Dimensionality Reduction for Online Learning Algorithms using Random Projections

Ran Gilad-Bachrach

School of Computer Science and Engineering
The Hebrew University, Jerusalem, Israel
ranb@cs.huji.ac.il

Abstract. Coping with high dimensional data is a challenge for machine learning. Modern methods, such as kernel machines, show that it is possible to learn concepts even in surprisingly high dimensional spaces. However, working in high dimensions takes it toll, both in computational complexity and in accuracy. A common way to overcome these deficiencies is to reduce the dimensionality of the data via some preprocessing. In the batch setting this procedure is commonly used, applying it in the online setting can be beneficial as well. However, working in the online setting requires some extra effort due to the nature of this model.
In this paper we present a dimensionality-reducing scheme for online learning. We construct a wrapper algorithm that uses an online learning algorithm as a subroutine. The wrapper projects the input data to a low dimensional space and applies the learning algorithm in its low dimensional representation. We demonstrate the effectiveness of this novel scheme by applying it to the online *Bayes Point Machine* (BPM) algorithm [1,2]. The use of dimensionality reduction allows us to obtain a dimension-free mistake bound for this algorithm which improves upon previous bounds [2].

1 Introduction

In many tasks to which machine learning is applied, the data are represented as vectors in high dimensional spaces. Some methods, such as kernel machines, appear to successfully challenge the "curse of dimensionality" [3] by embedding the data into infinite dimension spaces. These method work well in many cases. However, there is a price to pay when working in high dimensions. The dimension has noticeable effect on the prediction error of the classifier, and increases computational complexity. Overcoming these deficiencies requires some sort of dimensionality reduction.

Applying dimensionality-reduction schemes as preprocessing is a common practice in batch learning. A typical learning system will first analyze the original data to obtain a concise representation. The data, in reduced form, are then presented to a learning algorithm. In the online setting unlike the batch model,

The author is a student and is eligible for "Best Student Paper" award.

preprocessing of the data is forbidden. The dimensionality reduction must be pursued on the fly. To overcome this problem, we present a wrapper algorithm[1] which takes an online learning algorithm and wraps it.

Our novel dimensionality reduction scheme uses *random projections* [5] in its core. Johnson and Lindenstrauss showed that Euclidean distances between points are preserved when these points are projected to a low dimensional space along a random projection. This result was improved upon by Dasgupta and Gupta [6]. Random projections are widely used in various domains including machine learning. Arriaga and Vempala [7] used random projections to define robust concepts and study their properties. Balcan et al [8] combined kernel based large margin classifiers and random projections to obtain large margin classifiers in low dimensions. Kleinberg [9], Indyk and Motwani [10] and others used random projections to accelerate nearest neighbor classifiers. See [11,12,13,14,15] for more applications of random projections in machine learning.

Contrary to previous studies examining the use of random projections for learning, we are interested in the online learning model [16]. While in the batch model, data can be used to select the dimension to which to project the data, in online learning we have to "guess" the dimension in advance and make corrections as we go. We begin with a very low dimension and increase it until we find an adequate dimension. The increment size needs to be small enough to prevent "overshooting". On the other hand, the step size needs to be large enough so that the cumulative mistakes we make while searching will not be too high.

We present a wrapper algorithm that is appropriate for many online learning algorithms which have the Euclidean distance as their core. For brevity, we demonstrate it for the online *Bayes Point Machine* (BPM) algorithm [2,1]. We present the *Wrapped Bayes Point Machine* (WBPM) algorithm which takes the online version of the BPM algorithm and enhances it using a dimensionality reducing wrapper. The right choice of parameters yields a learning algorithm whose mistake bound is independent of the input dimension. This improves upon previous bounds for this algorithm [2].

We prove that the number of prediction mistakes WBPM will perform when predicting the labels of T instances is $O\left(\frac{1}{\widehat{\gamma}^2}\ln(T/\delta)\ln(1/\widehat{\gamma})\right)$ with probability $1-\delta$, where $\widehat{\gamma}$ is a margin term (see definition 1). WBPM achieves this mistake bound without knowing the margin or the length of the sequence (T) in advance. Surprisingly enough, if we would have known the margin and the length of the sequence, we would have obtained the same number of mistakes to within a constant factor.

2 Random Projections

The Johnson Lindenstrauss lemma [5] states that random projections of a vector in a Hilbert space tend to preserve its norm. We will use the following ramification:

[1] We borrow the term wrapper from Kohavi and John [4] who used this term in the feature selection setting.

Theorem 1. *Arriaga and Vempala [7]*

Let $x \in H$ and let $P : H \to R^d$ be a random matrix such that $p_{i,j} \sim N\left(0, \frac{1}{d}\right)$ and let $\epsilon > 0$ than

$$\Pr_P \left[1 - \epsilon \leq \frac{\|Px\|^2}{\|x\|^2} \leq 1 + \epsilon \right] \geq 1 - e^{-d\epsilon^2/8}$$

According to theorem 1, the norm of a vector has a small distortion when randomly projected. The distortion does not depend on the input dimension. However, the random projection is not a projection in the common mathematical sense, since $P^2 x \neq Px$.

Since we are interested in linear classifiers, the following theorem connects random projections and linear classifiers:

Theorem 2. *Let $x_1, \ldots, x_m$ be a set of points in a Hilbert space. Let $w_1, \ldots, w_k$ be a set of linear functionals. For any $\epsilon > 0$ if P is a random projection to a sub-space of dimension d then with probability $1 - (2mk + m + k)e^{-\frac{1}{200}d\epsilon^2}$ the following will hold:*

$$\forall i, j \ \left| \frac{w_j \cdot x_i}{\|w_j\| \, \|x_i\|} - \frac{Pw_j \cdot Px_i}{\|Pw_j\| \, \|Px_i\|} \right| \leq \epsilon \tag{1}$$

The proof of theorem 2 can be found in Appendix A. Next we present the online *Bayes Point Machine* algorithm to which we will apply the dimensionality reducing wrapper.

