Links Between Binary Classification and the Assignment Problem in Ordered Hypothesis Machines

Reid Porter[a], Beate G. Zimmer[b]
[a]Intelligence and Space Research Division, Los Alamos National Laboratory,
Los Alamos, New Mexico, USA 87545
[b]Dept. of Mathematics and Statistics, Texas A&M University - Corpus Christi,
Corpus Christi, Texas, USA 78412.

ABSTRACT

Ordered Hypothesis Machines (OHM) are large margin classifiers that belong to the class of Generalized Stack Filters which were originally developed for non-linear signal processing. In previous work we showed how OHM classifiers are equivalent to a variation of Nearest Neighbor classifiers, with the advantage that training involves minimizing a loss function which includes a regularization parameter that controls class complexity. In this paper we report a new connection between OHM training and the Linear Assignment problem, a combinatorial optimization problem that can be solved efficiently with (amongst others) the Hungarian algorithm. Specifically, for balanced classes, and particular choices of parameters, OHM training is the dual of the Assignment problem. The duality sheds new light on the OHM training problem, opens the door to new training methods and suggests several new directions for research.

Keywords: binary classification, stack filters, assignment problem, bipartite matching

1. INTRODUCTION

The linear assignment problem is one of the most famous and well-studied optimization problems in the field of operations research. It focuses on how best to assign a set of N items (e.g. jobs or tasks) to a set of N other items (e.g. workers or machines) [1]. Assignments are often described in terms of a permutation which uniquely maps each job to a worker:

$$\varphi : (1,2,\dots,N) \to \big(\varphi(1),\varphi(2),\dots,\varphi(N)\big) \tag{1}$$

The permutation φ belongs to the set of all possible permutations of N items, denoted ϕ_N, which has $N!$ elements. In the linear assignment problem there is a nonnegative cost $c\big(i,\varphi(i)\big)$ associated with each potential assignment of job i to worker $\varphi(i)$, and the objective is to find the permutation that minimizes the sum of assignment costs:

$$\hat{\varphi} \in \ argmin_{\varphi\in\phi_N} \sum_{i=1}^{N} c\big(i,\varphi(i)\big) \tag{2}$$

Binary classification is another combinatorial optimization problem that is perhaps as famous and as well-studied as the assignment problem. It focuses on how to best design a classifier from training examples, an important question in the field of machine learning. In binary classification we are given a training set of M points $\big\{\big(x(1),y(1)\big),\big(x(2),y(2)\big),\dots,\big(x(M),y(M)\big)\big\}$, with data $x(i) \in \mathbb{R}^D$, and labels, $y(i) \in \{-1,1\}$, drawn at random according to a probability distribution $P_{X,Y}$. The task is to find a decision function $F\colon \mathbb{R}^D \to \mathbb{R}$ that has small error: $e(F) = E_{X,Y}\big(sign\big(F(x)\big) \neq y\big)$. $P_{X,Y}$ is unknown, but machine learning theory links this error to the empirical risk, or training error [2]. In its simplest form, binary classification involves choosing a function from a function class $\mathcal{F}$ (or hypothesis space) that minimizes the number of mistakes:

$$\hat{F} \in \ argmin_{F\in\mathcal{F}} \sum_{i=1}^{M} sign\Big(F\big(x(i)\big)\Big) \neq y(i) \tag{3}$$

The motivation, definition and solution methods for the two optimization problems defined by equations 2 and 3 differ in many ways. For example, a solution to Equation 2 can be found in polynomial time by the Hungarian algorithm, but the solution to Equation 3 is known to be NP hard even for relatively simple function classes such as linear classifiers.

In this paper we show that using a convex surrogate for Equation 3 (based on the hinge loss) in combination with a particular choice of function class (based on Generalized Stack Filters) leads to a solution to Equation 3 which forms a primal-dual pair with Equation 2. In the next few sections we outline the prior work required to formalize this observation. In section 4 we discuss the relationship in more detail and illustrate some of the connections with synthetic experiments. We conclude in section 6 with a summary of future directions.

