Ising Bandits with Side Information

Shaona Ghosh(✉) and Adam Prügel-Bennett

University of Southampton, Southampton SO17 1BJ, UK
ghosh.shaona@gmail.com

Abstract. We develop an online learning algorithm for bandits on a graph with side information where there is an underlying Ising distribution over the vertices at low temperatures. We are motivated from practical settings where the graph state in a social or a computer hosts network (potentially) changes at every trial; intrinsically partitioning the graph thus requiring the learning algorithm to play the bandit from the current partition. Our algorithm essentially functions as a two stage process. In the first stage it uses "*minimum-cut* " as the regularity measure to compute the state of the network by using the side label received and acting as a graph classifier. The classifier internally uses a polynomial time linear programming relaxation technique that incorporates the known information to predict the unknown states. The second stage ensures that the bandits are sampled from the appropriate partition of the graph with the potential for exploring the other part. We achieve this by running the adversarial multi armed bandit for the edges in the current partition while exploring the "cut" edges. We empirically evaluate the strength of our approach through synthetic and real world datasets. We also indicate the potential for a linear time exact algorithm for calculating the max-flow as an alternative to the linear programming relaxation, besides promising bounded mistakes/regret in the number of times the "cut" changes.

1 Introduction

Many domains encounter a problem in collection of annotated training data due to the difficulty and costs in requiring efforts of human annotators, while the abundant unlabelled data come for free. What makes the problem more challenging is the data might often exhibit complex interactions that violate the independent and identically distributed assumption of the data generation process. In such domains, it is imperative that learning techniques can learn from unlabelled data and the rich interactions based structure of the data. Learning from unlabelled and a few labelled data falls under the purview of semi-supervised learning. Coupling it with an encoding of the data interdependencies as a graph, results in an attractive problem of learning on graphs.

Often, interesting applications are tied to such problems with rich underlying structure. For example, consider the system of online advertising; serving advertisements on web pages in an incremental fashion. The web pages can be represented as vertices in the graph with the links as edges. At given time t,

A. Appice et al. (Eds.): ECML PKDD 2015, Part I, LNAI 9284, pp. 448–463, 2015.
DOI: 10.1007/978-3-319-23528-8_28

the system receives a request to serve an advertisement on a randomly selected web-page. Moreover, at the same time, the system receives a side information about the state of the web-page: for simplicity we assume the side information to be a rating of 0 or 1. As a consequence, the advertisement pool change with the change in the state of the graph or the ratings, given the already known states and the current advert should be served from the appropriate pool. Once the chosen advertisement is served, the feedback is received and incorporated in serving the next request.

At a deeper level of understanding, the side information can be interpreted as the label of the vertex. There are few available labels at the start; the rest are only incrementally revealed. When a vertex is queried (request for an ad placement made), an action needs to be picked (an advertisement needs to be served) from a set of actions. The algorithm should be able to internally predict what the state of the queried vertex is (how the state of the graph changes) and then select the appropriate action from the action pool that (potentially)changes with the predicted label of the queried vertex.

In this paper, we attempt to tackle this problem by exploiting the knowledge of the non-independence graphical structure of the data in an online setting. We do so by associating a complexity with the labelling. We call this complexity "cut" or "energy" of the labelling on a Markov random field with discrete states (Ising model). The goal of our graph labelling procedure is to minimize the energy while being consistent with the information seen so far when predicting the intrinsic state of the queried vertex at every round. This prediction directs the overall goal towards minimizing the regret of our sequential action selection (bandit) algorithm within the online graph labelling that occurs over the entire sequence.

