressions but without text data, the Two-view PA algorithm performs better than the PA algorithm trained on the *math only* dataset by up to 6%. The results are encouraging because we are able to take advantage of the data of one view to improve the performance of the classifier on another view by enforcing agreement between the two views.

Similarly, for the *key phrase* dataset, the performance is improved by up to 4% compared to the *text only* dataset in the single view PA. If we train the Two-view PA with the *key phrase* and *math only* datasets, the Two-view performs better than the single view PA by nearly 2%.

Table 4. The Adaptive Two-view PA Results for All Datasets

Dataset	η	F-measure
Ads	0.342 ± 0.015	84.96 ± 2.41
Math Overflow	0.524 ± 0.001	75.70 ± 3.37
Product Review	0.795 ± 0.011	88.72 ± 1.80
WebKB	0.438 ± 0.001	98.02 ± 2.14

4.3 View Weight Parameter Learning

The average η value for the Adaptive Two-view PA algorithm for all datasets are listed in Table 4. Note that for all 3 datasets, View 1 performed better than View 2, individually. For the Ads and WebKB datasets, we note that $\eta < 0.5$ despite View 1 performing better than View 2. On the other hand, for the Math Overflow and Product Review datasets, we have $\eta > 0.5$. Therefore, we cannot rely solely on performance of individual views to determine the value of η. Generally, the better performing view does not automatically deserves a higher weightage. The Adaptive Two-view PA algorithm can solve this problem by adaptively updating η at each round of the learning process.

5 Conclusion

We proposed an Adaptive Two-view Passive Aggressive algorithm, which is able to take advantage of multiple views of data to achieve an improvement in overall classification performance. We formulated our learning framework into an optimization problem and derive a closed form solution. Making a simple assumption on the independence of other parameters on the weight parameter, we derived an approximated close-form update equation for the view mixing parameter.

We evaluated the proposed approach on practical applications such as product review classification, advertising image removal, and math topic classification. We also prepared a two-view Math Overflow dataset containing text and math expressions, which is useful for math topic classification since this is the first publicly available dataset of its kind.

Although in this research, we evaluated the proposed approach on math questions and answers, it can be applied in practice to deal with other kinds of math documents such as math questions in student books, scientific papers, etc. There remain some interesting open problems that warrant further investigations. First, the math feature extraction method should be investigated further because the performance based on math features only is not good enough. We would also like to extend the Two-view PA algorithm to handle multiple views and multiple classes. However, formulating a multi-view PA is non-trivial, as it involves defining multi-view relatedness and minimizing pairs of view agreements. Formulating a multi-class Two-view PA should be more feasible.

Acknowledgements

We thank the anonymous reviewers for their valuable comments and suggestions, especially one reviewer who pointed out the flaw in the original presentation of proposition 2. This research was supported in part by Singapore Ministry of Education's Academic Research Fund Tier 2 grant ARC 9/12 (MOE2011-T2-2-056).