2. LINEAR ASSIGNMENT PROBLEM

The problem of finding the minimal cost permutation in Equation 2 can be formulated (and solved) as a $\{0,1\}$-integer linear program. We associate a binary variable $a(i,j) \in \{0,1\}$ with each potential assignment of job i to worker j, and $a(i,j) = 1$ indicates worker i is assigned to task j (i.e. $\varphi(i) = j$).

$$\begin{aligned} &minimize \quad \sum_{i=1}^{N}\sum_{j=1}^{N} c(i,j)a(i,j) \\ subject\ to: \quad &\sum_{i=1}^{N} a(i,j) = 1 \ \ and \ \ \sum_{j=1}^{N} a(i,j) = 1 \ \ \forall i,j \in \{1 \dots N\} \\ &and \quad a(i,j) \in \{0,1\} \quad \forall i,j \end{aligned} \tag{4}$$

One of the reasons why this optimization problem has been so widely studied is because solutions to the linear program relaxation, where $a(i,j) \in \mathbb{R},\ a(i,j) \geq 0$, can be used to find optimal solutions for the integer variables. A large number of solution methods have been developed for Equation 4. Table 4.2 on page 128 of [1] categorizes some of these methods as combinatorial, Primal-Dual (such as the Hungarian Method), Primal, Dual, shortest path, Primal simplex, Auction, Dual simplex, Cost scaling, Pseudoflow and Decomposition.

2.1 Dual Solutions to the Assignment Problem

The duality theorems define the dual optimization problem to Equation 4 as:

$$\begin{aligned} &maximize \quad \sum_{i=1}^{N} u(i) + \sum_{j=1}^{N} v(j) \\ subject\ to: \quad &u(i) + v(j) \leq c(i,j) \end{aligned} \tag{5}$$

In this case we associate a real valued variable with each item defined in the two sets (jobs u and workers v). We denote the variables in Equations 4 and 5 at optimality as a^* and u^*, v^* respectively. These solutions are linked by the following informal definition:

$$a^*(i,j) = 1 \quad \rightarrow \quad u^*(i) + v^*(j) = c(i,j) \tag{6}$$

That is, if in the optimal permutation there is an assignment between job i and worker j, then the dual constraint involving $u(i)$ and $v(j)$ will be tight.

3. ORDERED HYPOTHESIS MACHINES

In previous work we developed novel solution methods for binary classification by investigating Stack Filters as the function class. Stack Filters are nonlinear digital filters that include the Median, Order Statistics, and Weighted Order Statistics as sub-classes[3]. The investigation led us to a new type of classifier, which we call Ordered Hypothesis Machines (OHM) [4], and we summarize some of the key components and results here.

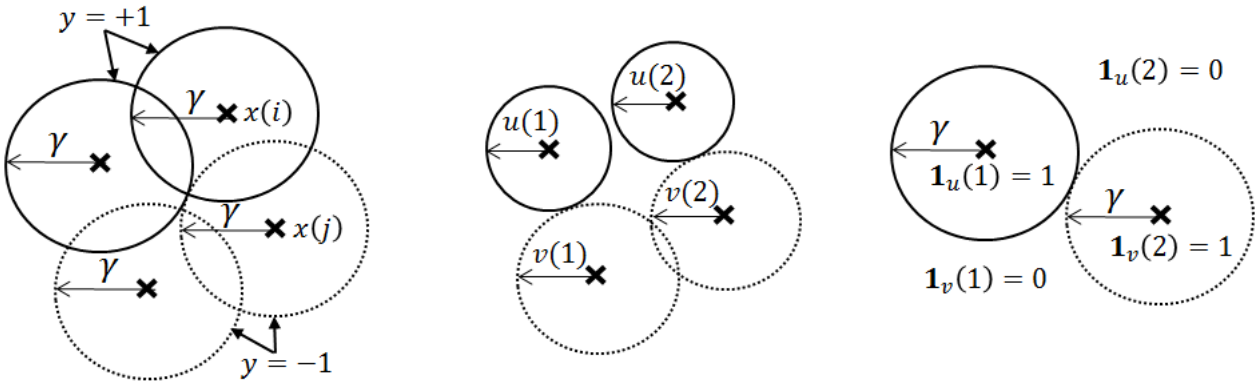

Fig. 1. Geometric interpretation of the OHM classifier design problem. Left) Partitions of radius γ are centered on training examples and compete to cover the examples. Middle) Hypothetical solution to Eq. 9 and partitions associated with different class labels no longer overlap. Right) Hypothetical solution to Stack Filter Classifier design problem which is discussed in Section 4.3.