Related Work. Broadly speaking, there are two central themes that run through our work unified under the common framework of online learning, namely, action selection using bandit feedback and semi-supervised graph labelling. The closest related work that addresses the intersection of these two themes is the work by Claudio et al. [10]. They use bandit feedback to address a multi-class ranking problem. The algorithm outputs a partially ordered subset of classes and receives only bandit feedback (partial information) among the selected classes it observes without any supervised ranking feedback. In contrast, we play the bandit game of sequential action selection, using side information as the class label of the current context. Our feedback for the action selected is still partial (only loss for the selected action is observed). Further, our bandits have a structure associated with the Ising model distribution over the vertices at low temperature. The work of Amin et al.[2], addresses the graphical models for bandit problems to demonstrate the rich interactions between the two worlds in the similar lines of what we try to achieve. Bearing a strong resemblance to our work, they address the similar context-action space. However, in their setting, there is a strong coupling between the context-action space; the algorithm needs to fulfil the entire joint assignment before receiving any feedback. In contrast, our concept-action space is decoupled, labels are revealed gradually determining the current active concept for the learner to choose the action and receive the feedback instantaneously. In their problem formulation under the Ising graph setting, the algorithm tries to pick the action (the label of the concept) that is NP hard. In contrast, we focus on the low temperature setting, where our actions lie on the edges, and are not the labels of the vertices. The computation of the marginal at the vertices is guided by the labels seen so far and the minimal cut. We approximate the labelling of the entire graph rather than predicting the spin configuration of a single vertex using the "cut" as the regularizer that dominates the action selection. The contextual bandits work on online clustering of bandits [9], deals with finding groups or clusters of bandits in the graphs. They have a stochastic assumption of a linear function for reward generation. Similarity is revealed by the parameter vector that is inferred over time. In contrast, we use the similarity over edges to determine the "cut" which in-turns guides the action selection process in adversarial settings. There work extends to running a contextual bandit for every node, whereas ours is a single bandit algorithm, where the context information is captured in the "cut". The work of Castro et al. [7] of edge bandits is similar in the sense that the bandits lie on the edges. However, instead of direct rewards of action selection, rewards are a difference in the values of the vertices. Further, this is the stochastic setting instead of the adversarial one. In Spectral bandits [18], the actions are the nodes, while there is a smooth Laplacian graph function for the rewards. We discuss later the limitations of Laplacian based methods for graph labelling. Further, they do not consider the Ising model that we study. The seminal work of semi supervised graph labelling prediction can be found in [6], where minimum label-separating cut is used for prediction. Laplacian based methods that results neighbouring nodes connected by an edge to share similar values are widely studied in the semi-supervised and manifold learning problems [5,11,12,19,20]. Typically, this information is captured by the semi-norm induced by the Laplacian of the graph. Essentially, the smoothness of the labelling is ensured by the "cut". The "cut" is the number of edges with disagreeing labels. Then, the norm induced by the Laplacian can be considered as the regularizer. However, there are limitations in these methods with increasing unlabelled data [1,16]. Here, we also use "cut" as the regularization measure over an Ising model distribution of the values over the vertices of the graph at low temperatures. We simultaneously find the partition using the "min-cut" and then sample the actions from the relevant partition.

2 Background and Preliminaries

2.1 Semi-supervised Graph Classifier Complexity

The standard approach in semi supervised learning is to construct the graph from the unlabelled and labelled data such that each datum is denoted as a vertex. Traditionally, the norm induced by the graph Laplacian is used to predict the labelling. Typically, either the norm induced by the Laplacian is directly minimized/interpolated with respect to constraints or is used as a regulariser. Both methods help build classifiers on graphs in order to learn sparse labels in

$\mathbb{R}^n$ by incorporating a measure of complexity also called "cut" or energy. For a graph $\mathcal{G} = (V, E)$, where the set of vertices $V = \{v_1, \ldots, v_n\}$ are connected by edges in E. Let a weight of A_{ij} be associated with every edge $(i, j) \in E$, such that $\mathbf{A}$ is the $n \times n$ symmetric adjacency matrix, then the Laplacian $\mathbf{L}$ of the graph is given by $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{D}$ is the degree matrix with its diagonal values given by $D_{ii} = \sum_j A_{ij}$. We re-state Definition 1 from [14] that relates the quadratic form of the Laplacian with the complexity of the "cut-size" for completeness.

Definition 1 ([14]). *If the labelling of the graph $\mathcal{G}$ is given by $\boldsymbol{u} \in \mathbb{R}^n$, the "cut size" of $\boldsymbol{u}$ is given by*

$$\psi_{\mathcal{G}}(\boldsymbol{u}) = \frac{1}{4}\boldsymbol{u}^T \boldsymbol{L} \boldsymbol{u} = \frac{1}{4} \sum_{(i,j) \in E} A_{ij} (u_i - u_j)^2 \ . \tag{1}$$

When $\boldsymbol{u} \in \{0, 1\}^n$, the "cut" is on the edge (i, j) where $u_i \neq u_j$, then $\psi_{\mathcal{G}}(\boldsymbol{u})$ is the number of "cut" edges.