References

1. R. Ausbrooks, S. Buswell, S. Dalmas, S. Devitt, A. Diaz, R. Hunter, B. Smith, N. Soiffer, R. Sutor, and S. Watt. Mathematical markup language (mathml) version 2.0, 2000.
2. H. Block. The perceptron: A model for brain functioning. *Rev. Modern Phys.*, 34:123–135, 1962.
3. S. Buswell, O. Caprotti, D. P. Carlisle, M. C. Dewar, M. Gaetano, and M. Kohlhase. *The Open Math standard version 2.0.* 2004.
4. N. Cesa-Bianchi, A. Conconi, and C. Gentile. A second-order perceptron algorithm. *Siam J. of Comm.*, 34, 2005.
5. C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.
6. T. Cover and P. Hart. Nearest neighbor pattern classification. 13:373–389, 2002.
7. T. M. Cover and J. A. Thomas. *Elements of information theory.* Wiley-Interscience, New York, NY, USA, 1991.
8. K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, pages 551–585, 2006.
9. K. Crammer, M. Dredze, and A. Kulesza. Multi-class confidence weighted algorithms. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 496–504, Singapore, August 2009. Association for Computational Linguistics.
10. M. Dredze, K. Crammer, and F. Pereira. Confidence-weighted linear classification. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pages 264–271, New York, NY, USA, 2008. ACM.
11. J. D. R. Farquhar, D. R. Hardoon, H. Meng, J. Shawe-Taylor, and S. Szedmk. Two view learning: Svm-2k, theory and practice. In *Proceedings of NIPS'05*, 2005.
12. P. Jipsen. Translating ascii math notation to mathml and graphics, 2007.
13. M. Kohlhase and I. Sucan. A search engine for mathematical formulae. In J. Calmet, T. Ida, and D. Wang, editors, *AISC '06: Proceedings of 8th International Conference on Artificial Intelligence and Symbolic Computation*, pages 241–253. Springer-Verlag, 2006.
14. N. Kushmerick. Learning to remove internet advertisements. In *Proceedings of the third annual conference on Autonomous Agents*, AGENTS '99, pages 175–181, New York, NY, USA, 1999. ACM.
15. P. Langley, W. Iba, and K. Thompson. An analysis of bayesian classifiers. In *AAAI'92: Proceedings of the tenth national conference on Artificial intelligence*, pages 223–228. AAAI Press, 1992.
16. D. D. Lewis. Naive (bayes) at forty: The independence assumption in information retrieval. In *ECML '98: Proceedings of the 10th European Conference on Machine Learning*, pages 4–15, London, UK, 1998. Springer-Verlag.
17. G. Li, S. C. H. Hoi, and K. Chang. Two-view transductive support vector machines. In *Proceedings of SDM'10*, pages 235–244, 2010.
18. C. D. Manning, P. Raghavan, and H. Schtze. *Introduction to Information Retrieval.* Cambridge University Press, 2008.
19. T. T. Nguyen, K. Chang, and S. C. Hui. Distribution-aware online classifiers. In T. Walsh, editor, *IJCAI*, pages 1427–1432. IJCAI/AAAI, 2011.
20. T. T. Nguyen, K. Chang, and S. C. Hui. Two-view online learning. In *PAKDD (1)*, pages 74–85, 2012.

21. A. Novikoff. On convergence proofs of perceptrons. In *Proceedings of the Symposium on the Mathematical Theory of Automata*, volume 7, pages 615–622, 1962.
22. J. R. Quinlan and R. L. Rivest. Inferring decision trees using the minimum description length principle. *Inf. Comput.*, 80(3):227–248, 1989.
23. V. Sindhwani, P. Niyogi, and M. Belkin. Beyond the point cloud: from transductive to semi-supervised learning. In *Proceedings of the 22nd international conference on Machine learning*, ICML '05, pages 824–831, New York, NY, USA, 2005. ACM.

A Proof of Proposition 1

To prove the Proposition 1, we first define the Lagrangian of the optimization problem as follows:

$$\begin{aligned}\mathcal{L} = & \frac{1}{2} \parallel \mathbf{w}^A - \mathbf{w}_t^A \parallel^2 + \frac{1}{2} \parallel \mathbf{w}^B - \mathbf{w}_t^B \parallel^2 + \gamma z + C\xi - \lambda\xi \\ & + \tau\Big(1 - \xi - \eta y_t \mathbf{w}^A \cdot \mathbf{x}_t^A - (1-\eta) y_t \mathbf{w}^B \cdot \mathbf{x}_t^B\Big) \\ & + \alpha\Big(\eta y_t \mathbf{w}^A \cdot \mathbf{x}_t^A - (1-\eta) y_t \mathbf{w}^B \cdot \mathbf{x}_t^B - z\Big) \\ & + \beta\Big((1-\eta) y_t \mathbf{w}^B \cdot \mathbf{x}_t^B - \eta y_t \mathbf{w}^A \cdot \mathbf{x}_t^A - z\Big) \\ = & \frac{1}{2} \parallel \mathbf{w}^A - \mathbf{w}_t^A \parallel^2 + \frac{1}{2} \parallel \mathbf{w}^B - \mathbf{w}_t^B \parallel^2 + (\gamma - \alpha - \beta) z + (C - \lambda - \tau)\xi \\ & + \eta(\alpha - \beta - \tau) y_t \mathbf{w}^A \cdot \mathbf{x}_t^A + (1-\eta)(\beta - \alpha - \tau) y_t \mathbf{w}^B \cdot \mathbf{x}_t^B + \tau\end{aligned} \tag{8}$$

where α, β, τ, and λ are positive Lagrangian multipliers.