3 Online Bayes Point Machine

The *Bayes Point Machine* algorithm (BPM) [1] is an algorithm for learning linear classifiers. The algorithm uses the center of gravity of the version space as its hypothesis. In [2], this algorithm was analyzed both in the batch and online setting. For the online setting (see algorithm 1), the authors prove that the algorithm will perform at most

$$\frac{d}{-\ln(1 - 1/e)} \ln\left(\frac{2R}{\theta}\right) \tag{2}$$

where d is the input dimension, R is the radius of a ball that contains all input points and θ is a lower bound on the margin. Unlike the perceptron algorithm [17], the mistake bound of BPM indeed depends on the input dimension.

We begin by providing a slightly improved mistake bound than the one in [2]. For this purpose, we define *margin ratio*, which will play a crucial role in our discussion.

Definition 1. *Let $x, x_1, \ldots, x_T$ be points in a Hilbert space and let w be a linear functional. Let $y, y_1, \ldots, y_T \in \pm 1$ be labels. The* ***margin ratio*** *of w when predicting x is*

$$\gamma(w, x, y) = \frac{yw \cdot x}{\|x\| \, \|w\|}$$

Algorithm 1 Bayes Point Machine (online)

1. $V_1 \leftarrow$ the unit ball. % the original version space
2. for $t = 1, 2, \ldots$
   (a) Receive x_t
   (b) $z_t \leftarrow$ be the center of gravity of V_t
   (c) predict $\hat{y}_t = \text{sign}\left(z_t \cdot x_t\right)$
   (d) receive y_t
   (e) $V_{t+1} \leftarrow V_t \cap \{v : y_t v \cdot x_t \geq 0\}$ % update the version space
3. endfor

*The **margin ratio** of w when predicting the labels of $x_1, \ldots, x_T$ is*

$$\gamma\left(w\right) = \min_t \gamma\left(x_t, y_t\right)$$

There is an alternative way of defining the margin ratio of a hypothesis:

Definition 2. *Let $x, x_1, \ldots, x_T$ be points in a Hilbert space and let w be a linear functional. Let $y, y_1, \ldots, y_T \in \pm 1$ be labels. The **margin ratio** of w when predicting x is*

$$\gamma\left(w, x, y\right) = \left(\inf_{w' \text{ s.t. } sign(w \cdot x) \neq sign(w' \cdot x)} \left(\left\| \frac{w}{\|w\|} - w' \right\| \right) \right) \left(sign\left(yw \cdot x\right)\right)$$

*The **margin ratio** of w when predicting the labels of $x_1, \ldots, x_T$ is*

$$\gamma\left(w\right) = \min_t \gamma\left(x_t, y_t\right)$$

First we show that these two definitions are equivalent.

Lemma 1. *The two definitions of the margin ratio in definitions 1 and 2 are identical.*

Definition 1 is an algebraic definition whereas definition 2 provides a geometric interpretation for the margin ratio. Lemma 1 proves that the margin ratio is the *hypothesis margin* of a classifier [18]. It measures the stability of the classifier with respect to perturbations in the classifier itself. In a sense, a classifier with a large margin-ratio needs to be described in fewer detail. The proof of lemma 1 can be found in Appendix A.

In the next theorem we analyze the mistake bound of BPM in terms of the margin ratio. This provides an improvement over the previous bound proved in [2].

Theorem 3. *Let $x_1, x_2, \ldots, x_t \in R^d$ and let $y_1, y_2, \ldots, y_t \in \pm 1$. Assume that there exists a hypothesis $w_t^* \in R^d$ such that the margin ratio of w_t^* is $\widehat{\gamma}_t$. The number of mistakes BPM will make when predicting the labels $y_1, y_2, \ldots, y_t$ of $x_1, x_2, \ldots, x_t$ is at most*

$$\frac{d}{-\ln\left(1 - 1/e\right)} \ln\left(\frac{2}{\widehat{\gamma}_t}\right)$$

Theorem 3 improves over the result in [2]. This is so since if we denote by $R = \max \|x_t\|$ and $\theta = \min y_t w^* \cdot x_t$ clearly $\widehat{\gamma}_t \geq \theta/R$. By plugging in θ/R instead of $\widehat{\gamma}_t$ in theorem 3 we fall back to the result in [2]. See appendix A for the proof of the theorem.

The gap between $\widehat{\gamma}_t$ and θ/R can be significant. For example, consider the following data: Assume that $x_t \in \mathrm{R}^2$ such that $x_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ and $x_2, x_3, \ldots = \begin{pmatrix} -\epsilon \\ \epsilon \end{pmatrix}$ for some $\epsilon > 0$. Let $y_1, y_2, \ldots = +1$. In this case, the maximal margin ratio is $1/\sqrt{2}$. On the other hand $R = \max \|x_t\| = \sqrt{2}$ and $\theta = \sup_w \min_t \frac{w \cdot x_t}{\|w\|} \leq \epsilon\sqrt{2}$ which can be arbitrarily small. Indeed, BPM will make only two prediction mistakes on this data whereas the perceptron algorithm [17] will make $1/\epsilon$ prediction mistakes. Although, as we have demonstrated, the difference between $\widehat{\gamma}$ and θ/R can be significant, if the data are normalized before processing, i.e. if we apply the transformation $x_t / \|x_t\| \longrightarrow x'_t$ then this difference disappears. This leads us to the conclusion that if the perceptron is applied to normalized data, its mistake bound is $1/\widehat{\gamma}^2$ which is always smaller or equal to R^2/θ^2.

