Robust Fusion Methods for Distribution Shifts

A Thesis Presented by:

Matt Higger

to

The Department of Electrical Engineering

in partial fulfillment of the requirements for the degree of

Master of Science

in

Electrical Engineering

Northeastern University
Boston, Massachusetts

March 2013
Abstract

We explore the effect of a shift in an underlying joint distribution between feature and target variables in a classification problem. This arises in the context of airport explosives detection where breaking a single sensor could add risk to the fused decision about whether a passenger is dangerous. We develop a method which mitigates the negative effects of a failure of subset of modalities or sensors within a single modality. In a Brain Computer Interface (BCI) application, we note that an artifact can be viewed as a shift in this joint distribution. A similar methods is applied to manage the effect of the superfluous signals introduced by artifacts in the Electroencephalography (EEG) features. For BCI, we develop a method which can be used to incorporate information, from EEG or external sensors, about whether artifact events have occurred.
Acknowledgements

I’d like to thank Deniz Erdogmus, Murat Akcakaya and Umut Orhan for their help in working on these projects. Additionally, thanks to Nat Beagley and Pacific Northwest National Laboratories who offered a chance to first consider sensor fusion.
Contents

Abstract i

Acknowledgements ii

1 Introduction 1
   1.1 Problem Statement: Explosives Detection .......................... 1
   1.2 Other Approaches: Explosives Detection ............................ 2
   1.3 Contribution: Explosives Detection ................................. 3
   1.4 Problem Statement: RSVP .............................................. 4
   1.5 Other Approaches: RSVP ............................................... 4
   1.6 Contribution: RSVP .................................................... 5
   1.7 Organization .......................................................... 5

2 Background 7
   2.1 Curse of Dimensionality .............................................. 7
   2.2 Naive Bayes Sensor Model ........................................... 8
   2.3 Sensor Model .......................................................... 9
   2.4 Risk ................................................................. 9
   2.5 Traditional Sensor Fusion ........................................ 10

3 Solution A: Learn Broken Sensor Parameters Online 11
   3.1 Information Theory .................................................. 11
   3.2 Entropy, Conditional Entropy and Mutual Information ............. 12
   3.3 Learning Parameters Online Cannot Improve Performance ........... 13
3.4 Discussion ................................................................. 14

4 Solution B: Robust Fusion for Sensor Failure .......................... 16
  4.1 Sensor failure model .................................................. 16
  4.2 Minimum risk fusion rule ......................................... 17
  4.3 Development of Robust Fusion .................................. 17
  4.4 Results ................................................................. 21
    4.4.1 Performance Across Different l .................. 21
    4.4.2 Computational Challenges ........................ 22
    4.4.3 Performance Across Different Number of Sensors .... 23
    4.4.4 Performance Across Different $P_T(1)$ ....... 23
  4.5 Discussion .......................................................... 23

5 Solution C: Identify Failed Sensors ................................... 25
  5.1 Other methods ....................................................... 25
  5.2 Non-Parametric Method .......................................... 26
    5.2.1 Detection of Sensor Failure ........................ 26
    5.2.2 Results ....................................................... 27
  5.3 Discussion .......................................................... 29

6 Artifact-Robust Fusion in RSVP Keyboard ............................ 30
  6.1 RSVP Keyboard$^{TM}$ .............................................. 30
    6.1.1 Visual Presentation .................................. 30
    6.1.2 Feature Extraction .................................... 31
    6.1.3 Language Modeling .................................... 33
    6.1.4 Classifier .................................................. 33
  6.2 Robust Classifier .................................................. 33
  6.3 Experiments ........................................................ 34
    6.3.1 Typing Accuracy ..................................... 35
    6.3.2 Typing Duration ....................................... 35
  6.4 Discussion ........................................................ 36
<table>
<thead>
<tr>
<th>CONTENTS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>7 Conclusions</td>
<td>38</td>
</tr>
<tr>
<td>Bibliography</td>
<td>40</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

This work focuses on shifting joint distributions between feature and target variables. Given such a general problem, it isn’t surprising to find two dramatically different applications: airport explosives detection and brain computer interfaces (BCIs). Given the sensitivity of explosives detection data, all development must be done for synthetic datasets. Synthetic data offers more control in defining the particular problem we solve. Fortunately, we are also given EEG data in the BCI application which offers all the applicability and complications of the real world. We begin with a theoretical exploration of sensor failure in a fused explosives detection scheme and continue on to a live simulation of artifact-robust fusion in BCI.

1.1 Problem Statement: Explosives Detection

This work was presented to Pacific Northwest National Laboratories in the context of explosives detection. The following is a brief discussion of how our work fits their application.

Currently, explosives detection relies on different modalities for detection. There is, of course, a metal detector which passengers walk through in addition to having their carry-on-baggage imaged. Additionally, imaging technologies such as Millimeter Wave and X-Ray Backscatter could examine their bodies and chemical detections by Mass Spectroscopy could find traces of explosives on passenger’s hands [3]. The
various modalities create a new problem, the information must be combined to make a decision about a passenger’s threat [5]. This is a problem of fusion, the combination of different types of information to make a decision about a commonly related variable.

This fusion is currently performed by a human who ultimately determines how safe a passenger is. Human fusion has the advantage of incorporating the currently unquantifiable elements of body language and social contextual information into account. (Such information, while subjective, shouldn’t be underestimated; Richard Reid, the shoe bomber, was initially detained for questioning for looking “dishevelled” [2]). However, when it comes to the fusion of harder, objective information, automated systems are more consistent, accurate and reliable. The industry is seeking methods which can make sense of disparate sources of information to inform a human operator who will make all final decisions about the movement of passengers.

As a threat detection system grows to incorporate a larger number of modalities, it is more vulnerable to sensor failure. For our purposes, we define sensor failure as the event that a source of information changes its relationship to our target variable but still produces output. In the context of explosives detection, this could be caused by a mis-calibration of equipment, hardware failure or error in estimation of its operating characteristic. Such errors, at best, could increase the false positive alarm rate, incorrectly identifying safe passengers as threats. Further, they could decrease the detection rate allowing potentially dangerous passengers to pass through checkpoints. (This problem is particularly challenging because it is not possible to detect online without true positive test cases. True-positive test cases amounts to moving live weapons through airport security, a solution with definite drawbacks).

1.2 Other Approaches: Explosives Detection

Many sensor fusion methods rely on the strong assumption that the sensors which generate the data operate according to a predetermined characteristic at all times. When a sensor fails, and this assumption is challenged, detection systems demonstrate behaviors different than the intended. In such occurrences, one possible approach is
to identify the broken sensors and update the fusion rule accordingly. The detection of broken sensors must be done online, without an access to the ground decision truth, relying on the relationships between sensor decisions. One common method of finding sensor failures is training a Neural Network to identify the particular relationship between sensors while all are known to be correctly functional. Then, during operation, the system can use the same Neural Network to declare a sensor broken if the relationship changes online [7], [9]. Another approach is to manually configure sensor relationships using a consistency measure [15] or fuzzy set membership [17]. All these methods only identify failed sensors.