Setting the partial derivatives of $\mathcal{L}$ with respect to the weight $\mathbf{w}^A$ to zero, we have

$$0 = \frac{\partial \mathcal{L}}{\partial \mathbf{w}^A} = \mathbf{w}^A - \mathbf{w}_t^A + \eta(\alpha - \beta - \tau) y_t \mathbf{x}_t^A \Rightarrow \mathbf{w}^A = \mathbf{w}_t^A - \eta(\alpha - \beta - \tau) y_t \mathbf{x}_t^A \tag{9}$$

Similarly, for the other view we have

$$\mathbf{w}^B = \mathbf{w}_t^B - (1-\eta)(\beta - \alpha - \tau) y_t \mathbf{x}_t^B \tag{10}$$

Setting the partial derivatives of $\mathcal{L}$ with respect to weight z to zero, we have

$$0 = \frac{\partial \mathcal{L}}{\partial z} = (\gamma - \alpha - \beta) \Rightarrow \alpha + \beta = \gamma \tag{11}$$

Setting the partial derivatives of $\mathcal{L}$ with respect to weight ξ to zero, we have

$$0 = \frac{\partial \mathcal{L}}{\partial \xi} = (C - \lambda - \tau) \Rightarrow \lambda + \tau = C \tag{12}$$

Note that $\lambda \geq 0$, so we can conclude that $0 \leq \tau \leq C$.

Substituting (9), (10), (11), and (12) into (8), we have

$$\begin{aligned}\mathcal{L} &= \frac{1}{2}\eta^2(\alpha-\beta-\tau)^2 \parallel \mathbf{x}_t^A \parallel^2 + \frac{1}{2}(1-\eta)^2(\beta-\alpha-\tau)^2 \parallel \mathbf{x}_t^B \parallel^2 \\ &\quad +\eta(\alpha-\beta-\tau)y_t\Big(\mathbf{w}_t^A - \eta(\alpha-\beta-\tau)y_t\mathbf{x}_t^A\Big)\cdot\mathbf{x}_t^A \\ &\quad +(1-\eta)(\beta-\alpha-\tau)y_t\Big(\mathbf{w}_t^B - (1-\eta)(\beta-\alpha-\tau)y_t\mathbf{x}_t^B\Big)\cdot\mathbf{x}_t^B + \tau \\ &= -\frac{1}{2}\eta^2(\alpha-\beta-\tau)^2 \parallel \mathbf{x}_t^A \parallel^2 - \frac{1}{2}(1-\eta)^2(\beta-\alpha-\tau)^2 \parallel \mathbf{x}_t^B \parallel^2 \\ &\quad +\eta(\alpha-\beta-\tau)y_t\mathbf{w}_t^A\cdot\mathbf{x}_t^A + (1-\eta)(\beta-\alpha-\tau)y_t\mathbf{w}_t^B\cdot\mathbf{x}_t^B + \tau\end{aligned} \tag{13}$$

Setting the partial derivatives of $\mathcal{L}$ with respect to weight τ to zero, we have

$$\begin{aligned}0 = \frac{\partial\mathcal{L}}{\partial\tau} &= \eta^2(\alpha-\beta-\tau) \parallel \mathbf{x}_t^A \parallel^2 + (1-\eta)^2(\beta-\alpha-\tau) \parallel \mathbf{x}_t^B \parallel^2 \\ &\quad +1-\eta y_t\mathbf{w}_t^A\cdot\mathbf{x}_t^A - (1-\eta)y_t\mathbf{w}_t^B\cdot\mathbf{x}_t^B) \\ \Rightarrow \tau &= \frac{(\alpha-\beta)\Big(\eta^2 \parallel \mathbf{x}_t^A \parallel^2 - (1-\eta)^2 \parallel \mathbf{x}_t^B \parallel^2\Big) + \ell_t}{\eta^2 \parallel \mathbf{x}_t^A \parallel^2 + (1-\eta)^2 \parallel \mathbf{x}_t^B \parallel^2}\end{aligned}$$

where the loss $\ell_t = 1-\eta y_t\mathbf{w}_t^A\cdot\mathbf{x}_t^A - (1-\eta)y_t\mathbf{w}_t^B\cdot\mathbf{x}_t^B$. For the sake of simplicity, we denote

$$a = \frac{1}{\eta^2 \parallel \mathbf{x}_t^A \parallel^2 + (1-\eta)^2 \parallel \mathbf{x}_t^B \parallel^2} \quad \text{and} \quad b = \parallel \mathbf{x}_t^A \parallel^2 \parallel \mathbf{x}_t^B \parallel^2 \tag{14}$$