4 Introducing Random Projections to the Bayes Point Machine.

The merits of using random projections when learning linear classifiers have been presented by several authors, see for example [8]. Random projections were used as a preprocessing of the data in a batch learning task. When trying to use random projections for online learning, we cannot assume that we know the margin in advance nor do we know the size of the sample we will need to work with. This poses a problem when selecting the dimension to which we project the data. As a solution we present an adaptive method of selecting this dimension.

Algorithm 2 is a wrapper for BPM which has a dimension independent mistake bound. The *Wrapped Bayes Point Machine* (WBPM) algorithm uses random projection to a low dimensional space in which the BPM algorithm is used to make predictions. The dimension to which the data are projected depends on three factors: the number of predictions the algorithm will be required to make, the margin ratio $\widehat{\gamma}$ and a confidence parameter $1 - \delta$.

Assume for the moment that we knew in advance the number of predictions we would be required to make, t, and the best possible margin-ratio $\widehat{\gamma}_t$. Using theorem 2 we deduce that by selecting a projection P to a dimension $d = \frac{800}{\widehat{\gamma}_t^2} \ln \frac{3t}{\delta}$, we can project all the data using P and use BPM in the d-dimensional space preserving a margin ratio of $\widehat{\gamma}_t/2$ in the projected data. Using the mistake bound of BPM presented in theorem 3 we conclude that the algorithm will make at most

$$\frac{800}{\widehat{\gamma}_t^2 \ln(1 - 1/e)} \left(\ln \frac{3t}{\delta} \right) \left(\ln \frac{4}{\widehat{\gamma}_t} \right) = O\left(\frac{1}{\widehat{\gamma}_t^2} \ln \frac{t}{\delta} \ln \frac{1}{\widehat{\gamma}_t} \right) \tag{3}$$

mistakes, with a probability of $1 - \delta$ over the random choice of the projection. Thus, using random projection we obtain a dimension free bound for online BPM.

Algorithm 2 Wrapped Bayes Point Machine (WBPM)

The algorithm receives a confidence parameter $1-\delta$.

1. $\gamma_1 \leftarrow 1$ % expected margin ratio
2. $T_1 \leftarrow 1$ % expected number of prediction
3. **for** $i = 1, 2, \ldots$
   (a) $d_i \leftarrow \frac{800}{\gamma_i^2} \ln \frac{3T_i i(i+1)}{\delta}$
   (b) $P_i \leftarrow$ random projection to dimension d_i.
   (c) $e_i \leftarrow \frac{d_i}{-\log(1-1/e)} \log\left(\frac{4}{\gamma_i}\right)$.
   (d) reset the BPM algorithm
   (e) *mistakes* $\leftarrow 0$ % number of mistakes in current iteration
   (f) *predictions* $\leftarrow 0$ % number of predictions made in current iteration
   (g) **do**
      i. receive x
      ii. *predictions* $\leftarrow$ *predictions* $+ 1$
      iii. use BPM to predict the label $\hat{y}$ of $P_i x$
      iv. receive y
      v. update the BPM using $(P_i x, y)$.
      vi. **if** $y \neq \hat{y}$ **then** *mistakes* $\leftarrow$ *mistakes* $+ 1$
   (h) **while** (*mistakes* $\leq e_i$)
   (i) **if** *predictions* $> T_i$ **then**
      i. $T_{i+1} \leftarrow T_i^2 \frac{3e^{1/600}(i+1)(i+2)}{\delta}$
      ii. $\gamma_{i+1} \leftarrow \gamma_i$
   (j) **else**
      i. $T_{i+1} \leftarrow T_i$
      ii. $\gamma_{i+1} \leftarrow \gamma_i/\sqrt{2}$
   (k) **endif**
4. **endfor**

However, the main shortcoming of the above description lies in the fact that we do not know either the number of predictions we will be required to make: t, or the margin ratio: $\widehat{\gamma}$. We overcome these problems in the *Wrapped Bayes Point Machine* (WBPM) algorithm (see algorithm 2) by gradually adapting our estimates of these parameters. Whenever our assumption fails, which results mistakes in excess of the expect amount, we make a new estimate and calculate a new dimension to which we should project the data. We choose a new projection and restart BPM in the new dimension.

We claim, that although we do not know in advance either the number of predictions we will be asked to make or $\widehat{\gamma}$, the number of prediction mistakes of WBPM is comparable to the one we would have made if we had known these values in advance. This is the main conclusion of the following theorem

Theorem 4. *Let $\{(x_\tau, y_\tau)\}_{\tau=1}^{\infty}$ be a dataset. Let $\widehat{\gamma}_t$ be the maximal possible margin ratio for the prefix $\{(x_\tau, y_\tau)\}_{\tau=1}^{t}$ of the data. With probability $1-\delta$ over the internal randomness of WBPM, for any t, the number of prediction mistakes*

that WBPM will make for this prefix is at most

$$\frac{9600}{-\widehat{\gamma}_t^2 \ln(1-1/e)} \ln\left(\frac{(t+1)\, e^{1/100}}{\delta}\right) \ln\left(\frac{4\sqrt{2}}{\widehat{\gamma}_t}\right) + 2 = O\left(\frac{1}{\widehat{\gamma}_t^2} \ln\left(\frac{t}{\delta}\right) \ln\left(\frac{1}{\widehat{\gamma}_t}\right)\right)$$

We construct the proof using a set of simple lemmas which we combine together to prove theorem 4. The proofs of these lemmas can be found in Appendix A.