The smoothness functional of $\boldsymbol{u}^T \boldsymbol{L} \boldsymbol{u}$ is generalized in the work of semi-norm interpolation [13] where the Laplacian $p-$seminorm is defined on $\boldsymbol{u} \in \mathbb{R}^n$ as:

$$||\boldsymbol{u}||_{\mathcal{G},p} \simeq \psi_{\mathcal{G}}(\boldsymbol{u}) = \left(\sum_{(i,j) \in E} A_{ij} \left| u_i - u_j \right|^p \right)^{\frac{1}{p}} \ . \tag{2}$$

When $p = 2$, this is equivalent to the harmonic energy minimization technique in [20]. Alternatively, this technique is also called the Laplacian interpolated regularization [4]. In [14], the online version of the $p = 2$ case is studied in the context of the already available labels. If $\mathcal{G}$ is a partially labelled graph as in our problem, such that $|V| = N$, and the partial labels $l \leq N$, with the labels given by $\boldsymbol{y}_l \in \{1, -1\}^l$ on the l vertices, then the minimum semi-norm interpolation gives the labelling:

$$\boldsymbol{y} = \text{argmin}\{\boldsymbol{u}^T \boldsymbol{L} \boldsymbol{u} : \boldsymbol{u} \in \mathbb{R}^n, u_r = y_r, r = 1, \ldots, l\} \ .$$

The prediction is made by using $\hat{y}_i = \text{sgn}(y_i)$ [13]. The rationale behind minimizing the cut enables the neighbouring vertices to have similarly valued labels. With $p \to 1$, the prediction problem is reduced to predicting using the label consistent minimum cut.

2.2 Ising Model at Low Temperature

As discussed above, the labelling of the whole graph is obtained by optimizing the objective function constrained on the given labels. From label propagation [20], we saw when $p = 2$, the harmonic energy function $E(\mathbf{u})$ minimized in (1) is quadratic in nature. The technique in (1), chooses the label as a function $u : V \to \mathbb{R}$ and a probability distribution on the function u given by a Gaussian field $P(\mathbf{u}) = \frac{\exp^{-\beta E(\mathbf{u})}}{Z}$, where Z is the partition function and β in the inverse temperature or the uncertainty in the model. There are multiple limitations of the quadratic energy minimization technique. This model is not applicable for $p \rightarrow 1$ in the limit. Not only is the computation slow, the mistake bounds obtained are not the best. Further, in our problem, we relax the values of the labels such that $u : V \rightarrow [0, 1]$. With $p \rightarrow 1$, the energy function is equivalent to the the one that finds the minimum cut. Further, when $p \rightarrow 1$ using (2) results in the minimization of a non-strongly convex function per trial that is not differentiable. Also, interesting is that the Laplacian based methods are limited with the abundance of unlabelled data [16]. Hence, we are interested in the Markov random field applicable here with discrete states also known as the Ising model. At low temperatures, the Ising probability distribution over the labellings of a graph $\mathcal{G}$ is defined by:

$$P_T^{\mathcal{G}}(\boldsymbol{u}) \propto \exp\left(-\frac{1}{T}\psi_G(\boldsymbol{u})\right) \ . \tag{3}$$

where T is the temperature, $\mathbf{u}$ is the labelling over the vertices of $\mathcal{G}$ and $\psi_{\mathcal{G}}(\boldsymbol{u})$ is the complexity of the labelling or the "cut-size". The probabilistic Ising model encodes the uncertainty about the labels of the vertices and at low temperatures favours labellings that minimise the number of edges whose vertices have different labels as shown in (2) with $p = 1$. If the vertex labels pairs seen so far is given by Z_t of vertex label pairs $(j_1, y_1), \ldots, (j_t, y_t)$ such that $(j, y) \in V(\mathcal{G}) \times \{0, 1\}$, then the marginal probability of the label of the vertex v being y conditioned on Z_t is given by: $P_T^{\mathcal{G}}(u_v = y|Z_t) = P_T^{\mathcal{G}}(u_v = y|u_{j_1} = y_1, \ldots, u_{j_t} = y_t)$. At low temperatures and in the limit of zero temperature $T \rightarrow 0$, the marginal favours the labelling that is consistent with the labelling seen so far and the minimum cut. Such label conditioning or label consistency in the context of graph labelling has been extensively studied [11,12,15]. In this paper, we are only interested in the low temperature setting of the Ising model as the environment in which the player functions. However, at low temperatures, the minimum cut is still not unique.