Once the broken sensor is detected, ideally the fusion rule must be amended to account for the shift in sensor characteristic. The most common way of achieving this is by weighing the importance of sensor data, using fuzzy sets, according to a sensor’s agreement with all other sensor data [11]. Other methods modify sensor output to maintain stability within a control system [12].

1.3 Contribution: Explosives Detection

Our first intuition would be to learn a sensor’s new characteristic after it has broken. If we could learn this characteristic we could update the parameters within the fusion model such that it continues to minimize some risk function. We prove that no such approach could improve the classification rule.

Given such a discouraging result, we proceed with a second approach: create a fusion rule which is robust to shifts in a sensor’s characteristic. Relying only on the mean of the sensor failure model, our fusion rule minimizes the expected Bayesian risk without any additional training. Moreover, if any sensor breaks intermittently, operating outside its initial characteristic for disjoint periods of time, we can take full advantage of the classification information it offers when it becomes operational again, an advantage over sensor removal methods. Our method is shown to decrease risk, depending heavily on the probability of sensor failure, the number of sensors, the probability distribution of the true decision, and the particular characteristic of the sensors themselves.
Our final solution is to detect sensor failures. Once a sensor failure has been detected, we remove the sensor from fusion. (According to our results with the first solution, we know that adding it back into fusion cannot yield any improvement of performance). We show a parametric and non-parametric algorithm for detecting sensor failure.

1.4 Problem Statement: RSVP

Locked-In Syndrome (LIS) is a condition when a person cannot control their own muscle movement, quite literally taking one’s voice away. Given how isolating this condition can be, we become interested in creating systems which empower people with LIS to communicate as efficiently as possible. Currently, the best way to do this is to create an interface based on a user’s eye gaze. However, there is a subset of people who are without control of any muscle group or eye gaze, this group is said to be ‘totally Locked-In’. Brain computer interfaces offer a promising avenue to give a voice to this group. Generally, BCIs are methods which extract a person’s intent through measurement of internal body signals. A common method, as we employ here, is to use the voltage of a person’s scalp measured through EEG. EEG is a relatively cheap, portable, non-invasive way of measuring brain waves.

1.5 Other Approaches: RSVP

There are a number of EEG brain-phenomena which have been used to classify user intent. In motor imagery, a BCI system is designed to detect the signal generated by imagining the movement of a body part [1]. Additionally a steady state visually evoked potential (SSVEP) appears when a user is exposed to a periodic visual stimulus. Exposing a user to flickering checkerboard patterns, the induced SSVEP signals can be used to learn the user’s gaze position from the frequency content of their brain waves [10].

Moreover, the EEG signals are sufficient for simple letter selection in the context of a typing algorithm for people with total-LIS. The P300 signal, an event related
potential (ERP) which occurs when a user is surprised by a circumstance, is commonly used for BCI spelling systems. P300 speller and Berlin BCI’s Hexo Spell are well known examples of such systems [8], [16]. In our approach, we utilize Rapid Serial Visual Presentation (RSVP), which presents the stimuli on the same location of the screen with temporal separation. The accuracy and speed of P300 typing systems suffer from low signal-to-noise ratio (SNR), the presence of artifacts in the signal and sensor failure and other effects that cause non-stationarity in the observed EEG signals.

1.6 Contribution: RSVP

The artifacts and/or sensor failure change the underlying distribution of the EEG data in a BCI system causing a change in the optimal stimuli classification rule and degrading the system performance. Our goal is to develop a classification rule that is robust to changes in the assumed data distributions. To achieve this, we estimate the distribution of the data under different conditions, and using these distribution we develop our classification rule.

1.7 Organization

In Chapter 2 we present the necessary mathematical background to put this work into context.

In Chapter 3 we examine the possibility of learning broken sensor characteristics on line in an abstract setting. If one could learn the new parameters of a sensor they could update the fusion rule so that it would continue to optimize some objective function (minimize risk). However, we prove that any such effort can not yield an improvement in performance.

In Chapter 4 we examine how to build a fusion rule which is robust to shifts in sensor characteristics in an abstract setting. Relying only on the mean of the sensor failure model, our fusion rule minimizes the expected Bayesian risk without any additional training. Moreover, if any sensor breaks intermittently we can take
full advantage of the classification information it offers when it becomes operational again, an advantage over sensor removal methods. Our method is shown to decrease risk, depending heavily on the probability of sensor failure, the number of sensors, the probability distribution of the true decision, and the particular characteristic of the sensors themselves.

In our last abstract section, Chapter 5, we propose algorithms which find broken sensors. The removal of broken sensors may or may not be the best course of action. If, for example, the sensor distribution has only shifted slightly, its inclusion in fusion may still be better than its exclusion, though sub-optimal.

Next, we turn to our BCI application in Chapter 6. We build a method which allows a fusion mechanism to adapt to the possibility of a predictable shift in the joint distribution between the feature and artifact variables.

We end with a brief summary of our results in Chapter 7.
Chapter 2

Background

Our problem can be stated as one of classification. Given some set of random variables, \( D_1 \ldots D_S \), how can we determine the value of our target variable \( T \)? Probability theory offers a graceful way of drawing conclusions about \( T \) from \( D_1 \ldots D_S \) as it handles both the uncertainty in the relationship between our data and target variables as well as the uncertainty in our knowledge of such a relationship. Given this, a first intuition to building a classification mapping from \( D_1 \ldots D_S \) to \( T \) is to learn the joint distribution of all variables, \( P_{D_1,\ldots,D_S,T} \). With such a wealth of information we could simply apply Bayes rule to incorporate our observations and build a posterior belief of our target variable. Such an approach still has problems.

2.1 Curse of Dimensionality

The joint distribution, \( P_{D_1,\ldots,D_S,T} \), is not found easily in many problems of interest. Assuming that we don’t receive any hints about what such a distribution might look like, we must estimate it somehow. Counting the number of unique values the set of variables could take on:

\[
|D_1,\ldots,D_S,T| = |T| \prod_{i=1}^{S} |D_i| \tag{2.1}
\]
CHAPTER 2. BACKGROUND

where $|X|$ is the number of values the random variable, $X$ can take. Under the assumption that an estimation method requires some amount of samples linearly related to the number of values a its variables can take, we notice the large challenge posed by large data sets. As the dimensionality of the data grows we need an exponentially increasing number of samples to effectively estimate the distribution. While higher dimensional data may lead to better classification performance it doesn’t come for free, it dramatically increases the difficulty of building the joint distribution.

2.2 Naive Bayes Sensor Model

Fortunately, we can impose further structure to limit the difficulties posed by higher dimensional data. Intuitively, we understand that the problem with adding a new variable into our model is that we have to model its independent relationship to all the other variables already in the model. What if we minimize the number of relationships between variables to the minimum amount which still connects each data variable to our target variable? Then we wouldn’t need as much data to estimate our joint. More precisely, such a simplification comes at a cost, we’ve restricted our estimated distribution in a way that might prevent it from accurately reflecting the true joint distribution.