Lemma 2. *Using the notation of theorem 4, with probability* $1-\delta$ *over the internal randomness of WBPM: if WBPM is in its* i*'th iteration when predicting the label of* x_t *then*

$$T_i \leq t^2 \frac{3e^{1/600} i\,(i+1)}{\delta} \tag{4}$$

and

$$\gamma_i \geq \widehat{\gamma}_t / \sqrt{2} \tag{5}$$

Lemma 3. *For* $i = 1, 2, \ldots$ *the following holds:*

$$e_{i+1} \geq 2\,(e_i + 1)$$

Lemma 4. *Using the notation of theorem 4, with probability* $1-\delta$ *over the internal randomness of WBPM, for all* t*: if WBPM is in its* i*'th iteration when predicting the label of* x_t *then*

$$e_i \leq \frac{4800}{-\widehat{\gamma}_t^2 \ln(1-1/e)} \ln\left(\frac{(t+1)\, e^{1/100}}{\delta}\right) \ln\left(\frac{2\sqrt{2}}{\widehat{\gamma}_t}\right)$$

Finally, we are ready to provide the proof for theorem 4.

Proof. (of theorem 4)

Assume that WBPM is at its i'th iteration when predicting the label of x_t. In the i'th iteration, it will make at most $e_i + 1$ mistakes before proceeding to the $(i+1)$'th iteration. Thus, the number of prediction mistakes that WBPM have done so far is bounded by $\sum_{j=1}^{i} (e_j + 1)$. Using lemma 3 we have that

$$\sum_{j=1}^{i} (e_j + 1) \leq 2\,(e_i + 1) \tag{6}$$

Plug in the result of lemma 4 in (6) to obtain the stated result. □

5 Discussion

Theorem 4 provides a mistake bound for WBPM which is dimension free. The difficulty in the algorithm and its proof comes from the fact that we do not know either the horizon we will be looking at (t), or the margin ratio ($\widehat{\gamma}_t$) in advance. Nevertheless, the bound we obtained for WBPM in theorem 4 is qualitatively the same as the one obtained in (3) when both t and $\widehat{\gamma}_t$ are known.

The techniques used in the WBPM algorithm and its proof extend beyond the BPM algorithm. These techniques can be used to enrich other learning algorithm with dimensionality reduction. Our wrapping method is suitable for many online learning algorithms which use Euclidean distances in their core. In order to do so, the update steps (steps 3(i)i and 3(j)ii) should be modified in a way that will guarantee that the updates are small, but on the other hand $e_{i+1} \geq 2\left(e_i + 1\right)$. The calculation of e_i (step 3c) should reflect the mistake bound of the algorithm at hand.

However, the method presented here has its limitations. First note that the projections P_i need to be explicitly given. This poses a computational problem when the input dimension is extremely large. Another caveat of this method is that it is not applicable in all metric systems. For example, in the l_1 norm space it is impossible to find low-dimensional embedding of points which will preserve pairwise distances [19]. Hence applying our dimensionality reducing wrapper with algorithms such as Winnow [16] might not work.

Theorem 2 has implications that go beyond online learning. Assume $x_1, \ldots, x_m$ is a finite sample. If this sample is separable with margin ratio $\widehat{\gamma}$ then a random projection of this sample to a space of dimension $d = O\left(\frac{1}{\widehat{\gamma}^2} \ln\left(m\right)\right)$ will also be separable with margin $\sim \widehat{\gamma}$. This is the main idea in many preprocessing methods for batch learning (see [8] for example). This can be further extended when multiple classifiers are used. Consider for example the learning a multi-categorical classification task. It is common practice to split such a learning problem into a multiple binary classification tasks [20]. When doing so, we would like to preserve large margin for all binary classifiers at once. From theorem 2 it follows that this can achieved by projecting the data to a space of dimension $d = O\left(\frac{1}{\widehat{\gamma}^2} \ln\left(mk\right)\right)$ where k is the number of binary classifiers. Finally we can project $x_1, \ldots, x_m$ to a space of dimension $d = O\left(\frac{1}{\widehat{\gamma}^4} \ln^2\left(m\right)\right)$, in this case all possible linear bisections of the data with margin $\widehat{\gamma}$ will be preserved.

Acknowledgments

We thank Claudio Gentile, Naftali Tishby and Amir Navot for fruitful discussions. The author thanks the Clore foundation for financial support.