2.3 Multi-Armed Bandit Problem (MAB)

As with any sequential prediction game, the MAB is played between the learner and the environment and proceeds in a series of rounds $t = 1, \ldots, n$. At every time instance t, the forecaster chooses an action I_t from the set of actions or arms $a_t \in \mathcal{A}$, where $\mathcal{A}$ is the action set with K actions. When sampling an arm, the learner suffers a loss l_t that the adversary chooses in a randomized way. The forecaster receives the loss for the selected action only in the bandit setting. The objective of the forecaster is to minimize the regret given by the difference between the incurred cumulative loss on the sequence played and the optimal cumulative loss with respect to the best possible action in hindsight. The decision making process depends on the history of actions sampled and losses received up until time $t - 1$. The notion of regret is expressed as expected (average) regret and pseudo regret, where pseudo regret is the weaker notion because of the comparison with the optimal action in expectation. For the adversarial case, it is given by:

$$\overline{R_n} = \mathbb{E}\sum_{t=1}^{n} l_{I_t,t} - \min_{i=1,\ldots,K} \mathbb{E}\sum_{t=1}^{n} l_{i,t} \ . \tag{4}$$

The expectation is with respect to the forecaster's internal randomization and possibly the adversary's randomization. In this work, we consider the adversarial bandit setting with side information (information at queried vertex). Note that unlike in the standard MAB problems where there is no structure defined over the actions, in our setup of the problem, we not only have a structure over the action set but also potentially utilize the associated structural side information that makes the problem more realistic. One more deviation from the standard MAB framework is that at every round, the adversary randomly selects a vertex as the current concept; the value of the concept queried is unknown until after the trial and action selection. Further, our adversary is restricted in that the complexity or "cut-size" of the model of the environment that we have chosen cannot increase across trials. The intuition being, the number of times the learner makes a mistake (predicts the queried state wrong) or does not choose the optimal action, is bounded by the number of times the "cut" changes for the minimum.

2.4 Formulation

We consider an undirected graph $\mathcal{G} = (V(\mathcal{G}), E(\mathcal{G}))$ where the elements of E are called edges that form an unordered pair between the unique elements of V that are called vertices. We assume an unit weight on every edge. The number of vertices in the graph are denoted by N. The vertices of the graph are associated with partially unknown concept values or labels s_i that are gradually revealed, while the bandits lie on the edges in $E(\mathcal{G})$ to form the action set $\mathcal{A}$ with cardinality $|K|$. We assume a κ connected graph, the maximum value of κ such that each vertex has at least κ neighbours. Vertices i and j are neighbours if there is an edge/action connecting them. Note, the number of rounds $n \leq |K|$. In our case, n is equal to number of vertices queried by the environment with unknown labels. A vertex is randomly selected by the environment at every round t, in our case, the queried vertex is given by x_i where $i \in \mathbb{N}_N$. In our example, the queried vertex could represent the request to place an advert on the product website the user currently visits . More specifically, the connections in our graph, not only capture the explicit connections between vertices given by locality, but our bandits or edges also capture the implicit connections between the values of vertices that are possibly differently labelled. In our case, the labels are relaxed such that the label for the i-th vertex is denoted by $s_i = \{-1, 1\}$.

At the start we are given the labels of a small subset of observed vertices, $s^{\ o} \in \mathcal{S}^o \subset V(\mathcal{G})$. The labels of the unlabelled vertices $s^u \in \mathcal{S}^u \subset V(\mathcal{G}) \setminus \mathcal{L}$, with $S = S^o \cup S^u$ is revealed one at a time sequentially as at the end of each round as side information. We assume that there are at least two vertices labelled at the start, one in each category. The learning algorithm plays the online bandit game where the adversary at each trial reveals the loss of the selected action and the label of a randomly selected vertex. The goal of the learner is to be able to predict the label of the randomly selected vertex and then sample the appropriate action given the prediction.

3 Maximum Flow Computation

Given a partially labelled graph, the Ising model associates a probability with every labelling that is a consistent completion of the partial labellings. Now, if "cut" of a labelling defines the "energy" of the labelling, then the *low – temperature* Ising is a simplified landscape made up of all such minimum cut (energy) labellings. In a way, the Ising model induces an "energy landscape" over labellings via the "cut." For a n-vertex graph, the energy levels sit inside the n-dimensional hypercube. One can minimize the energy while being consistent with the observations seen so far to achieve the desired goal.