This assumption is quite common in practice. It is a great starting point for many problems. For those problems with weak dependencies between feature variables (outside the dependence through the target variable), it is a good ending point too.
2.3 Sensor Model

For simplicity, we assume a feature variable from a binary model. Given a set of decisions: \( \{D_1, ..., D_S\} \) where \( D_i \in \{0, 1\} \), each sensor has characteristic \( P_{D_i|T}(1|1) = \alpha_i, P_{D_i|T}(0|0) = \beta_i \), then

\[
P_{D_i|T}(d_i|t) = \left(\alpha_i^{d_i}(1 - \alpha)^{(1-d_i)}\right)^t \left(\beta_i^{(1-d_i)}(1 - \beta_i)^{d_i}\right)^{(1-t)}
\]  

(2.3)

Moreover, we make the Naive Bayes assumption, such that:

\[
P_{D_{1:S},T}(d_{1:S}, t) = P_T(t) \prod_{i=1}^{S} P_{D_i|T}(d_i, t)
\]

\[
= P_T(t) \prod_{i=1}^{S} \left(\alpha_i^{d_i}(1 - \alpha)^{(1-d_i)}\right)^t \left(\beta_i^{(1-d_i)}(1 - \beta_i)^{d_i}\right)^{(1-t)}
\]

(2.4)

2.4 Risk

With what we’ve defined so far it is possible to create a classifier which minimizes the probability of mis-classification. However, not all mistakes are created equal. Consider our application: explosives detection at an airport. Is it worse to have a machine go off when a safe passenger walks through or to have the machine stay silent as a dangerous passenger walks through? Clearly, stopping a passenger for extra screening is a minor consequence when compared to letting a dangerous passenger onto an airplane. However, we haven’t yet mentioned the mathematics required to distinguish between these consequences. In this context, and many others, we can see that simply minimizing the classification error is a poor objective function.

With this in mind, we define the risk as a quantity which describes how negative an outcome is. We define \( F \) as the random variable which is our classifiers estimate of \( T \), our target variable. Then \( r_{f|t} \) is the risk of choosing \( F = f \) while \( T = t \). In our application, \( r_{0|1} \) is the risk associated with allowing a dangerous passenger to pass through security while \( r_{1|0} \) is the risk associated with stopping a safe passenger for extra screening. \( r_{0|0} \) and \( r_{1|1} \) are the risks associated with correctly identifying a safe
or dangerous passenger, and are set to zero by default). By choosing the values of risk we answer the question, “How much more negative is it to make one mistake vs making another?”. Such an addition to our formulation allows us to minimize risk, a more useful objective function than simply minimizing classification error.

We note in passing that the task of assigning numerical values to risk can be a subjective task in an objective algorithm. Within the explosives detection community, the values are chosen to tune the algorithm to pass a particular validation test which requires some maximum value of probability of false alarm and minimum value of probability of detection. Determining the performance of this hardware by such an objective function may or may not be optimal in terms of keeping terrorists off airplanes; though it is simple.

2.5 Traditional Sensor Fusion

Traditionally if a constant characteristic, $\Theta = [\alpha_1, ..., \alpha_S, \beta_1, ..., \beta_S]$, is assumed through the system’s operation, the fusion rule is obtained by the loglikelihood ratio approach which minimizes the Bayesian risk:

$$ F(d_{1,S}) = 1 \quad \text{when} \quad L(d_{1,S}) = \log \frac{r_{0|1} P_T(1)}{r_{1|0} P_T(0)} + \sum_{D_i} \log \frac{P_{D_i|T}(d_i|1)}{P_{D_i|T}(d_i|0)} > 0 $$

$$ = 0 \quad \text{otherwise} \quad \quad (2.5) $$
Chapter 3

Solution A: Learn Broken Sensor Parameters Online

We know that we can minimize the risk by using 2.5, but what happens when one of our sensors fails? When a conditional distribution, $P_{D_i|T}$, is no longer accurate because a sensor has failed we can no longer rely on 2.5 to minimize our risk. However, just because a sensor has failed does not necessarily mean that it won’t be useful for classification. In fact, if we could learn its new conditional distribution, $P'_{D_i|T}$, then we could incorporate it into our old fusion method and achieve a minimum expected risk. In this chapter, we prove that such an approach cannot help. We present a proof that any such method with attempts to learn broken sensor parameters online cannot increase classification performance.

3.1 Information Theory

Information theory offers an elegant way of quantifying how much information is held within a message. Intuitively, the message, ”the sky will be blue or gray tomorrow”, has little information because we are not at all surprised by its content. However, ”the sky will be plaid patterned tomorrow”, contains much more information because we are so surprised by its content. Information theory quantifies exactly how much information a message contains based on how surprised we are to see it. Presented
here is a brief introduction to the small piece of information theory necessary for our result, a more thorough treatment is in [4].

3.2 Entropy, Conditional Entropy and Mutual Information

The entropy of a random variable is a measure of its uncertainty. A random variable with a high entropy has samples which are consistently surprising. For example, the lottery digits tomorrow have a high entropy because we do not know which numbers to expect. A random variable with a low entropy has samples which are consistently expected. The value of the date tomorrow has a low entropy because we know exactly what to expect (In fact, we know with certainty what the date will be tomorrow, so this ”random” variable has an entropy of zero). More rigorously, the entropy, $H$, of a random variable, $X$, is defined as:

$$H(X) = -\sum_x P_X(x) \log P_X(x)$$  

(3.1)

Shannon was the first to suggest that this entropy was the quantification of the average information which would be most useful for describing communication channels [14].

We can extend our definition to highlight how information is shared between variables. The conditional entropy of a random variable $X$, given random variable $Y$ is defined as:

$$H(X|Y) = -\sum_x P_{X|Y}(x|y) \log P_{X|Y}(x|y)$$  

(3.2)

Intuitively, this can be understood as how uncertain one event is given another. The message, “it is going to rain tomorrow” might be surprising, but given that “the sky will be dark and cloudy tomorrow” we may not be as surprised to hear it will be raining. In other words, conditional entropy allows us to partition the information in $X$. There is the information about $X$ we would expect to know from $Y$, $H(X|Y)$,
and the mutual information between the variables, $I(X,Y)$. Explicitly, we define the mutual information as:

$$I(X,Y) = H(X) - H(X|Y)$$

(3.3)

These definitions obey many of our intuitions about what a definition of information should be. It is true that:

1. Information is always positive

2. Information about one random variable only decrease our surprise about another random variable. (Conditioning never increases entropy).

3. Information common to two random variables is never greater than either of the variables taken separately.

Our result relies on the fact that if the mutual information between $X$ and $Y$ is zero, then $Y$ cannot be used to predict $X$ in any way which has greater accuracy than all methods which don’t utilize $Y$.