References

1. Herbrich, R., Graepel, T., Campbell, C.: Bayes point machines. Journal of Machine Learning Research (2001)

2. Gilad-Bachrach, R., Navot, A., Tishby, N.: Bayes and tukey meet at the center point. In: Proc. 17'th Conference on Learning Theory (COLT). (2004) 549–563
3. Bellman, R.: Adaptive Control Processes: A Guided Tour. Princeton University Press (1961)
4. Kohavi, R., John, G.: Wrapper for feature subset selection. Artificial Intelligence **97** (1997) 273–324
5. Johnson, W., Lindenstrauss, J.: Extensions of lipschitz maps into a hilbert space. In: Proceedings of the Conference in modern analysis and probability. (1984) 189–206
6. Dasgupta, S., Gupta, A.: An elementary proof of a theorem of johnson and lindenstrauss. Random Structures and Algorithms **22** (2003) 60–65
7. Arriaga, R., Vempala, S.: An algorithmic theory of learning: Robust concepts and random projection. In: IEEE Symposium on Foundations of Computer Science. (1999) 616–623
8. Balcan, M.F., Blum, A., Vempala, S.: Kernels as features: On kernels, margins, and low-dimensional mappings. In: In the proceedings of the 15'th International Conference on Algorithmic Learning Theory (ALT '04). (2004) 194–205
9. Kleinberg, J.: Two algorithms for nearest-neighbor search in high dimensions. In: Proceedings of the 29th ACM Symposioum on Theory of Computing (STOC). (1997)
10. Indyk, P., Motwani, R.: Approximate nearest neighbors: towards removing the curse of dimensionality. In: Proceedings of the 30th ACM Symposium on the Theory of Computing. (1998) 604–613
11. Bingham, E., Mannila, H.: Random projection in dimensionality reduction: applications to image and text data. In: in proceeding of the The 7'th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD). (2001) 245–250
12. Dasgupta, S.: Learning mixtures of gaussians. In: Proceedings of the 14th Annual IEEE Symposium on Foundations of Computer Science (FOCS). (1999)
13. Fradkin, D. abd Madigan, D.: Experiments with random projections for machine learning. In: in proceeding of the The 9'th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD). (2003)
14. Garg, A., Har-Peled, S., Roth, D.: On generalization bounds, projection profile, and margin distribution. In: in Proceeding 19'th International Conference on Machine Learning (ICML). (2002)
15. Garg, A., Roth, D.: Margin distribution and learning algorithms. In: Proceedings of the International Conference on Machine Learning (ICML). (2003)
16. Littlestone, N.: Mistake Bounds and Logarithmic Linear-threshold Learning Algorithms. PhD thesis, University of California Santa Cruz (1989)
17. Novikoff, A.: On convergence proofs on perceptrons. In: Proceedings of the Symposium on the Mathematical Theory of Automata. Volume 12. (1962) 615–622
18. Crammer, K., Gilad-Bachrach, R., Navot, A., Tishby, N.: Margin analysis of the lvq algorithm. In: Proc. 17'th NIPS. (2002)
19. Charikar, M., Sahai, A.: Dimension reduction in the l1 norm. In: Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS). (2002) 551–560
20. Crammer, K., Singer, Y.: On the learnability and design of output codes for multiclass problems. Machine Learning (2002)
21. Grunbaum, B.: Partition of mass-distributions and convex bodies by hyperplanes. Pacific Journal of Mathmatics **10** (1960) 1257–1261

A Supplementary Material

Proof. of theorem 2

Assume that the following properties hold for the projection P and any choice of i and j:

$$1-\frac{\epsilon}{5} \leq \frac{\left\|\frac{Pw_j}{\|w_j\|}-\frac{Px_i}{\|x_i\|}\right\|^2}{\left\|\frac{w_j}{\|w_j\|}-\frac{x_i}{\|x_i\|}\right\|^2} \leq 1+\frac{\epsilon}{5} \tag{7}$$

$$1-\frac{\epsilon}{5} \leq \frac{\left\|\frac{Pw_j}{\|w_j\|}+\frac{Px_i}{\|x_i\|}\right\|^2}{\left\|\frac{w_j}{\|w_j\|}+\frac{x_i}{\|x_i\|}\right\|^2} \leq 1+\frac{\epsilon}{5} \tag{8}$$

Then

$$\left|\frac{Pw_j \cdot Px_i}{\|w_j\|\,\|x_i\|}-\frac{w_j \cdot x_i}{\|w_j\|\,\|x_i\|}\right| \tag{9}$$

$$\begin{aligned}
&= \frac{1}{4}\left|\left\|\frac{Pw_j}{\|w_j\|}+\frac{Px_i}{\|x_i\|}\right\|^2-\left\|\frac{Pw_j}{\|w_j\|}-\frac{Px_i}{\|x_i\|}\right\|^2-\left\|\frac{w_j}{\|w_j\|}+\frac{x_i}{\|x_i\|}\right\|^2+\left\|\frac{w_j}{\|w_j\|}-\frac{x_i}{\|x_i\|}\right\|^2\right| \\
&\leq \frac{1}{4}\left|\left\|\frac{Pw_j}{\|w_j\|}+\frac{Px_i}{\|x_i\|}\right\|^2-\left\|\frac{w_j}{\|w_j\|}+\frac{x_i}{\|x_i\|}\right\|^2\right|+\frac{1}{4}\left|\left\|\frac{Pw_j}{\|w_j\|}-\frac{Px_i}{\|x_i\|}\right\|^2-\left\|\frac{w_j}{\|w_j\|}-\frac{x_i}{\|x_i\|}\right\|^2\right| \\
&\leq \frac{\epsilon}{20}\left\|\frac{w_j}{\|w_j\|}+\frac{x_i}{\|x_i\|}\right\|^2+\frac{\epsilon}{20}\left\|\frac{w_j}{\|w_j\|}-\frac{x_i}{\|x_i\|}\right\|^2 \\
&\leq \frac{2}{5}\epsilon
\end{aligned} \tag{10}$$

We further assume that for any i and j

$$1-\frac{\epsilon}{5} \leq \frac{\|Px_i\|^2}{\|x_i\|^2} \leq 1+\frac{\epsilon}{5} \tag{11}$$

$$1-\frac{\epsilon}{5} \leq \frac{\|Pw_j\|^2}{\|w_j\|^2} \leq 1+\frac{\epsilon}{5} \tag{12}$$

and thus we have

$$\left|\frac{Pw_j \cdot Px_i}{\|Pw_j\|\,\|Px_i\|}-\frac{Pw_j \cdot Px_i}{\|w_j\|\,\|x_i\|}\right| \tag{13}$$

$$= |Pw_j \cdot Px_i|\left|\frac{\|w_j\|\,\|x_i\|-\|Pw_j\|\,\|Px_i\|}{\|w_j\|\,\|x_i\|\,\|Pw_j\|\,\|Px_i\|}\right|$$