As a first step in the learning process, the learner has to detect the underlying hidden partition in the graph, given the available labels with respect to the currently queried vertex. It can do so by using efficient graph partitioning methods. However, given the partial labelling, the partition detected should respect or be consistent with the labels seen so far. One way to address this is using optimization methods that satisfies the label consistency through constraints. Alternatively, there are very efficient linear time exact methods that can solve this in practise. One such method is "Ford- Fulkerson"[8] algorithm. If one can characterize the labelled vertices in such a way to designate a single source, single sink network, running "Ford Fulkerson"[8] in an online fashion for every round using the side information can be used to efficiently detect the partition. Here, we choose to use a simplified linear programming relaxation to the classic Linear Programming (LP) maximal flow problem (5). Although, the LP formulation we use, can be solved in polynomial time, there is nothing restricting us in using the linear time modified "Ford-Fulkerson" algorithm to achieve the same goal. The objective here is to enable the learner for better predictions and hence lower its regret quicker by detecting the partition early, rather than to illustrate the computational efficiency of the method.

It is known by Menger's theorem of linear programming duality, that maximum flow and the minimum cut are related given a source and a target vertex. Let us introduce the maximum flow or label consistent minimum cut in the graph using the following notation $c^* = \min\{\mathbf{S} \in \{-1, 1\}^N : \psi_{\mathcal{G}}(\mathbf{S}|\mathcal{H})\}$ consistent with the trial sequence $\mathcal{H}$ seen so far.

$$E(\mathbf{S}) = \underset{\mathbf{S}\in\{-1,1\}^N}{\arg\min} \sum_{(i,j)\in E(\mathcal{G})} |s_i - s_j| \leq c^* \ . \tag{5}$$

In general. linear programming relaxations are much easier to analyse. Interested readers are referred to the article [17], where LP relaxations are discussed. We use a linear programming relaxation of the above objective as shown in Fig.1

that has auxiliary variables introduced such that there is one variable for every vertex v and one variable for every edge f_{ij}. Since, we have an undirected graph, we assume a directed edge in each direction, for every undirected edge. Hence we have two flow variables per edge in the graph. Essentially, the free variables in the optimization are the unlabelled vertices s^u_i, s^u_j and the flows across every direction f_{ij}. The total flow across all the edges will be our maximum flow for this low temperature Ising model. The formulation in Fig.1 below is what the learner follows to find the minimum cut $\psi_{\mathcal{G}}$. The output from the computation is a directed graph with the value of flow at every edge and the labelling of the vertices consistent with the labels seen so far; $w_{(i,j)}$ is the cost variable of the LP. The sum of the flows is the maximum flow in the Ising model at low temperatures. We fix one of the labelled vertices as a source, and one as target, each with different labels. We assume a unit capacity on every edge. The constraints in Fig. 1 ensure the capacity constraint $f_{(ij)}$ and conservation constraint $s_i - s_j$are adhered to i.e. the flow in any vertex v other than the source and target, is equal to flow out from v. The largest amount of flow that can pass through any edge is at most 1, as we have unit capacity on every edge. We know that the cost of the maximum flow is equal to the capacity of the minimum cut. The minimum cut obtained as a solution to the optimization problem is an integer.

3.1 Playing Ising Bandits

Figure 2 describes the main algorithm for Ising bandits. It is important to note that `ComputeMaxFlow` can only guide the player towards the active partition with respect to the current context (queried vertex) by detecting the partition early on. $\mathcal{P}$ is a subgraph of $\mathcal{G}$, $\mathcal{P} \subseteq \mathcal{G}$ iff $V(\mathcal{P}) \subseteq V(\mathcal{G})$ and $E(\mathcal{P}) = \{(i,j) : i, j \in V(\mathcal{P}), (i,j) \in E(\mathcal{G})\}$. `SelectPartition` samples the Ising bandits from the best partition with respect to the active concept if the minimum cut changes from previous round. $E(\mathcal{R}), E(\mathcal{J})$ are the partitions of the action set at trial t. Since $\mathbf{S}'$ provides the labelling, it is easy to see which bandits fall in which partition with respect to x_t . The probability distribution r_t over $E(\mathcal{R})$, and j_t over $E(\mathcal{J})$ sum to p_t. Note that if the cut remains the same, player keeps playing the same partition until the cut changes. This has an important implication. Since we assume that the adversary cannot increase the cut at any trial, the cut can only decrease or stay the same. For the rounds it stays the same, the regret that the player suffers is well bounded by the number of times the cut changes. In the best case, the algorithm behaves as a typical Multi-armed bandit (MAB) and in the worst case when the partition changes at every round, the algorithm plays the modified `Ising Bandits`. The algorithm parameter η is the standard MAB value $\eta = \sqrt{\frac{\log |K|}{3n}}$.