The first step in minimizing Equation 3 is to replace the zero-one loss function which counts the number of mistakes with a convex loss function:

$$\hat{F} \in \ argmin_{F \in \mathcal{F}} \frac{1}{M} \sum_{i=1}^{M} L\big(F\big(x(i)\big), y(i)\big) \tag{7}$$

where $L(\mathbb{R} \times \{-1,1\}) \rightarrow \mathbb{R}$. OHM classifiers use a large margin hinge loss similar to that used in Support Vector Machines:

$$L(F(x), y) = max\big(0, \gamma - yF(x)\big) \tag{8}$$

where γ is a free parameter (called the margin) used to balance empirical error minimization with function class complexity (analogous to C in Support Vector Machines). OHM classifiers minimize this loss function for a particular choice of function class $\mathcal{F}$ which can be interpreted as a Generalized Stack Filter [5]. For a detailed description of this function class, and how it relates to Stack Filters, and Generalized Stack Filters we refer readers to [4]. Here we simply state that for this function class the optimization problem in Equation 7 can be expressed as the following optimization problem:

$$\begin{aligned} maximize \ & \sum_{i=1}^{PC} u(i) + \sum_{j=1}^{NC} v(j) - \lambda \sum_{i=1}^{PC} \big(u(i)\big)^2 - \lambda \sum_{j=1}^{NC} \big(v(j)\big)^2 \\ subject\ to: \quad & u(i) + v(j) \leq 4\gamma - \Delta_\gamma(i,j) \\ and \quad & u(i), v(j) \geq 0 \quad \forall i \in \{1 \dots PC\}, \forall j \in \{1 \dots NC\} \end{aligned} \tag{9}$$

Where $u(i)$ are variables associated with samples $x(i)$ with positive class labels ($y(i) = 1$), and $v(j)$ are variables associated with samples $x(j)$ with negative class labels ($y(j) = -1$). PC and NC are the number of positive and negative examples in the training set respectively and $PC + NC = M$. In addition,

$$\Delta_\gamma(i,j) = \max\big(0, 2\gamma - d(i,j)\big) \tag{10}$$

and $d(i,j) = \|x(i) - x(j)\|$ is a distance function (such as the Euclidean distance). A geometric interpretation of this optimization problem is illustrated in Figure 1. Variables in the optimization problem correspond to the radius of partitions, centered on training examples. Maximization corresponds to maximizing each radii, up to a maximum value specified by γ, and subject to the constraint that partitions from different classes do not overlap.