$$
\begin{aligned}
&\leq \frac{|Pw_j \cdot Px_i|}{\|w_j\| \|x_i\| \|Pw_j\| \|Px_i\|} \left(\|w_j\| \left|\|x_i\| - \|Px_i\|\right| + \|Px_i\| \left|\|w_j\| - \|Pw_j\|\right|\right) \\
&\leq \frac{|Pw_j \cdot Px_i|}{\|w_j\| \|x_i\| \|Pw_j\| \|Px_i\|} \left(\|w_j\| \left(\sqrt{1+\frac{\epsilon}{5}} - 1\right) \|x_i\| + \|Px_i\| \left(\sqrt{1+\frac{\epsilon}{5}} - 1\right) \|w_j\|\right) \\
&\leq \frac{|Pw_j \cdot Px_i|}{\|w_j\| \|x_i\| \|Pw_j\| \|Px_i\|} \left(\|w_j\| \|x_i\| \left(\sqrt{1+\frac{\epsilon}{5}} - 1\right) \left(2+\frac{\epsilon}{5}\right)\right) \\
&= \frac{|Pw_j \cdot Px_i|}{\|Pw_j\| \|Px_i\|} \left(\left(\sqrt{1+\frac{\epsilon}{5}} - 1\right) \left(2+\frac{\epsilon}{5}\right)\right) \\
&\leq \left(\sqrt{1+\frac{\epsilon}{5}} - 1\right) \left(2+\frac{\epsilon}{5}\right) && (14) \\
&\leq \frac{\epsilon}{5} \left(2+\frac{\epsilon}{5}\right) && (15) \\
&\leq \frac{3}{5}\epsilon && (16)
\end{aligned}
$$

where (15) follows since $\left(\sqrt{1+\epsilon} - 1\right) \leq \epsilon$ and (16) follows since w.l.o.g. $\epsilon \leq 5$. Combining (10) and (16) we obtain

$$
\left| \frac{Pw_j \cdot Px_i}{\|Pw_j\| \|Px_i\|} - \frac{w_j \cdot x_i}{\|w_j\| \|x_i\|} \right| \leq \epsilon \tag{17}
$$

However, (17) holds only when (7), (8), (11) and (12) hold. Using theorem 1 and the union bound we conclude that this conditions holds with a probability of $1 - (2mk + m + k)e^{-d\epsilon^2/200}$. □

Proof. of lemma 1

It suffices that the two definitions of $\gamma(w, x, y)$ in definition 1 and definition 2 be equivalent. First note that the sign of $\gamma(w, x, y)$ in both definitions is equivalent. Furthermore, note that if we flip the sign of y than $\gamma(w, x, y)$ flips it sign. Hence it suffices to show that

$$
\inf \left\{ \left\| \frac{w}{\|w\|} - w' \right\| : \operatorname{sign}(w \cdot x) \neq \operatorname{sign}(w' \cdot x) \right\} = \frac{|w \cdot x|}{\|w\| \|x\|}
$$

Let $\epsilon > 0$ and let $w' = \frac{w}{\|w\|} - (1+\epsilon) \frac{|w \cdot x|}{\|w\| \|x\|^2} x$. We have that

$$
w' \cdot x = -\epsilon \frac{w \cdot x}{\|w\|}
$$

and thus $\operatorname{sign}(w \cdot x) \neq \operatorname{sign}(w' \cdot x)$. For this choice of w' we have

$$
\left\| \frac{w}{\|w\|} - w' \right\| = (1+\epsilon) \frac{|w \cdot x|}{\|w\| \|x\|}
$$

Since this is true for any $\epsilon > 0$ we conclude that

$$
\inf \left\{ \left\| \frac{w}{\|w\|} - w' \right\| : \operatorname{sign}(w \cdot x) \neq \operatorname{sign}(w' \cdot x) \right\} \leq \frac{|w \cdot x|}{\|w\| \|x\|} \tag{18}
$$

On the other hand, let w' be such that $\operatorname{sign}(w \cdot x) \neq \operatorname{sign}(w' \cdot x)$. We have that

$$\begin{aligned}\left\| \frac{w}{\|w\|} - w' \right\| &\geq \left| \left(\frac{w}{\|w\|} - w' \right) \cdot \frac{x}{\|x\|} \right| \\ &= \left| \frac{w \cdot x}{\|w\| \, \|x\|} - \frac{w' \cdot x}{\|x\|} \right| \\ &\geq \left| \frac{w \cdot x}{\|w\| \, \|x\|} \right|\end{aligned}$$

where the last inequality follows from $\operatorname{sign}\left(\frac{w \cdot x}{\|w\|\|x\|}\right) \neq \operatorname{sign}\left(\frac{w' \cdot x}{\|x\|}\right)$. Therefore we have

$$\inf \left\{ \left\| \frac{w}{\|w\|} - w' \right\| : \operatorname{sign}(w \cdot x) \neq \operatorname{sign}(w' \cdot x) \right\} \geq \frac{|w \cdot x|}{\|w\| \, \|x\|} \tag{19}$$

Combining (18) with (19) we complete the proof of the lemma. □

Proof. of theorem 3

The proof follows the same path as the proof in [2]. Grunbaum [21] proved that any linear cut through the center of gravity of a convex body cuts its body into two pieces, each has at least $1/e$ of the volume of the original body. Hence the volume of the version space V_t decreases by a factor of $1 - 1/e$ at least whenever BPM makes prediction mistake. Thus if BPM made k prediction mistakes, then the volume of the version space is at most $(1 - 1/e)^k$ of its original size. Since the original version space is the unit ball B_d then the volume of the version space after misclassifying k times is at most $\operatorname{vol}(B_d)(1 - 1/e)^k$.