4 Experiments

In our experiments, we compare three competitor algorithms with our algorithm `IsingBandits`. The three are `LabProp` [19,20], `Exp3` [3] and `Exp4`[3]. `Exp3` and

ComputeMaxFlow(*target vertex:* $s_\sqcup$; *source vertex:* $s_\sqcap$; *trial sequence:* $\mathcal{H} = (x_k, s_k)_{k=1}^{t}$; *graph:* $\mathcal{G}$)

$$\textit{minimize} \sum_{(i,j) \in E(\mathcal{G})} w(i,j) f(i,j)$$

subject to:

$$f_{(i,j)} \geq 0 \tag{6}$$

$$s_i - s_j \leq f_{(ij)} \tag{7}$$

$$s_i \geq -1 \tag{8}$$

$$s_i \leq 1$$

Return: *min-cut:* c^*; *flows:* f; *consistent partition:* $\mathbf{S}^{'}$

Fig. 1. Computing the Max-flow

```
Parameters: Graph: $\mathcal{G}$; $\eta \in \mathbb{R}^+$
Input: Trial Sequence: $\mathcal{H} = \langle (x_1, -1), (x_2, 1), (x_3, s_3), \ldots, (x_t, s_t) \rangle$
Initialization: $p_1$ is the initial distribution over $\mathcal{A}$ such that, $p_1 = (\frac{1}{|K|}, \frac{1}{|K|}, \ldots, \frac{1}{|K|})$,
Initial cut-size $c = \infty$; active partition distribution $r_1 = p_1$
for $t = 1, \ldots, n$ do
    Receive: $x_t \in \mathbb{N}_N$
    $(c^*, f, \mathbf{S}^{'}) =$ ComputeMaxFlow$(s_\sqcup, s_\sqcap, \mathcal{H}, \mathcal{G})$
    if $(c \neq c^*)$ then                % if cut has changed
        $(E(\mathcal{R}), E(\mathcal{J}), r_t, j_t) =$ SelectPartition$(x_t, p_t, \mathtt{S}^{'}, \mathcal{A})$
    Assign: $q_t$ be the distribution over Ising bandits w.r.t $x_t$, such that,
    $\sum_{i=1}^{|E(\mathcal{R})|} q_{i,t} = r_t$. For any $t$, $p_t = r_t \cup j_t$
    Play: $I_t$ from $q_t$
    Receive: Loss $z_t$; side information $s_t$
    Compute: Estimated loss $\tilde{z}_{i,t} = \frac{z_{i,t}}{q_{i,t}} \mathbb{1}_{I_t = i}$
    Cumulative estimated loss: $\tilde{Z}_{i,t} = \tilde{Z}_{i,t} + \tilde{z_{i,t}}$
    Update: $q_{i,t+1} = \frac{q_{i,t} \exp(\eta \tilde{Z}_{i,t})}{\sum_{j=1}^{|E(\mathcal{R})|} \exp(\eta \tilde{Z}_{j,t})}$
end
```

Fig. 2. Ising Bandits Algorithm

Exp4 are from the same family of algorithms for bandits in the adversarial setting. Exp4 is the contextual bandit setting, the close competitor to Ising from the contextual perspective. The experts or contexts in Exp4 for our problem setting are a number of possible labellings. Note that the number of experts selected for prediction have a bearing on the performance of the algorithm. In our experiments, we fixed the number of experts to 10. In reality, even at low temperatures for the model we consider, the set of all possible labellings is exponential in size. LabProp [19,20] is the implementation where the state-of-the-art graph Laplacian based labelling procedure is used to optimize the labelling consistent with the labels seen so far. For all of the above algorithms, we use our own implementation in MATLAB. Since online experiments are extremely time consuming while processing one data point at a time, we have averaged each set of experiments over five trials but for ISOLET, where we average over ten trials. The datasets that we use are the standardized UCI datasets namely the USPS and the ISOLET datasets. All datasets are nearly balanced in our experiments to demonstrate the fairness of the class distribution and for avoiding any majority vote cases where the class with the majority vote wins.

4.1 Dataset Description

The summary of datasets used is captured in Table 1. The USPS handwritten digits is an optical character recognition dataset comprising 16x16 grayscale images of "0" through "9" obtained from scanning handwritten digits. The preprocessed dataset has each image with 256 real valued features without missing values scaled to [-1,1]. We randomly sample the examples for the graph from the 7291 original training points. Each vertex in the graph thus sampled is a digit. We perform several binary graph generation of sampling one digit vs. the other digit to form our underlying graph with edges or connections between the two digits forming our action set.