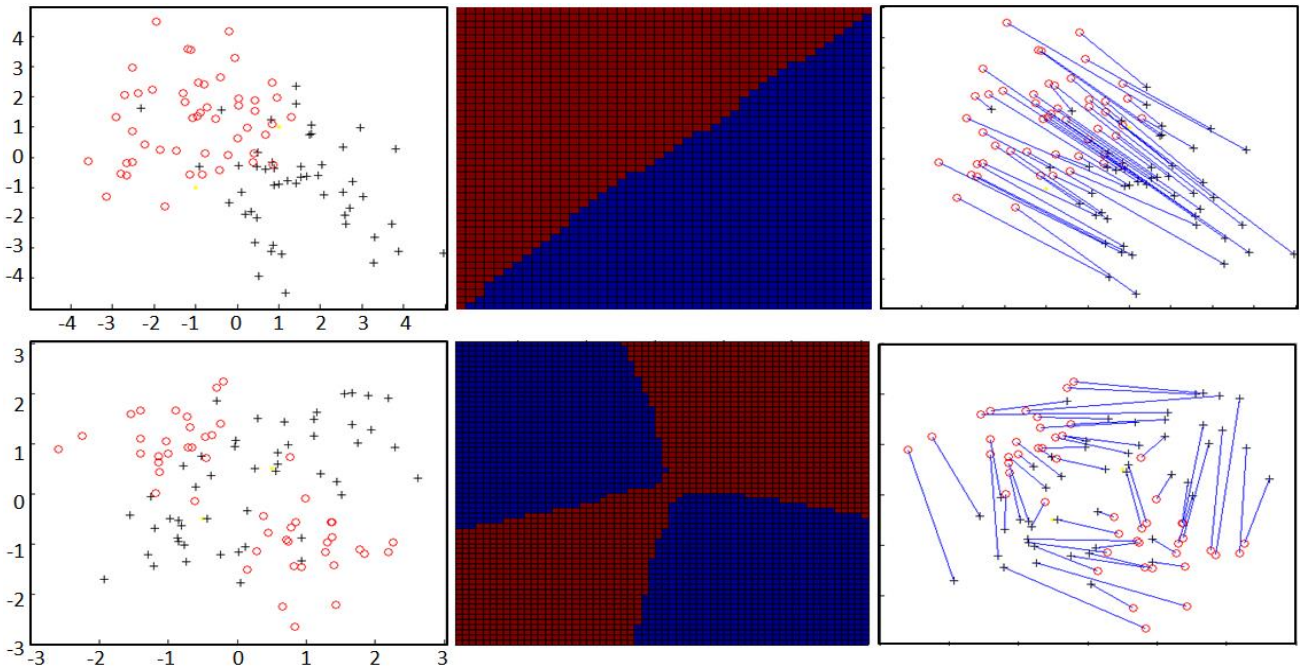

Fig. 2. Left) Synthetic training sets of 100 samples for 2-dimensional linear (top) and xor (bottom) learning tasks. Middle) Decision surfaces found by training with Eq. 9 and then applying Eq. 11. Right) The pairs found with tight constraints that correspond to a solution of the assignment problem in Equation 4.

Once an optimal solution has been found, application of the OHM classifier to test data has a simple form, which is very similar to 1-Nearest Neighbor classification:

$$\hat{F}(x) = \begin{cases} +1 & if \quad min_{i \in PC}\{d(x(i), x) - u(i)\} < min_{j \in NC}\{d(x(j), x) - v(j)\} \\ -1 & otherwise \end{cases} \tag{11}$$

In fact, the only difference between Equation 11 and standard 1-Nearest Neighbor classification is the training sample specific offsets $u(i)$ and $v(j)$ that were found during OHM training. Further discussion and comparison of OHM classifiers to Nearest Neighbor classifiers can be found in [6].

4. RELATIONSHIPS BETWEEN CLASSIFICATION AND MATCHING PROBLEMS

If we ignore the quadratic terms in Equation 9, Equations 5 and 9 are in fact very similar. For balanced classes ($PC = NC = N$), and for large values of the margin parameter ($\gamma \gg max_{i,j} d(i,j)$), $d(i,j) \propto (4\gamma - \Delta_\gamma(i,j))$ the right hand side of the constraints in Equations 5 and 9 are effectively equivalent ($c(i,j) \equiv d(i,j)$). To illustrate this point we perform a number of synthetic experiments. On the left in Figure 2 are 2-dimensional datasets, each with $PC = 50$ and $NC = 50$ ($M = 100$). In the first dataset (top row) the samples are randomly drawn from overlapping Gaussians with means on opposite corners of the unit square and diagonal covariance $4\mathbf{I}$. In the second dataset (bottom row) samples are draw from 4 Gaussians at all corners of the unit square (xor configuration) and diagonal covariance $2\mathbf{I}$. In the middle plots, we show the decision surfaces obtained by optimizing Equation 9 with $\gamma = 10$ and then applying the decision rule in Equation 11.

Note, there are a total of 50^2constraints in the linear program, which we solve with the cvx, a package for specifying and solving convex programs [7] [8]. On the right in Figure 2 we show the assignments that are defined by the dual. These pairs were identified by testing each of the 50^2 constraints and keeping the ones that satisfy $\left|u(i) + v(j) - 4\gamma - \Delta_\gamma(i,j)\right| \leq 1 \times 10^{-8}$. In both datasets there were exactly 50 constraints that satisfied this test, and as seen in the figures, each training sample in class 1 is assigned to exactly one training sample in class -1, indicating that the one-to-one constraints of the primal problem in Equation 4 have indeed been satisfied.