On the other hand since there exists a classifier w_t^* with a margin ratio of $\widehat{\gamma}_t$. It follows from lemma 1 that there exists a ball of radius $\widehat{\gamma}_t$ of classifiers around $\frac{w_t^*}{\|w_t^*\|}$ such that all the classifiers in this ball classify correctly $x_1, \ldots, x_t$. Inside this ball, there is a smaller ball of radius $\widehat{\gamma}_t/2$ such that all the classifiers in the smaller ball are both consistent classifiers and at the same time lie in the unit ball. Thus, the version space must contain this ball of radius $\widehat{\gamma}_t/2$. Note that the volume of this ball is $\operatorname{vol}(B_d)(\widehat{\gamma}_t/2)^d$ thus we obtain

$$\operatorname{vol}(B_d)(1 - 1/e)^k \geq \operatorname{vol}(B_d)(\widehat{\gamma}_t/2)^d$$

and thus

$$\frac{k}{-\ln(1 - 1/e)} \leq d \ln(\widehat{\gamma}_t/2)$$

□

Proof. of lemma 2

We will use induction to prove the statement of this lemma. In order to simplify the notation, we denote by T_t and γ_t the T and γ used by WBPM when predicting the label of x_t. In other words, if WBPM was at its i'th iteration then $T_t = T_i$ and $\gamma_t = \gamma_i$.

First, lets look at (4). For $t = 1$ it holds as $i = 1$ and a simple calculation shows that (4) and (5) hold. Lets assume that (4) holds for t and show that it holds for $t + 1$. We consider three cases. First assume that both T and γ were not updated between time t and time $t + 1$. In this case we have

$$T_{t+1} = T_t \leq t^2 \frac{3e^{1/600} i\,(i+1)}{\delta} < (t+1)^2 \frac{3e^{1/600} i\,(i+1)}{\delta}$$

The second case to consider is when between time t and time $t + 1$, there was an update step but only γ was updated while T remained the same (step 3(j)ii). In this case we have

$$T_{t+1} = T_t \leq t^2 \frac{3e^{1/600}\,(i-1)\,(i)}{\delta} < (t+1)^2 \frac{3e^{1/600} i\,(i+1)}{\delta}$$

Now assume that $T_{t+1} \neq T_t$ then from step 3(i)i it follows that $T_{t+1} = T_t^2 \frac{3e^{1/600} i(i+1)}{\delta}$. Furthermore, this update took place since there were more than T_t predictions made since T_t was updated. Therefore, $T_t \leq t$ and thus

$$T_{t+1} = T_t^2 \frac{3e^{1/600} i\,(i+1)}{\delta} \leq t^2 \frac{3e^{1/600} i\,(i+1)}{\delta} < (t+1)^2 \frac{3e^{1/600} i\,(i+1)}{\delta}$$

which asserts that (4) holds for any t.

In order to assert that (5) holds we first note the following: d_i is chosen such that with probability $1 - \frac{\delta}{i(i+1)}$ the projection P_i will satisfy the conditions in theorem 2 with $\epsilon = \gamma_i/2$, $k = 1$ and $m = T_i$. By summing over i we have that with probability $1 - \delta$, the conditions in theorem 2 hold for all i. Hence we can assume that we are in the situation where these conditions hold.

We would like to show that (5) holds, i.e. $\gamma_t \geq \widehat{\gamma}_t/\sqrt{2}$. We use induction to show this. For $t = 1$ (5) holds as $\gamma_1 = 1$ which is the maximal possible margin ratio. Assume that the condition holds up to time t and we will show that it holds at time $t + 1$. We consider two cases: if $\gamma_{t+1} = \gamma_t$ the condition holds as

$$\gamma_{t+1} = \gamma_t \geq \widehat{\gamma}_t/\sqrt{2} \geq \widehat{\gamma}_{t+1}/\sqrt{2}$$

where the rightmost inequality follows since $\widehat{\gamma}_t$ is non-increasing. The second case is when $\gamma_{t+1} \neq \gamma_t$, in this case $\gamma_{t+1} = \gamma_t/\sqrt{2}$. WBPM updated the value of γ_{t+1} (step 3(j)ii) since the number of prediction mistakes made since the last update exceeded e_t while making fewer than T_t predictions (again we abuse the notation and use e_t as the value of e_i when making the prediction for x_t). According to the assumption that the conditions of theorem 2 hold, it follows that $\gamma_t > \widehat{\gamma}_t$ as though the original data points were separable with margin ratio γ_t then the projected data points were separable with margin ratio $\gamma_t/2$. In this case, according to the mistake bound for BPM (theorem 3), the number of prediction mistakes should not exceed $\frac{d_t}{\ln(1-1/e)} \ln \frac{2}{\gamma_t}$ which is exactly e_t. Thus the original data are not separable with margin γ_t and thus $\gamma_t > \widehat{\gamma}_t$. Therefore we have that

$$\gamma_{t+1} = \gamma_t/\sqrt{2} > \widehat{\gamma}_t/\sqrt{2} \geq \widehat{\gamma}_{t+1}/\sqrt{2}$$

Again, the rightmost inequality follows since $\widehat{\gamma}$ is non-increasing. □

Proof. of lemma 3

From the definition of e_i and d_i it follows that

$$e_i = \frac{800}{-\gamma_i^2 \ln(1-1/e)} \ln\left(\frac{3T_i i(i+1)}{\delta}\right) \ln\left(\frac{4}{\gamma_i}\right)$$

We consider two cases: first assume that $\gamma_{i+1} = \gamma_i$ while $T_{i+1} = T_i^2 \frac{3e^{1/600}(i+1)(i+2)}{\delta}$. In this case we have

$$\begin{aligned}
e_{i+1} &= \frac{800}{-\gamma_{i+1}^2 \ln(1-1/e)} \ln\left(\frac{3T_{i+1}(i+1)(i+2)}{\delta}\right) \ln\left(\frac{4}{\gamma_{i+1}}\right) \\
&= \frac{800}{-\gamma_i^2 \ln(1-1/e)} \ln\left(\frac{3^2 T_i^2 (i+1)^2 (i+2)^2}{\delta^2} e^{1/600}\right) \ln\left(\frac{4}{\gamma_i}\right) \\
&= 2e_i + \frac{800}{-600\gamma_i^2 \ln(1-1/e)} \ln\left(\frac{4}{\gamma_i}\right) \\
&> 2e_i + 2
\end{aligned}$$

where the last inequality follows since $\gamma_i \leq 1$.