As mentioned in Equation (12), we have $\tau + \lambda = C$ and $\lambda \geq 0$, we can conclude that $\tau \leq C$. Now τ can be determined as follows:

$$\tau = \min\Big\{C, a\Big((\alpha-\beta)(\eta^2 \parallel \mathbf{x}_t^A \parallel^2 - (1-\eta)^2 \parallel \mathbf{x}_t^B \parallel^2) + \ell_t\Big)\Big\} \tag{15}$$

Substituting (15) into (13), we have

$$\begin{aligned}\mathcal{L} &= -\frac{1}{2}a^2\eta^2\Big((\alpha-\beta)(1-\eta)^2 \parallel \mathbf{x}_t^B \parallel^2 - \ell_t)\Big)^2 \parallel \mathbf{x}_t^A \parallel^2 \\ &\quad -\frac{1}{2}a^2(1-\eta)^2\Big((\beta-\alpha)\eta^2 \parallel \mathbf{x}_t^A \parallel^2 - \ell_t)\Big)^2 \parallel \mathbf{x}_t^B \parallel^2 \\ &\quad +a\eta((\alpha-\beta)(1-\eta)^2 \parallel \mathbf{x}_t^B \parallel^2 - \ell_t)y_t\mathbf{w}_t^A\cdot\mathbf{x}_t^A \\ &\quad +a(1-\eta)((\beta-\alpha)\eta^2 \parallel \mathbf{x}_t^A \parallel^2 - \ell_t)y_t\mathbf{w}_t^B\cdot\mathbf{x}_t^B \\ &\quad +a\Big((\alpha-\beta)(\eta^2 \parallel \mathbf{x}_t^A \parallel^2 - (1-\eta)^2 \parallel \mathbf{x}_t^B \parallel^2) + \ell_t\Big)\end{aligned} \tag{16}$$

Setting the partial derivatives of $\mathcal{L}$ with respect to weight α to zero, we have

$$\begin{aligned}0 = \frac{\partial\mathcal{L}}{\partial\alpha} &= -a^2\eta^2\Big((\alpha-\beta)(1-\eta)^2 \parallel \mathbf{x}_t^B \parallel^2 - \ell_t\Big)b \\ &\quad +a^2(1-\eta)^2\Big((\beta-\alpha)\eta^2 \parallel \mathbf{x}_t^A \parallel^2 - \ell_t\Big)b \\ &\quad +a\eta(1-\eta)^2 \parallel \mathbf{x}_t^B \parallel^2 y_t\mathbf{w}_t^A\cdot\mathbf{x}_t^A - a(1-\eta)\eta^2 \parallel \mathbf{x}_t^A \parallel^2 y_t\mathbf{w}_t^B\cdot\mathbf{x}_t^B \\ &\quad +a(\eta^2 \parallel \mathbf{x}_t^A \parallel^2 - (1-\eta)^2 \parallel \mathbf{x}_t^B \parallel^2)\end{aligned} \tag{17}$$

Simplifying the above equality, we have

$$\begin{aligned} &-b\eta(1-\eta)(\alpha-\beta)(\eta^2 \parallel \mathbf{x}_t^A \parallel^2 +(1-\eta)^2 \parallel \mathbf{x}_t^B \parallel^2) \\ &+(1-\eta) \parallel \mathbf{x}_t^B \parallel^2 y_t \mathbf{w}_t^A \cdot \mathbf{x}_t^A - \eta \parallel \mathbf{x}_t^A \parallel^2 y_t \mathbf{w}_t^B \cdot \mathbf{x}_t^B = 0 \end{aligned} \tag{18}$$

Hence, we have

$$\alpha - \beta = \frac{1}{\eta} \frac{y_t \mathbf{w}_t^A \cdot \mathbf{x}_t^A}{\parallel \mathbf{x}_t^A \parallel^2} - \frac{1}{1-\eta} \frac{y_t \mathbf{w}_t^B \cdot \mathbf{x}_t^B}{\parallel \mathbf{x}_t^B \parallel^2}$$