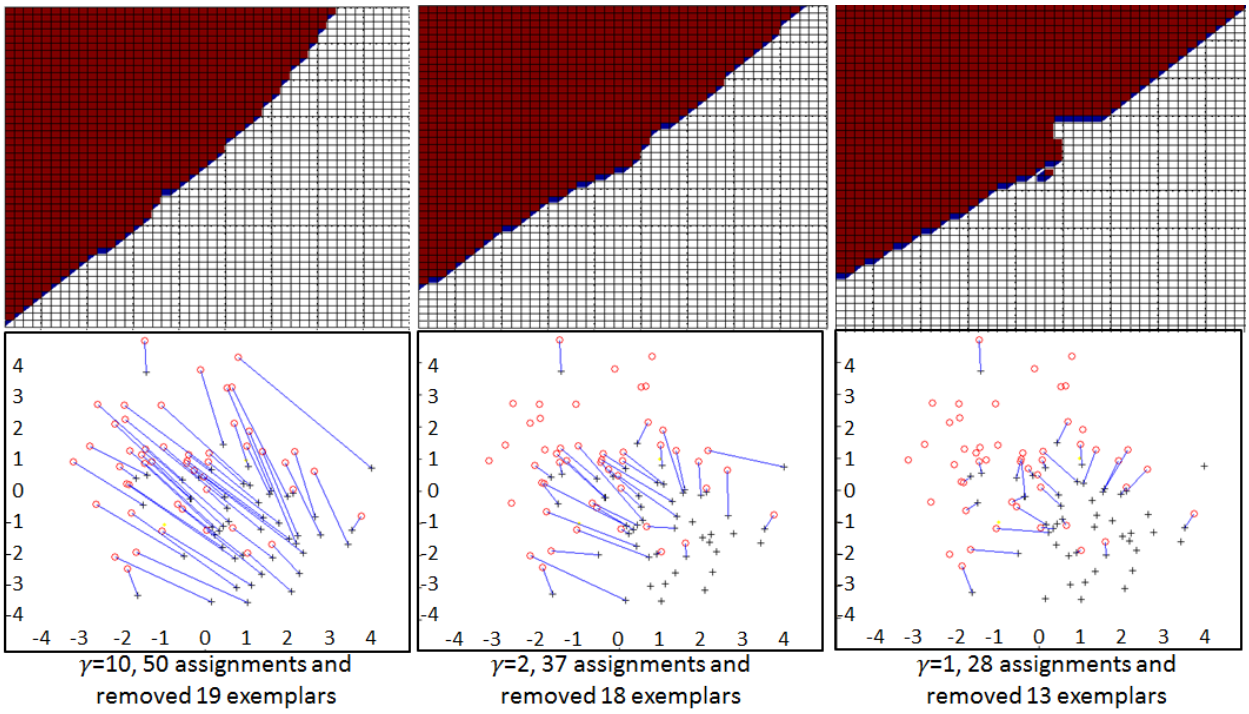

Fig. 3. Top) Decision surfaces found by OHM classifier at various values of margin and Bottom) Assignments associated with these classifiers as the amount of regularization is decreased from left to right.

4.1 Choosing Between Optimal Solutions

One of the difference between Equations 5 and 9 are the quadratic terms. In prior work we observed that without these terms there were many solutions to Equation 9, and that the performance of OHM varied greatly amongst these solutions. The connection to the Assignment problem sheds new light on this observation. Specifically the fact that linear programming relaxations of integer variables in Equation 4 are exact explains why there are many optimal solutions to Equation 9. For the assignment problem, solution methods often *round* real-valued results from the linear program to obtain optimal integer solutions. However, in OHM classification, this *rounding* affects the decision surface and performance of the OHM classifier. In previous work we chose between solutions by adding the quadratic terms to the objective functions at very small cost, e.g. $\lambda = 1x10^{-10}$. These terms select the solution where $u(i)$ and $v(j)$ have similar magnitude which effectively splits any slack in the constraints equally between the classes. This additional term was critical for OHM classifiers to perform well in experiments. However, it also meant that the optimization problem was transformed from a linear program into a quadratic program.

4.2 Regularization

One of the most important properties of the OHM approach to classification is the free parameter γ and the ability to control class complexity in 1-Nearest Neighbor classifiers. One of the effects of this regularization is that the number of exemplars (training examples) that must be memorized is reduced as the regularization is increased (this is analogous to the reduction to support vectors in Support Vector Machines). In its standard form, the Assignment problem does not include any notion of regularization. Although in the previous section we saw how the Assignment problem is equivalent to OHM training when it is most regularized (γ is very large). In the more common case, γ is chosen through cross-validation and is typically much smaller, and we can think of γ as modifying the Assignment problem costs. Specifically, the costs associated with pairs of examples with distances $d(i,j) > 2\gamma$ are replaced by $c(i,j) = Inf$ which essentially removes their influence on the remaining assignments. We note that this can effectively reduce the number of items N in the Assignment problem (and therefore the number of assignments) and can also lead to cases where the number of jobs is different to the number of workers. In Figure 3 we show assignments found as the value of γ

was reduced. We observe decreases in the number of assignments and the amount of training set reduction as the amount of regularization was decreased.

4.3 Stack Filter Classifiers

The function class for OHM classifiers is related to Generalized Stack Filters, but it is also possible to use Weighted Order Statistics [9] and Stack Filters as function classes for binary classification [10]. The optimization problem for Stack Filter Classifiers (SFC) is similar to Equation 9, but the variables associated with training samples and the constants on the right hand side of the constraints are binary $\{0,1\}$. As shown on the right in Figure 1, SFC training can also be interpreted geometrically. Instead of maximizing real-valued variables that represent the radius of each partition (as we did in OHM training), we maximize the sum of binary indicator variables that select between fixed sized partitions. Further details of this optimization problem are available in [10].