We use a noisy perceptual dataset for spoken letter recognition called ISOLET consisting of 7797 instances with 617 real valued features. A total of 150 subjects spoke each letter of the English alphabets twice resulting in 52 training examples from each speaker. The total of 150 speakers are split into 30 speakers each into files named as Isolet 1 through to Isolet 5. For the purpose of our experiments here, we build the graph from Isolet 1 comprising 1560 examples from 30 speaker with each letter being spoken twice. Again, we are only interested in binary classified graphs here where we sample the first 13 spoken letters and the last 13 spoken letters as two separate underlying concepts in our graph, the connections between which form our action set.

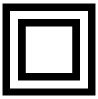


Fig. 3. Squares image.

Table 1. Datasets used in this paper.

Data set	#Instances	#Features	#Classes
USPS	7291	256	10
Isolet	7797	617	26

4.2 Synthetic Dataset

Our synthetic data uses a 2D grid like topology. Figure 3 shows the image used to construct the graph in our experiments. Our interest in using the image for our simulation experiment stems from the natural occurring graph structure in such 2D grids. The image style of `Squares` is chosen based on our interest in smooth and wide regions of similar labels interspersed with dissimilar labelled boundary regions. We use a square image that is constructed using a set of pixels, each with an intensity of 0 or 1. The 0 and 1 intensities are balanced across the pixels i.e. there are equal number of pixels with 0 and 1 intensities. Each pixel in the image corresponds to a vertex in the graph and the intensities correspond to the label or class of the vertex. Here, our graph has 3600 vertices. The neighbourhood system in the graph comprises of edges connecting pairs of neighbouring similar pixels. The connectivity is typically guided by if the pixels are of comparable intensities, if the pixels are structurally close to each other or both. Here, we are only interested in the physical pixel locations that are used to determine connectivity i.e. pixels closer to each other on the grid are connected. The connections eventually form our bandits action set. In this paper, we are only interested in undirected and unweighted graphs. Our grid graph thus generated have a weight of 1 on every edge and there is an edge in either direction. Further, we investigate the type of neighbourhood system, called torus. In the torus grid, each pixel has four neighbours; achieved by connecting the top with the bottom edge pixels and the left with the right edge pixels. Our graph is the same across trials. We randomly sample the available labelled vertices from the graph such that there are equal number of labels from each concept class.

4.3 Graph Generation from Datasets

We design our experiments to test the action selection algorithm under a number of different criteria of graph creation: balanced labels, varying degree of connectedness, varying sizes of initial labels and noise. The parameters that are varied across the experiments are graph size indicated by N, labels available as L, connectivity K, noise levels nse.

In the set of experiments with `ISOLET`, we chose to build the graph from the first 30 speakers in Isolet1 that forms a graph of 1560 vertices of 52 spoken letters (each letter spoken twice) by 30 speakers. The concept classes that are sampled are the first 13 letters of English alphabets as one concept vs. the next 13 letters as the other concept. We build a 3 nearest neighbour graph from the

Euclidean distance matrix constructed using the pairwise distances between the examples (spoken letters). In order to ensure that the graph is connected for such low connectivity, we sample a MST for each graph and always maintain the MST edges in the graph. The MST uses the Euclidean distances as weights. The same underlying graph is used across trials. The edges or connections form the bandits. The available side information is sampled randomly such that the two classes are balanced over the entire graph size.

In the USPS experiments, we randomly sample a different graph for each trial. While sampling the vertices of the graph, we ensure to select vertices equally from each concept class. We use a variety of concept classes 1 vs. 2, 2 vs. 3 and 4 vs. 7. We use the pairwise Euclidean distance as the weights for the MST construction. All the sampled graphs maintain the MST edges. In all the experiments on the datasets, the unweighted minimum spanning tree (MST) and "$K = 3$"-NN graph had their edge sets' "unioned" to create the resultant graph. The motivating reason being that most empirical experiments had shown competitive performance of algorithms at $K = 3$, while the MST guaranteed connectivity in the graph. Besides, MST based graphs are sparse in general, enabling computational efficient completion of the experiments. All the experiments were carried out in a quad-core processor notebooks (@2.30 GHz each) with 8GB RAM and 16 GB RAM.

4.4 Evaluation Criteria

We measure the performance of the algorithms by means of the instantaneous regret or per-round regret of the learning algorithm as compared with the optimal algorithm (lower the better). The instantaneous regret should sub-linearly reduce to zero. The instantaneous regret of the algorithm is measured against time. In our case, time indicates each unlabelled vertex queried in an iterative fashion by the environment, until all unlabelled vertices had been queried. Ideally, the more vertices has been queried and more side information obtained, the lower should be the instantaneous regret of the algorithms. In all the experiments, the hidden concept class distribution in the underlying graph is balanced.