Recall that we have $\alpha + \beta = \gamma$. Therefore, we can conclude that

$$\alpha = \frac{1}{2}\Big(\gamma + \frac{1}{\eta} \frac{y_t \mathbf{w}_t^A \cdot \mathbf{x}_t^A}{\parallel \mathbf{x}_t^A \parallel^2} - \frac{1}{1-\eta} \frac{y_t \mathbf{w}_t^B \cdot \mathbf{x}_t^B}{\parallel \mathbf{x}_t^B \parallel^2}\Big) \tag{19}$$

Similarly, we have

$$\beta = \frac{1}{2}\Big(\gamma - \frac{1}{\eta} \frac{y_t \mathbf{w}_t^A \cdot \mathbf{x}_t^A}{\parallel \mathbf{x}_t^A \parallel^2} + \frac{1}{1-\eta} \frac{y_t \mathbf{w}_t^B \cdot \mathbf{x}_t^B}{\parallel \mathbf{x}_t^B \parallel^2}\Big) \tag{20}$$

Recall that we have $\alpha \geq 0$, $\beta \geq 0$, and $\alpha + \beta = \gamma$. Hence, we can conclude that $0 \leq \alpha \leq \gamma$ and $0 \leq \beta \leq \gamma$. That is

$$\begin{aligned} \alpha &= \min\Big\{\gamma, \frac{1}{2}\Big(\gamma + \frac{1}{\eta} \frac{y_t \mathbf{w}_t^A \cdot \mathbf{x}_t^A}{\parallel \mathbf{x}_t^A \parallel^2} - \frac{1}{1-\eta} \frac{y_t \mathbf{w}_t^B \cdot \mathbf{x}_t^B}{\parallel \mathbf{x}_t^B \parallel^2}\Big)\Big\} \\ \beta &= \min\Big\{\gamma, \frac{1}{2}\Big(\gamma - \frac{1}{\eta} \frac{y_t \mathbf{w}_t^A \cdot \mathbf{x}_t^A}{\parallel \mathbf{x}_t^A \parallel^2} + \frac{1}{1-\eta} \frac{y_t \mathbf{w}_t^B \cdot \mathbf{x}_t^B}{\parallel \mathbf{x}_t^B \parallel^2}\Big)\Big\} \end{aligned}$$

B Proof of Proposition 2

For the Adaptive PA algorithm, we have new Lagrangian as follows:

$$\begin{aligned} \mathcal{L} = &\frac{1}{2} \parallel \mathbf{w}^A - \mathbf{w}_t^A \parallel^2 + \frac{1}{2} \parallel \mathbf{w}^B - \mathbf{w}_t^B \parallel^2 \\ &+(\gamma - \alpha - \beta)z + (C - \lambda - \tau)\xi + \eta(\alpha - \beta - \tau) y_t \mathbf{w}^A \cdot \mathbf{x}_t^A \\ &+(1-\eta)(\beta - \alpha - \tau) y_t \mathbf{w}^B \cdot \mathbf{x}_t^B + \tau + \frac{\zeta}{2}(\eta - \eta_t)^2 \end{aligned} \tag{21}$$

Assuming that α, β, and τ are independent on the new value of η, we have $\frac{\partial \alpha}{\partial \eta} = 0$, $\frac{\partial \beta}{\partial \eta} = 0$, and $\frac{\partial \tau}{\partial \eta} = 0$.

Setting the partial derivatives of $\mathcal{L}$ with respect to the variable η to zero, we have

$$\begin{aligned} 0 = \frac{\partial \mathcal{L}}{\partial \eta} &= \zeta(\eta - \eta_t) + (\alpha - \beta - \tau) y_t \mathbf{w}^A \cdot \mathbf{x}_t^A - (\beta - \alpha - \tau) y_t \mathbf{w}^B \cdot \mathbf{x}_t^B \\ \Rightarrow \eta &= \eta_t - \frac{1}{\zeta}\Big((\alpha - \beta - \tau) y_t \mathbf{w}^A \cdot \mathbf{x}_t^A - (\beta - \alpha - \tau) y_t \mathbf{w}^B \cdot \mathbf{x}_t^B\Big) \end{aligned} \tag{22}$$