The use of binary variables and constants in the SFC design problem means that the constraint matrix for the integer linear program is totally unimodular, and this means that the linear program relaxation can produce optimal integer (binary) solutions. We observe that this problem is the dual of the maximal matching problem on a bipartite graph. When the margin parameter is large the SFC design problem defines a fully connected bipartite graph and a maximal matching can be found easily. But as the margin parameter decreases, there are smaller partitions, and therefore less potential of overlap, which means the bipartite graph becomes increasingly sparse and SFC training is equivalent to the classical combinatorial problem of finding a maximal matching in a bipartite graph [11].

5. EXPERIMENTS WITH THE HUNGARIAN ALGORITHM

In this section we consider how we might solve the OHM classification problem in Equation 9 with the traditional solution methods for the assignment problem, such as the Hungarian [12] or Munkres [13] algorithms. These are both examples of primal-dual solution methods. They start with a feasible solution to the dual (Equation 5) and a partial solution (often a single assignment) to the primal problem (Equation 4). They then iterate by finding augmenting paths that increase the cardinality of the matching (similar to maximal matching algorithms [11]) and then adjusting the dual solution to maintain feasibility.

In our experiments we consider the restricted OHM classification problem described in Section 4. Specifically we consider the balanced problem where the number of training examples in class 1 is equal to the number of training examples in class -1, which leads to a square coefficient matrix where $PC = NC = N$. Also, the value of γ is set to be very large so that the coefficients for the assignment problem are simply the Euclidean distance between points: $c(i,j) = \|x(i) - x(j)\|$. With these choices, OHM classification and the assignment problem are equivalent, however there are still multiple solutions to the optimization problem (described in Section 4.1) which is problematic for classification. In OHM this was addressed by introducing quadratic terms into the dual solution. However this transforms the problem into a quadratic programming problem.

An alternative approach, that might be a better match for the primal problem, is to be more explicit about our class priors and introduce class (or even sample) specific coefficients into the two sums in Equation 5. This weighted version of the assignment problem relaxes the bijection constraints and enables each job to be assigned partially to a number of workers. This generalization is known as the Transportation problem and typically has a much smaller set of optimal solutions compared to the Assignment problem.

5.1 Obtaining Different Dual Solutions

In this paper we investigate an alternative which was inspired by the analysis of dual solutions presented by Kindervater et.al. [14]. They provide an algorithm which can generate $2N$ solutions to the assignment problem. The key observation is that a reduced assignment problem comes into play: remove one agent a and then calculate the total cost $\tilde{z}_{ab}$ of the new problem where task b is removed from the list, doing this for every task $b \in \{1 \dots N\}$. The optimal assignment is given by a permutation σ of $\{1 \dots N\}$, and has cost $z = \sum_{i=1}^{N} c\big(i, \sigma(i)\big)$. The increase in cost over the optimal cost if agent a is assigned to task b is given by Equation 12.

$$h_{ab} = \min\left\{ \sum_{i=1}^{N} c\big(i, \pi(i)\big) \,|\pi \; is \; a \; permutation \; with \; \pi(a) = b \right\} - z \tag{12}$$

Observing that

$$\min\left\{\sum_{i=1}^{N} c\big(i,\pi(i)\big)\,|\pi \text{ is a permutation with } \pi(a)=b\right\} = c(a,b)+\tilde{z}_{ab}$$

we can write $h_{ab}-c(a,b)=\tilde{z}_{ab}-z$. Theorem 2 in [14] states: let a be any given index in $\{1\ldots N\}$, then an optimal dual solution is given by:

$$\big(u_a(i),v_a(i)\big) = \big(c\big(i,\sigma(i)\big)-c\big(a,\sigma(i)\big)+h_{a\sigma(i)},c(a,i)-h_{ai}\big) \text{ for } i\in\{1\ldots N\}.$$

With our notation this can be rewritten as:

$$\big(u_a(i),v_a(i)\big) = \big(c\big(i,\sigma(i)\big)+\tilde{z}_{a\sigma(i)}-z,z-\tilde{z}_{ai}\big) = \big(\tilde{z}_{a\sigma(i)}-\tilde{z}_{i\sigma(i)},z-\tilde{z}_{ai}\big). \tag{13}$$

This specifies N different solutions for the dual problem, one for each choice of row a. The solutions differ in that $u_a(a)=0$ for each choice of a. An additional N solutions can be found by applying a similar procedure to the columns, in which case $v_b(b)=0$ for each choice of b.