### 3.3 Learning Parameters Online Cannot Improve Performance

Given the problem of sensor failure, a first intuition might be to identify failed sensors and learn their new parameters online. Ideally, once $\alpha'_i$ and $\beta'_i$ were found, they could be inserted into (2.5) to maintain minimum risk. Unfortunately, under the conditional independence assumption on the decisions of different sensors, even if identification of broken sensors is both instant and as accurate as our ability to measure will allow, learning $\alpha'_i$ and $\beta'_i$ can not yield an improvement in performance. We prove this is the case by examining the equivalent problem of adding a new sensor $S + 1$ and its decision, $D_{S+1}$, by learning its characteristic online.
CHAPTER 3. SOLUTION A: LEARN BROKEN SENSOR PARAMETERS ONLINE

Given the binary, Naive Bayes Model with known joint distribution of \( S \) sensors, 
\[
P_{D_1:S,T} = \prod_{i=1}^{S} P_{D_i|T}(d_i|t),
\]
we would like to add the decision \( D_{S+1} \) in the fusion rule by learning \( P_{D_{S+1}|T} \). However, because online processing has no access to the values of \( T \), we restrict ourselves to using only samples from \( D_{1:S} \). We define the approximated conditional distribution \( \tilde{P}_{D_{S+1}|T}(d_{S+1}|t) = P_{D_{S+1}|D_{1:S}}(d_{S+1}|d_{1:S}) \) for approximated decision \( \tilde{D}_{S+1} \). We investigate the mutual information between \( \tilde{D}_{S+1} \) and the truth \( T \) according to [4] such that
\[
I(\tilde{D}_{S+1}, T|D_{1:S}) = H(\tilde{D}_{S+1}|D_{1:S}) - H(\tilde{D}_{S+1}|T, D_{1:S}).
\]

Note that
\[
H(\tilde{D}_{S+1}|T, D_{1:S}) = \sum_{\tilde{d}_{S+1}} \sum_{t} \sum_{d_{S+1}} P(\tilde{d}_{S+1}, t, d_{1:S}) \log P(\tilde{d}_{S+1}|t, d_{1:S})
\]
\[
= \sum_{\tilde{d}_{S+1}} \sum_{t} \sum_{d_{S+1}} P(\tilde{d}_{S+1}, t, d_{1:S}) \log P(\tilde{d}_{S+1}|d_{1:S})
\]
\[
= H(\tilde{D}_{S+1}|D_{1:S}),
\]

where the second equality is due to the essential fact that \( \tilde{D}_{S+1} \) has no dependence on \( T \) outside the influence of \( D_{1:S} \). Therefore, \( I(\tilde{D}_{S+1}, T|D_{1:S}) = 0 \). The lack of mutual information between \( \tilde{D}_{S+1} \) and \( T \) shows that \( \tilde{D}_{S+1} \) cannot add additional information to be used in the classification of \( T \) once we are given \( D_{1:S} \).

3.4 Discussion

We examine the efficacy of adding a broken sensor back into the classification rule by learning its characteristic online. We find that if the conditional distribution is learned using only the other feature variables \( D_1...D_S \), then adding the broken sensor back into classification does not increase the performance. The biggest criticism of this method is that we only approximate the conditional distribution of our new or broken sensor using the other feature variables. This assumption is quite natural though, no better approximation is available as this takes all the available information into
account (T isn’t available online). The impact of this result is that when a sensor fails there is nothing that can be done online to fix it short of some method of retraining for new conditional distributions. In this section we assume that no information is known about the failure of the sensor itself, we shall see in 4 that such information could be utilized to make a system robust to failure.
Chapter 4

Solution B: Robust Fusion for Sensor Failure

While we cannot incorporate broken sensors into the classification in such a way which is helpful, we can minimize their negative effects. In this section, we build a probabilistic model of sensor failure and use it to mitigate the negative effects of sensor failure.

4.1 Sensor failure model

We assume that every sensor fails independently with some $P_{R_i}(1) = l$. Furthermore, given that $D_i$ has failed, it is assigned a new characteristic: $\alpha'_i$ and $\beta'_i$, sampled from distributions $P_{A_i}$ and $P_{B_i}$ respectively. We impose only the natural restriction that the domain of $P_{A_i}$ and $P_{B_i}$ be $[0,1]$. We define $R_i$ as the event that sensor $S_i$ fails, and $R = \sum R_i$ as the number of failed sensors. We note that $R$ follows a binomial distribution, $R \sim B(S,l)$. 

16
4.2 Minimum risk fusion rule

This sensor failure model leads to a natural rule which minimizes the risk of a potentially failing sensor system. The resulting robust fusion rule is

\[
F'(d_{1:S}) = \begin{cases} 
1 & \text{when } L(d_{1:S}) + \log \frac{\sum_{r=0}^{\ell_1} C_1(d_{1:S}, r)}{\sum_{r=0}^{\ell_0} C_0(d_{1:S}, r)} > 0 \\
0 & \text{otherwise}
\end{cases}
\] (4.1)

where \( L(d_{1:S}) \) is defined by equation 2.5 and

\[
C_1(d_{1:S}, r) = \frac{P_R(r)}{\binom{S}{r}} \sum_{r_1, \ldots, r_r} \prod_{n=r_1 \ldots r_r} \frac{1}{P_{D_n|T}(d_n|1)} \left( E[A_n]^{d_n} (1 - E[A_n])^{1-d_n} \right) \\
C_0(d_{1:S}, r) = \frac{P_R(r)}{\binom{S}{r}} \sum_{r_1, \ldots, r_r} \prod_{n=r_1 \ldots r_r} \frac{1}{P_{D_n|T}(d_n|0)} \left( E[B_n]^{1-d_n} (1 - E[B_n])^{d_n} \right)
\]

Such a fusion rule obeys our intuitions at the extreme cases. When \( l = 0 \), the case where no sensors fail, the rule simplifies to the traditional likelihood ratio fusion rule as defined in (2.5). When \( P_R(S) = 1 \), the case where all sensors fail, it simplifies to always choosing \( F(d_{1:S}) \) based exclusively on prior information, \( \frac{r_{0|1}P_T(1)}{r_{1|0}P_T(0)} \), a guess as good as any other given that we have no useful information related to \( T \) from \( D_{1:S} \).