We now consider the second case in which $T_{i+1} = T_i$ while $\gamma_{i+1} = \gamma_i/\sqrt{2}$. In this case we have

$$\begin{aligned}
e_{i+1} &= \frac{800}{-\gamma_{i+1}^2 \ln(1-1/e)} \ln\left(\frac{3T_{i+1}(i+1)(i+2)}{\delta}\right) \ln\left(\frac{4}{\gamma_{i+1}}\right) \\
&= 2\frac{800}{-\gamma_i^2 \ln(1-1/e)} \ln\left(\frac{3T_i(i+1)(i+2)}{\delta}\right) \ln\left(\frac{4\sqrt{2}}{\gamma_i}\right) \\
&\geq 2e_i + 2\frac{800}{-\gamma_i^2 \ln(1-1/e)} \ln\left(\frac{3T_i(i+1)(i+2)}{\delta}\right) \ln\sqrt{2} \\
&\geq 2e_i + 2\frac{800}{-\ln(1-1/e)} \ln(6) \ln\sqrt{2} \\
&> 2e_i + 2
\end{aligned}$$

□

Proof. of lemma 4

Recall that

$$\begin{aligned}
e_i &= \frac{d_i}{-\ln(1-1/e)} \ln\left(\frac{4}{\gamma_i}\right) \\
&= \frac{800}{-\gamma_i^2 \ln(1-1/e)} \ln\left(\frac{3T_i i(i+1)}{\delta}\right) \ln\left(\frac{4}{\gamma_i}\right)
\end{aligned}$$

Following lemma 2 we have that $T_i \leq t^2 \frac{3e^{1/600} i(i+1)}{\delta}$ and $\gamma_i \geq \widehat{\gamma}_t/\sqrt{2}$. Since $i \leq t$ we have

$$e_i \leq \frac{800}{-\gamma_i^2 \ln(1-1/e)} \ln\left(\frac{t^2 3e^{1/600} i^2 (i+1)^2}{\delta^2}\right) \ln\left(\frac{4}{\gamma_i}\right)$$

$$
\begin{aligned}
&\leq \frac{800}{-\gamma_i^2 \ln(1-1/e)} \ln\left(\frac{(t+1)^6\, 3e^{1/600}}{\delta^2}\right) \ln\left(\frac{4}{\gamma_i}\right) \\
&\leq \frac{4800}{-\gamma_i^2 \ln(1-1/e)} \ln\left(\frac{(t+1)\, e^{1/100}}{\delta}\right) \ln\left(\frac{4}{\gamma_i}\right) \\
&\leq \frac{4800}{-\widehat{\gamma}_t^2 \ln(1-1/e)} \ln\left(\frac{(t+1)\, e^{1/100}}{\delta}\right) \ln\left(\frac{4\sqrt{2}}{\widehat{\gamma}_t}\right)
\end{aligned}
$$

□

Lemma 5. *Using the notation of theorem 4, assume that WBPM is in its i'th iteration when predicting the label of x_t then*

$$
i \leq \log_2 \log_2 \left(T_i 6 e^{1/4}\right) - 2 \log_2 \gamma_i + 1
$$

Proof. We argue that the number of times that T was updated by WBPM (step 3(i)i) is at most $\log_2 \log_2 \left(T_i 6 e^{1/4}\right)$ and the number of times γ was updated (step 3(j)ii) is exactly $-2 \log_2 \gamma_i$ and thus the statement of this lemma follows.

First, if T_i was never updated, then $T_i = 1$ and thus $\log_2 \log_2 \left(T_i 6 e^{1/4}\right) \sim 0.71$ which upper bounds the number of times T_i was updated. Assume that for some $i \geq 1$ indeed $\log_2 \log_2 \left(T_i 6 e^{1/4}\right)$ upper bounds the number of times T_i was updated. Furthermore, assume that between iterations i and $i+1$, T was updated then

$$
\begin{aligned}
\log_2 \log_2 \left(T_{i+1} 6 e^{1/4}\right) &= \log_2 \log_2 \left(T_i^2 \frac{e^{1/4}\,(i_0+1)\,(i_0+2)}{\delta} 6 e^{1/4}\right) \\
&\geq \log_2 \log_2 \left(T_i 6 e^{1/4}\right)^2 \\
&= \log_2 2 \log_2 \left(T_i 6 e^{1/4}\right) \\
&= 1 + \log_2 \log_2 \left(T_i 6 e^{1/4}\right)
\end{aligned}
$$

Since according to the assumption $\log_2 \log_2 \left(T_i 6 e^{1/4}\right)$ upper bounds the number of times T_i was updated than $\log_2 \log_2 \left(T_{i+1} 6 e^{1/4}\right)$ upper bounds the number of times T_{i+1} was updated.

We would like to show that the number of times γ was updated is exactly $-2 \log_2 \gamma$. To see this note that if rat_i was updated k times then $\gamma_i = 2^{-k/2}$ thus

$$
-2 \log_2 \gamma_i = k
$$

as stated.

Since we have shown that $\log_2 \log_2 \left(T_i 6 e^{1/4}\right)$ upper bounds the number of times T was updated, and $-2 \log_2 \gamma_i$ is exactly the number of times γ was updated, and since in each iteration one of T and γ is updated, it follows that

$$
i \leq \log_2 \log_2 \left(T_i 6 e^{1/4}\right) - 2 \log_2 \gamma_i + 1
$$

□