5.2 Averaging Dual Solutions

In terms of OHM classification, each of the solutions found by the method described in Section 5.1 is biased towards one class depending on whether the method is applied to the rows, or to the columns. One way to obtain a more symmetric solution is to average the value of the dual variables across the $2N$ solutions, which is described by Equation 14.

$$\big(u^*(i),v^*(i)\big) = \left(\frac{1}{2N}\left(\sum_{a=1}^{N}u_a(i)+\sum_{b=1}^{N}u_b(i)\right),\frac{1}{2N}\left(\sum_{a=1}^{N}v_a(i)+\sum_{b=1}^{N}v_b(i)\right)\right) \tag{14}$$

In experiments, the values of the dual variables found by Equation 14 were found to be identical to the values found by solving the dual problem with the quadratic terms described by Equation 9 (to a constant offset). The identical decision surfaces associated with these classifiers are shown on the left in Figure 4 for the linear (top) and xor (bottom) problems used in Section 4. We compare these to the biased solutions which average solutions from just the rows (Equation 15) or just the columns. Decision surfaces associated with these solutions are illustrated in the middle and right of Figure 4.

$$\big(u^*(i),v^*(i)\big) = \left(\frac{1}{N}\sum_{a=1}^{N}u_a(i),\frac{1}{N}\sum_{a=1}^{N}v_a(i)\right) \tag{15}$$

6. SUMMARY

This paper has shown that for certain function classes binary classification and the Assignment problem are intimately related, and under certain conditions the two problems form a primal-dual pair. This observation means OHM training can potentially benefit from the wide range of solution methods and problem generalizations that have been developed for assignment problems. We gave an example of this by adapting a procedure that uses the Hungarian method to find a set of degenerate solutions to the dual problem. In practice, the efficiency of this approach is not competitive with direct solution of the dual problem with a standard solver. However we believe further work, particularly in the area of incremental (or online) solvers to the assignment problem, may prove more competitive [15].

The observations made in this paper may also benefit applications and methods beyond OHM. The SFC integer linear program described in Section 4.3 is a fairly straightforward adaptation of the original Stack Filter integer linear program, which was used to design filters to minimize Mean Absolute Error (MAE) [3]. In the original work, the linear relaxation of the MAE design problem was shown to be tight (due to the fact that the constraint matrix is totally unimodular) and this contributed to much activity in Stack Filter design and related non-linear filter classes over the years. In the MAE problem, variables do not fall naturally into two sets as they do in binary classification and therefore the connection to maximal matching on bipartite graphs is not clear. However, an interesting topic for future work is to investigate connections between MAE optimization and matching problems on more general graphs. Matching on general graphs has also inspired much historical and recent work [16] [17] and there may be opportunities to improve solution methods for non-linear signal processing.

REFERENCES

1. Burkard, R., M. Dell'Amico, and S. Martello, *Assignment Problems*. 2012: Society for Industrial and Applied Mathematics.
2. Vapnik, V.N., *Statistical learning theory*, ed. Wiley. 1998, New York.