4.3 Development of Robust Fusion

We first extend the definition of the minimum expected Bayesian risk to include the sensor decision distributions such that
\[ E_{F,T}[\text{risk}_{F|T}] = \sum_{f} \sum_{t} P_{F,T}(f,t) r_{f|t} \]
\[ = \sum_{d_{1:S}} \sum_{f} \sum_{t} P_{F,D_{1:S},T}(f,d_{1:S},t) r_{f|t} \]
\[ = \sum_{d_{1:S}} \sum_{f} \sum_{t} P_{F'|T,D_{1:S}}(f|d_{1:S},t) P_{D_{1:S}}(d_{1:S},t) r_{f|t} \]
\[ = \sum_{d_{1:S}} \sum_{t} P_{D_{1:S},T}(d_{1:S},t) r_{F(d_{1:S})|t} \] (4.2)

The fourth equality is due to the fact that \( P_{F'|T,D_{1:S},T}(f|d_{1:S},t) = \delta(f - F(d_{1:S})) \), where \( \delta(t - t_0) = 1 \) for \( t = t_0 \), and 0 otherwise. Then, we divide the set of all \( d_{1:S} \) into subsets:

\[ S_i = \{ d_{1:S} | F(d_{1:S}) = i \} \] (4.3)

Using (4.3) in (4.2), we have

\[ E_{F,T}[\text{risk}_{F|T}] = \sum_{S_0} \sum_{t} P_{D_{1:S},T}(d_{1:S},t) r_{F(d_{1:S})|t} + \sum_{S_1} \sum_{t} P_{D_{1:S},T}(d_{1:S},t) r_{F(d_{1:S})|t} \]
\[ = \sum_{S_0} P_{D_{1:S}|T}(d_{1:S}|1) P_T(1) r_{0|1} + \sum_{S_1} P_{D_{1:S}|T}(d_{1:S}|0) P_T(0) r_{1|0} \] (4.4)

We obtain the final equality assuming \( r_{0|0} = r_{1|1} = 0 \).

In order to include the effect of sensor failure in the expected risk definition we seek \( P_{D_{S}|T,R_i} \) to use in 4.4. Recalling the conditional distribution in (2.3) and dropping \( R_i = 1 \) for notational convenience, we compute

\[ P_{D_s|T}(d_s|1) = \int_{0}^{1} P_{A_s,D_s|T}(\alpha_s',d_s|1) d\alpha_s' \]
\[ = \int_{0}^{1} P_{D_s|T,A_s}(d_s|1,\alpha_s') p_{A_s}(\alpha_s') d\alpha_s' \]
\[ = E[A_s]^{d_s} (1 - E[A_s])^{(1-d_s)} \] (4.5)
From a similar line of reasoning, it also follows that $P_{D_s|T}(d_s|0) = E[B_s]^{1-d_s}(1 - E[B_s])^{d_s}$.

We proceed to calculate the expected risk associated with a particular set, $[r_1...r_r]$, of sensors failing. We use the well known identity $E_{F,T}(\cdot) = E_R[E_{F,T|R=r}(\cdot)]$ where $R$ is defined as the total number of broken sensors in Section 4.1. Then, using the result in (4.5), we have

\[
E_{F,T|R=r}[risk_{F|T}]
= \int_0^1 \cdots \int_0^1 \left( \sum_{S_0} P_{D_{r_1},T}(d_{r_1}|1) r_{0|1} + \sum_{S_1} P_{D_{r_1},T}(d_{r_1}|0) r_{1|0} \right) P_A(\alpha_{r_1}) \cdots p_B(\beta_{r_r}) d\alpha_{r_1} \cdots d\beta_{r_r}
\]

\[
= \sum_{S_0} \left( \sum_{r_1=0}^1 \int_0^1 \cdots \int_0^1 \left( \sum_{S_1} P_{D_{r_1},T}(d_{r_1}|0) r_{1|0} \right) P_A(\alpha_{r_1}) d\alpha_{r_1} \cdots p_B(\beta_{r_r}) d\beta_{r_r} \right)
\]

\[
+r_{0|1} P_T(1) \sum_{S_0} \prod_{r_i=0}^1 P_{D_{r_i}|T}(d_{r_i}|1) \prod_{r_i=1}^1 \left( E[A_n]^{d_n} (1 - E[A_n])^{1-d_n} \right)
\]

\[
+r_{1|0} P_T(0) \sum_{S_1} \prod_{r_i=0}^1 P_{D_{r_i}|T}(d_{r_i}|0) \prod_{r_i=1}^1 \left( E[B_n]^{1-d_n} (1 - E[B_n])^{d_n} \right)
\]

\[
= r_{0|1} \sum_{S_0} \prod_{r_i=1}^1 P_{D_{r_i}|T}(d_{r_i}|1) \prod_{r_i=0}^1 \frac{1}{P_{D_{r_i}|T}(d_{r_i}|1)} \left( E[A_n]^{d_n} (1 - E[A_n])^{1-d_n} \right)
\]

\[
+r_{1|0} \sum_{S_1} \prod_{r_i=0}^1 P_{D_{r_i}|T}(d_{r_i}|0) \prod_{r_i=1}^1 \frac{1}{P_{D_{r_i}|T}(d_{r_i}|0)} \left( E[B_n]^{1-d_n} (1 - E[B_n])^{d_n} \right)
\]

(4.6)

We take the average of the (4.6) for every possible selection of $[r_1...r_r]$ (for every $r$ dimensional subset of $S$ sensors) such that
CHAPTER 4. SOLUTION B: ROBUST FUSION FOR SENSOR FAILURE

\[
E_{F,T|R=r}[\text{risk}_{F|T}]
\]
\[
= r_0|1 \sum_{S_0} P_{D_1:S,T}(d_{1:S}, 1) \frac{1}{r} \sum_{r_1 \ldots r_r} \prod_{n=r_1 \ldots r_r} \frac{1}{P_{D_n|r}(d_n|1)} \left( E[A_n]^{d_n} \left( 1 - E[A_n] \right)^{(1-d_n)} \right)
\]
\[
+ r_1|0 \sum_{S_1} P_{D_1:S,T}(d_{1:S}, 0) \frac{1}{r} \sum_{r_1 \ldots r_r} \prod_{n=r_1 \ldots r_r} \frac{1}{P_{D_n|r}(d_n|0)} \left( E[B_n]^{1-d_n} \left( 1 - E[B_n] \right)^{(d_n)} \right)
\]

(4.7)

Then we take the expected value of (4.7) with respect to \( R \) such that

\[
E_{F,T}[\text{risk}_{F|T}]
\]
\[
= r_0|1 \sum_{S_0} P(d_{1:S}, 1) \sum_{r=0}^{r=S} C_1(d_{1:S}, r) + r_0|1 \sum_{S_1} P(d_{1:S}, t) \sum_{r=0}^{r=S} C_0(d_{1:S}, r) \tag{4.8}
\]

where

\[
C_1(d_{1:S}, r) = \frac{P_R(r)}{r} \sum_{r_1 \ldots r_r} \prod_{n=r_1 \ldots r_r} \frac{1}{P_{D_n|r}(d_n|1)} \left( E[A_n]^{d_n} \left( 1 - E[A_n] \right)^{(1-d_n)} \right)
\]
\[
C_0(d_{1:S}, r) = \frac{P_R(r)}{r} \sum_{r_1 \ldots r_r} \prod_{n=r_1 \ldots r_r} \frac{1}{P_{D_n|r}(d_n|0)} \left( E[B_n]^{1-d_n} \left( 1 - E[B_n] \right)^{(d_n)} \right)
\]

The advantage of such a formulation is that it allows us to express the increase in risk for each combination of decisions. Because of this, we can minimize the risk by comparing the potential contribution of the fusion different sum values in (4.8).

Namely,

\[
F(d_{1:S}) = 1 \quad \text{when} \quad r_0|1 P_{D_1:S,T}(d_{1:S}, 1) \sum_{r=0}^{r=S} C_1(d_{1:S}, r) < r_0|1 P_{D_1:S,T}(d_{1:S}, 0) \sum_{r=0}^{r=S} C_0(d_{1:S}, r)
\]
\[
\leftrightarrow \quad L(d_{1:S}) + \log \frac{\sum_{r=0}^{r=S} C_1(d_{1:S}, r)}{\sum_{r=0}^{r=S} C_0(d_{1:S}, r)} < 0 \tag{4.9}
\]
\[
= 0 \quad \text{otherwise}
\]
CHAPTER 4. SOLUTION B: ROBUST FUSION FOR SENSOR FAILURE

4.4 Results

The algorithm is implemented in MATLAB 2012b. Due to the computational requirements of the algorithm, we model for $S < 10$. All results are the mean of 100 Monte Carlo simulated $\theta$s. Unless explicitly mentioned otherwise, in the following sections the parameters used are:

$$S = 5, l = .1, \alpha_i \sim U(0, 1), \beta_i \sim U(0, 1), r_{0|1} = 100, r_{1|0} = 1$$

Shown here is the case where the sensor failure model is uniform. Namely,

$$P_{D_i|T,R_0}(1|1,1) = \alpha'_i \sim U(0,1), \quad P_{D_i|T,R_0}(0|0,1) = \beta'_i \sim U(0,1)$$

4.4.1 Performance Across Different $l$

As expected, our fusion rule is identical in the case where $l = 0$. Additionally, the expected risk of our fusion rule is always less than traditional fusion, see Figure 5.1.
4.4.2 Computational Challenges

A decrease in risk comes at a computational cost. The calculation of the $\sum_{r=0}^{S} C_i(d_{1:S}, r)$ term in the fusion requires $O(S!)$ calculations. One rudimentary method to manage such cost is impose an upper limit, $r_{\text{max}}$, on the maximum number of functions calculated in the sums. Such an estimation would yield the following fusion:

$$F'(D_{1:S}) = 1 \text{ when } L(D_{1:S}) + \log \frac{\sum_{r=0}^{r_{\text{max}}} C_1(d_{1:S}, r)}{\sum_{r=0}^{r_{\text{max}}} C_0(d_{1:S}, r)} > 0 \quad (4.10)$$

$$= 0 \text{ otherwise}$$

The cost, in risk, of such an estimation can be seen in Figure 5.1. The risk is identical for $r_{\text{max}} = 3, 4$ or $5$. In our experiments, using an $r_{\text{max}} = \frac{S}{2}$ resulted in little or no cost to the risk. As a future extension, we will investigate further estimations to reduce the computational complexity. For example, an adaptive method could create an $r_{\text{max}}$ for each individual $d_{1:S}$. Depending on the magnitude of $L(d_{1:S})$, it is reasonable to set $r_{\text{max}}$ dynamically, as the shape of $C_i(d_{1:S}, r)$ is reliably similar to its binomial multiplier.
CHAPTER 4. SOLUTION B: ROBUST FUSION FOR SENSOR FAILURE

4.4.3 Performance Across Different Number of Sensors

With a larger $S$, the relative performance advantage of our fusion rule grows (Figure 4.2). As the number of sensors increases there are more opportunities for sensors to fail. Because of this, mitigating the failure of sensors becomes increasingly important.

4.4.4 Performance Across Different $P_T(1)$

When $P_T(1)r_{0|1} = P_T(0)r_{1|0}$ the risk (of both the new and old fusion) is at a maximum. This is the case because the threshold term, $\log \frac{P_T(1)_{r_{0|1}}}{P_T(0)r_{1|0}}$, is closest to zero so that fusions are most dependent on the decisions, which themselves are possibly faulty. The risk lowers as the magnitude of this threshold term becomes greater. Furthermore, the relative advantage of our fusion increases as the magnitude of the threshold term increases, see Figure 4.3.

4.5 Discussion

Given a prior on how the joint distribution shifts during the course of operation, we develop a method which minimizes Expected Bayesian Risk. In this simple, Naive Bayes, binary case, the method relies only the first order characteristics of the joint
characteristics of sensor failure. This method can be thought of as a generalization of traditional Naive Bayes Fusion, and as a result the expected risk of traditional fusion is a strict lower bound. This performance, however, comes at the cost of increased computational cost. We develop a simple method for reducing the computational complexity of this new fusion rule and find it offer much of the benefit in decreasing the expected risk while limiting the computational time of the operation.
Chapter 5

Solution C: Identify Failed Sensors

The robust methods of 4 are useful in that they minimize the expected risk, but we still notice that sensor failure introduces a large amount of risk into a systems operation. It may be useful to run the system with the ”non-failure” joint distribution and incorporate an sensor failure detection scheme which would alert a user that the system if malfunctioning. The question of whether removal, robust-fusion, or no action should be taken in the event of failure must be taken on a case by case basis, being largely dependant on the relationship between a shift in the underlying joint distribution and the expected risk. Our focus, in this chapter, is simply to detect the event that a distribution has shifted out of its normal operating characteristic.

5.1 Other methods

To determine if a sensor has failed, it is common to perform hypothesis testing on the online feature variables with the null hypothesis, “these feature variables are drawn from the given joint distribution”. Certianly, if the distribution is known and parametrized, this problem has been well studied. We approach the more general case of non-parametric hypothesis testing.
5.2 Non-Parametric Method

We are interested in detecting sensor failures online (when deployed). As a result, any method we propose cannot use the target classes themselves in sensor failure detection. This is due to the fact that target classes are not available online. (In the case that target classes are known during operation, we could apply any of the typical online learning methods). Given that we cannot access threat classes, we restrict ourselves to using only the sensor decisions, $D_{1:S}$, themselves to determine if a failure has occurred. We proceed to describe the problem in two steps:

1. Detection of some sensor failure
2. Adjustment of Fusion to account for sensor failure

Given that sensor parameters cannot be learned in such a way which aids classification (Chapter 3), we could only remove failing sensors from fusion. Given our Naive Bayes assumption the removal of a subset of sensors is trivial.

5.2.1 Detection of Sensor Failure

We propose a method which compares the true joint distribution, $P_{D_{1:S}}$, with online estimates of it built using a sliding window of length $N$. Namely, we build a distribution over the Kullback-Liebler Divergence between the joint distribution, $P_{D_{1:S}}$, and the $M$ estimates of the joint ($P'_{D_{1:S}}$) each created from $N$ samples of $D_{1:S}$. From this distribution we can generate a test statistic, $a$, which we can use to either reject or accept the hypothesis that the joint sensor distribution has changed.