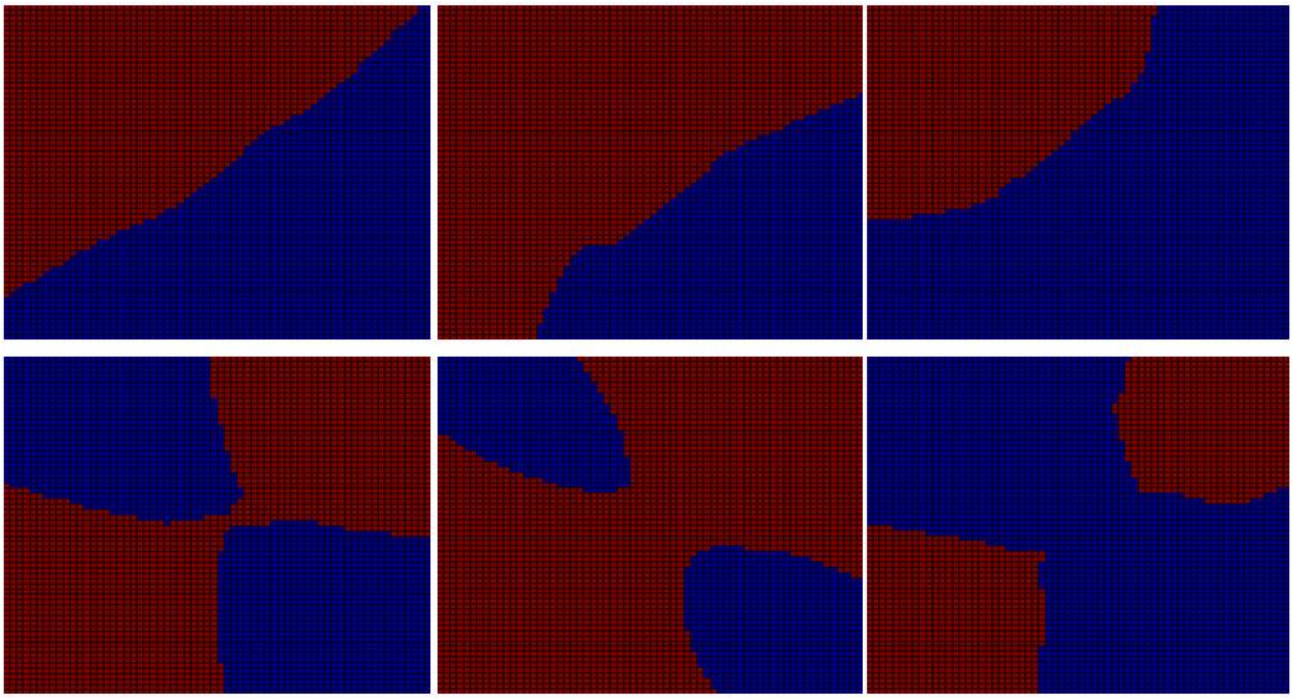

Fig. 4. Left) Decision surfaces found by solutions to Equation 14. Middle) Solution to equation 15 which biases solutions towards class -1 when row variables are set to 0, and Right) when column variables are set to 0.

3. Wendt, P., E.J. Coyle, and N.C. Gallagher, Jr., *Stack filters.* Acoustics, Speech and Signal Processing, IEEE Transactions on, 1986. **34**(4): p. 898-911.
4. Zimmer, G.B., D. Hush, and R. Porter, *Ordered Hypothesis Machines.* Journal of Mathematical Imaging and Vision, 2012. **43**(2): p. 121-134.
5. Lin, J.H. and E.J. Coyle. *Generalized stack filters and minimum mean absolute error estimation*. in *Circuits and Systems, 1988., IEEE International Symposium on*. 1988.
6. Porter, R.B., D. Hush, and G.B. Zimmer. *Error minimizing algorithms for nearest neighbor classifiers.* in *Image Processing: Algorithms and Systems IX*. 2011. San Francisco.
7. Grant, M. and S. Boyd. *CVX: Matlab Software for Disciplined Convex Programming, version 2.1*. 2014 March.
8. Grant, M. and S. Boyd, *Graph implementations for nonsmooth convex programs,* in *Recent Advances in Learning and Control*. 2008, Springer-Verlag Limited. p. 95-110.
9. Porter, R., et al. *Weighted order statistic classifiers with large rank-order margin*. in *20TH International Conference on Machine Learning* 2003. Washington.
10. Porter, R.B., D. Hush, and B.G. Zimmer. *Stack Filter Classifiers*. in *International Symposium on Mathematical Morphology*. 2009. Groningen, The Netherlands.
11. Hopcroft, J.E. and R.M. Karp, *An n^5/2 algorithm for maximum matchings in bipartite graphs.* SIAM Journal on Computing, 1973. **2**(4): p. 225–231.
12. Kuhn, H.W., *The Hungarian method for the assignment problem.* Naval Research Logistics Quarterly, 1955. **2**(1-2): p. 83-97.
13. Munkres, J., *Algorithms for the Assignment and Transportation Problems.* Journal of the Society for Industrial and Applied Mathematics, 1957. **5**(1): p. 32-38.
14. Kindervater, G., et al., *On dual solutions of the linear assignment problem.* European Journal of Operational Research, 1985. **19**(1): p. 76-81.
15. Toroslu, I.H., et al., *Incremental assignment problem.* Inf. Sci., 2007. **177**(6): p. 1523-1529.
16. Edmonds, J., *Maximum matching and a polyhedron with 0,1-vertices.* Journal of Research National Bureau of Standards, 1965. **Section B** (69): p. 125–130.
17. Chertkov, M., A. Gelfand, and J. Shin, *Loop calculus and bootstrap-belief propagation for perfect matchings on arbitrary graphs.* J. Phys.: Conf. Ser. 473 012007, 2013.