We mention, in passing, that there is a more appropriate choice for our particular example of Naive Bayes fusion. The joint distribution of $P_{D_{1:S}}$ is a multinomial distribution with known parameters. The optimal choice, given such a distribution, is to perform parametric hypothesis testing on the multinomial. We opt against such a method as it would restrict our joint distribution to be this multinomial. In order that our results be as applicable as possible to real-life explosives detection schemes, we have created a method which does not require such strong assumptions.


5.2.2 Results

For simulation, we implemented $S = 6$ identical sensors with:

$$\alpha_i = \beta_i = .9, P_T(1) = .1, r_{0|1} = r_{1|0}$$ (5.1)

After changing either $\alpha_1$ or $\beta_1$ we then generated data from their joint distribution and performed 100 Monte Carlo Simulations of our hypothesis testing.

Our method successfully detects failures which change a particular $\beta_i$ (fig 1). Selecting $\alpha$ at a sensitivity of .99, we notice that there is a no false alarm and a shift in $\beta$ of .2 results in reasonably high failure detections.

However, our method struggles to detect changes in $\alpha$ (fig 2). Even if we select
Figure 5.2: Detection of Sensor Failure in $\alpha$
a at a sensitivity of .99, we notice that there is still some false positive detection (roughly .03). Additionally, under all sensitivities detections of failure with a reasonable accuracy (.70) requires a shift in $\alpha$ of greater than .5. In short, to identify a change in $\alpha$ would require that it break to the point of severely biasing the ATR system and drastically diminish any discriminative capability.

This problem, however, is not limited to this approach. (We noticed a very similar result in our multinomial hypothesis testing). The issue is caused by the relatively small $P_T(1)$. In the example shown, we were generous in setting $P_T(1) = .1$, which would offer a very optimistic detection rate of failures in $\alpha$.

5.3 Discussion

We develop a method of non-parametric sensor failure detection. Our method succeeds at detecting the joint distribution of sensors under the high probability 'no target' class, even without parametrization. However, our method struggles with the 'target' class distribution shifts. Such a problem is common given that so few 'target' class events are observed.

Speaking plainly in the explosives detection context, it is a problem for sensor failure detection methods that so few passengers walk through ATR (Automatic Threat Recognition) systems with a "safe" threat class. For a failure detection algorithm to successfully find a sensor failure, it needs enough samples from the "dangerous" cases to compare. Any deployed ATR system will not have enough dangerous passengers walk through to find changes in their $\alpha$ parameter.

With this in mind, we suggest that some minimum testing be done by ATR systems to ensure that detection rates do not fall without being noticed. This may or may not be reasonable given the modality (iron bars to test metal detectors are safe at a checkpoint, while chemical agents to test mass spectroscopy machines would not be!).
Chapter 6

Artifact-Robust Fusion in RSVP Keyboard

RSVP keyboard is a method uses the p300 signal to detect a user’s selection among sequences of letters. The system can suffer from muscle signals in the users head such as an eye blink, jaw movement or smile. Each of these events adds noise to the EEG, corrupting the signal and impacting classification. In this chapter we build a method which fuses EEG features to detect the p300 which is robust to muscle movement artifacts.

6.1 RSVP Keyboard™

The RSVP Keyboard™ consists of four main components: visual presentation, feature extraction, language modeling and the classifier used to select a symbol.

6.1.1 Visual Presentation

RSVP is a presentation technique in which visual stimuli are displayed as a temporal sequence at a fixed location on the screen. An example screen snapshot from the current RSVP Keyboard prototype is given in Figure 6.1. In the current study, RSVP contains random permutations of the 26 letters in English alphabet, a space symbol...
and a backspace symbol (a total of 28 symbols to choose from). We use the term "sequence" to mean a showing of all 28 symbols. If repetition is needed, all symbols are repeated multiple times to improve classification accuracy until a preset desired confidence level or a maximum number of repetition is reached. The process of repetition of sequences to choose a single symbol is named as an epoch. In an epoch, we make the assumption that the user shows positive intent for a single symbol.

6.1.2 Feature Extraction

The feature extraction starts by extracting stimulus-time-locked bandpass filtered EEG signals for each stimulus in the sequence. Since physiologically, the most relevant signal components are expected to occur within the first 500ms following the stimuli, the [0,500] ms portion of the EEG following each stimulus is extracted. At this stage it is important to design bandpass filters whose group delay does not shift the physiological response to outside this interval. A linear dimension reduction is applied on the temporal signals using Principal Component Analysis in order to remove zero variance directions (i.e. zero-power bands based on the estimated covariance). The final feature vector to be classified is obtained as a concatenation of the PCA-projected temporal signals for each channel. Regularized Discriminant Analysis (RDA) [6] is used to further project the EEG evidence into scalar-feature for use in fusion with language model evidence.

RDA is a modification of quadratic discriminant analysis (QDA). QDA yields the optimal minimum-expectation-risk Bayes classifier under the assumption of multivariate
Gaussian class distributions. This classifier depends on the inverses of covariance matrices for each class, which are estimated from training data. To keep the calibration phase short few training samples are acquired - especially for the positive intent class. Therefore, the sample covariance estimates may become singular or ill-conditioned for high-dimensional feature vectors, which is the case here. RDA applies shrinkage and regularization on class covariance estimates. Shrinkage forces class covariances closer towards the overall data covariance as:

$$
\hat{\Sigma}_C(\lambda) = \frac{(1 - \lambda)\Sigma_C + \lambda \hat{\Sigma}}{(1 - \lambda)N_C + \lambda \hat{N}}
$$

(6.1)

Where $\lambda$ is the regularization parameter, $\Sigma_C, N_C$ are the class covariance estimate and number of samples for classes $C \in \{0, 1\}$ respectively. $C = 0$ is the non-p300 class. $\hat{\Sigma}, \hat{N}$ is the total covariance estimate and number of samples over all classes. Regularization is administered as:

$$
\hat{\Sigma}_C(\lambda, \gamma) = (1 - \gamma)\hat{\Sigma}_C(\lambda) + \frac{\gamma}{d} Tr[\hat{\Sigma}_C\lambda]I
$$

(6.2)

where $\gamma$ is the regularization parameter, $Tr[]$ is the trace function and $d$ is the dimension of the data vector.

After regularization and shrinkage, the covariance and mean estimates for each class are used in generating a scalar feature that minimizes expected risk under the Gaussianity assumption of class distributions. This is the log-likelihood ratio

$$
\delta_{RDA}(x) = \log \frac{f_N(x; \hat{\mu}_1, \hat{\Sigma}^1(\lambda, \gamma)\hat{\pi}_1)}{f_N(x; \hat{\mu}_0, \hat{\Sigma}^0(\lambda, \gamma)\hat{\pi}_0}
$$

(6.3)

where $\mu_c, \hat{\pi}_c$ are estimates of class means and priors respectively; $x$ is the data vector to be classified and $f_N(x; \mu, \Sigma)$ is the pdf of a multivariate Gaussian (normal) distribution.
6.1.3 Language Modeling

In letter-by-letter typing, we adopt an n-gram language models at the symbol level. These models estimate the conditional probability of a letter given by the $n - 1$ previously typed letters. In this study, a 6-gram model that is trained using a one-million sentence (210M character) sample of the NY Times portion of the English Gigaword corpus. Corpus normalization and smoothing methods are described in [13]. Finally, we note that the backspace symbol is assumed to have a constant conditional probability of 0.05 and the conditional probabilities of the other symbols are normalized accordingly.

6.1.4 Classifier

Using the class conditional score and the language model probabilities in a naive Bayes’ rule based fusion model, we compute the posterior probabilities of symbols given all the evidence. We compute these probabilities for each symbol after every sequence, and a decision is made if one symbol probability reaches a desired confidence level or number of repetitions exceeds a predefined limit.

6.2 Robust Classifier

In the classifier, the class conditional score distributions are used assuming that these distributions remain stationary during a typing session. However, possible changes in the distribution of the EEG data, possibly due to artifacts or sensor failure, should be incorporated in the score distribution. For example, as we also explain in Section 6.3, we apply our method on artifact reduction assuming artifacts as possible reasons for changes in the distribution. We introduce a variable $a$ which describes the artifact class of a particular trial. Artifact classes include a control group (no artifacts present), eye blink, jaw movement and smiling. For use in the language model fusion, we compute the score conditional distributions for the mixed conditional score distribution as

$$P(\delta_{RDA}(x)|c) = \Sigma_i P(\delta_{RDA}(x)|c, a_i)P(a_i)$$  \hspace{1cm} (6.4)
where, \( i \) is the artifact index, \( c = 0 \) or \( 1 \) is the class label, \( P(a_i) \) is the prior for artifact \( a_i \). For each class and artifact \( P(\delta_{RDA}(x)|c, a_i) \) is computed using (6.1.2).

### 6.3 Experiments

Four healthy operators participated in this study. For each subject, four RSVP sessions with pre-designated targets were performed using a 16-channel g.USBamp and g.Butterfly electrodes (g.Tec, Graz, Austria) in one sitting. The second session was the control session, while the first, third and fourth sessions had the subjects produce intentional jaw movement, eye blinks, and face muscle artifacts, respectively. Subjects continued to attend to the RSVP presentation during all sessions. This data is used to build and test robust and non-robust fusion models using 10-fold cross validation as explained in Section 6.2.

We perform Monte Carlo simulations on multiple pre-recorded calibration data sets to build kernel density estimates (KDEs) of the RDA score distribution for target symbol present and not-present conditions.

We select ten different sentences and aim to spell a phrase in each sentence (called the copy phrase task). Task difficulty is determined by requiring each letter of the target phrase to have a likelihood ratio against the highest likelihood competing non-target letter within a specified interval: (1) Hard: \((0.3, 0.5]\), (2) Very hard: \((0, 0.3]\).

In summary, we model typing performance by building a distribution of RDA scores from real training data under different artifact conditions. This model is then simulated typing 10 sentences 15 times to compare the performances of robust and non-robust classifiers. We report our results in terms of typing accuracy and duration (total seconds per word completion), see Figures 6.2 and 6.3. For reference, we include the area under the curve (AUC) values for each subject under all artifact conditions in 6.3.
### 6.3.1 Typing Accuracy

As can be noted in Figure 6.2, typing accuracy changes dramatically between subjects. In the simulation, as with other trials we’ve performed, subject 3 struggles to produce accurate classifications. Additionally, we note that robust classification consistently outperforms non-robust methods. The performance advantage of our method is correlated to the magnitude of the difference in AUC between the control and artifact classes. In other words, the stronger the drop in AUC when an artifact is introduced (Table 6.3), the greater the performance benefit of using robust fusion.

### 6.3.2 Typing Duration

From Figure 6.3, we immediately notice that the robust case typically types faster than the non-robust case. Additionally, considering the AUC values from Table 6.3
and the results from Figure 6.3, we notice that higher AUC values offer quicker typing performance. Both these effects share a common motivation. The typing system repeats sequences until a sufficiently high confidence threshold is reached. Accurate typing, because of robust methods or high user AUC, will yield fast typing.

6.4 Discussion

We designed a robust classification method for ERP detection in a BCI typing paradigm. We tested the proposed method on the RSVP Keyboard™, which is an in-house BCI typing system. Considering the possible changes in the EEG data, we developed a mixture density model for class conditional EEG evidence to use in the fusion with n-gram language model. To compare the robust and non-robust classification methods, using pre-recorded calibration data, we simulated the performance of four subjects typing 10 sentences 15 times and reported results on accuracy and speed of their typing.

Each of our simulations was run under a single artifact class (rather than a mixture
of multiple classes). When implemented with a true mixture of artifact classes we observed nearly identical results between robust and non-robust methods. We suggest this is due to the ability of our classifier to accumulate additional EEG evidence when an input doesn’t reach the confidence threshold. In the true mixture case, where artifacts aren’t very frequent, the classifier is bound to receive useful information during the following sequences. For further analysis, we are interested in examining the cause of mis-classified symbols. We hypothesize that the risk in artifacts is not readily seen in their prior distribution as artifacts frequently occur in bursts during operation. Future artifact class models which are conditioned on previous artifact classes, allowing for bursts of artifacts to occur while still keeping artifact priors at reasonable levels, will be studied.
Chapter 7

Conclusions

In this thesis we explore various methods of mitigating a shift in the distribution between feature and target variables. A first intuition might be to learn this shift online so that the system can learn the new distribution, essentially performing an online learning problem to accommodate the shift. We show in Chapter 3 that without any information besides the feature variables themselves, this approach cannot yield any improvement. In Chapter 4 we make use of assumptions about how the feature variables change. Given some distribution of how the sensors will perform in the event of a failure, as well as a probability that the sensors will fail, we develop a method which minimizes the Expected Bayesian Risk of our fusion rule. While as successful as possible, sensor failure can still introduce a large amount of risk into a classification. For this reason, it may be more useful to simply detect a sensor failure event. We develop a non-parametric way of doing so (Chapter 5). Each of these methods were proved using numerical methods. In the context of explosives detection, we would suggest that a more formal analysis of sensor failure and fusion be done on live data to determine the frequency and impact of sensor failures on fused systems. We then followed the theme of distribution shifts to the problem of EEG muscle artifacts in a p300 classification method. A method was built and simulated on live data which accepts some distribution of artifact events. Our method outperforms traditional fusion. In the future, we plan on exploring the relationship between expected risk, fusion rules and the joint distribution between feature and target variables in the
discrete case. (By fusion rule, we mean the mapping from all the possible feature vectors to all possible target variables). Let us define the function $b$ as a mapping from a distribution to the minimum yes risk classifier. We have developed a method which determines all the possible joint distributions which are mapped to a particular fusion rule under $b$. We are interested in how a shift in the distribution changes the expected risk within the pre-image of a particular fusion rule under $b$, and how that compares to the added risk associated with a shift in distribution between pre-images of different fusion rules under $b$. 
Bibliography