4.5 Results

In the synthetic dataset of concentric squares experiment in Fig. 4, `Ising` always outperforms `Exp3`, `Exp4` and `LabProp`. `LabProp` and `Ising` are very competitive over uninformed competitors of `Exp3`, `Exp4`. `Exp3`, `Exp4` do not use the available side information to sample their action. Note, the overlapping squares create a difficult dataset where closely connected clusters of similar labels `white with intensity 1` are surrounded by clusters of opposite labels `black with intensity 0` around its boundary. Although, `LabProp` is good at exploiting connectivity, here we see that `Ising` captures the opposing boundary side information better than `LabProp`.

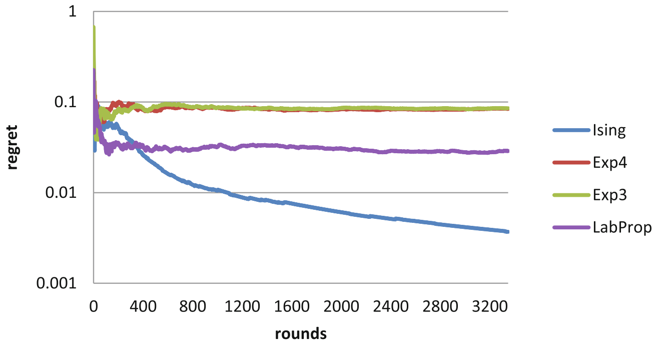


Fig. 4. Results on torus graph generated from `Squares` image with equal number of neighbours $K = 4$, $N = 3600, L = 250$.

Our dataset experiments begin with the `USPS 2 Vs.3` experiment with connectivity $K = 3$, available labels $L = 8$, and number of data points $N = 1000$. In Fig. 5 below, algorithms `Ising` and `LabProp` are very competitive when side information about more than half of the dataset is obtained. When the side information is very limited at the beginning of the game, `LabProp` outperforms `Ising`.

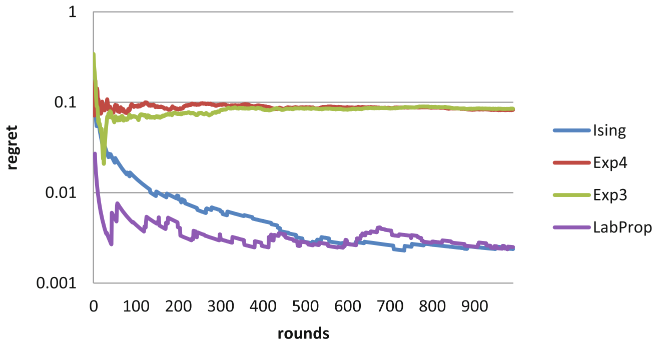
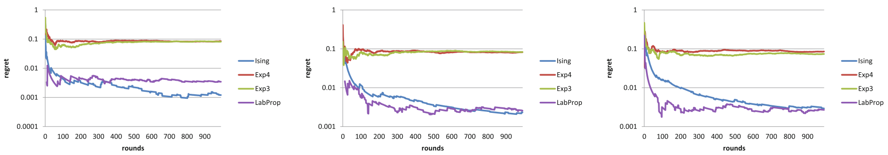


Fig. 5. `USPS 2 Vs.3 with` $K = 3, N = 1000, L = 8$

In Fig. 6 below, we test the behaviour of the algorithms with varying degree of connectivity. We vary the parameter K over a range to check how well the cluster size affects the performance. It is known from labelling over graph literature that with increasing K the behaviour deteriorates. Here, we see `Ising` outperforms `LabProp` for lower values of K, while `LabProp` wins for higher K.



(a) K=3 (b) K=5 (c) K=7

Fig. 6. `USPS 4 Vs.7`, with varying connectivity $K = 3, K = 5, K = 7$ on randomly sampled graphs with $N = 1000, L = 8$. The color coding is uniform over all the graphs and as indicated in (c) above.

In our experiments over the dataset `ISOLET`, we sample the graph from `ISOLET 1`. In Fig. 7, we observe that with $K = 3$ and $L = 128$, `Ising` outperforms `LabProp` throughout. The overall regret achieved in `ISOLET` is higher than the regret achieved in `USPS` as `ISOLET` is a noisy dataset.

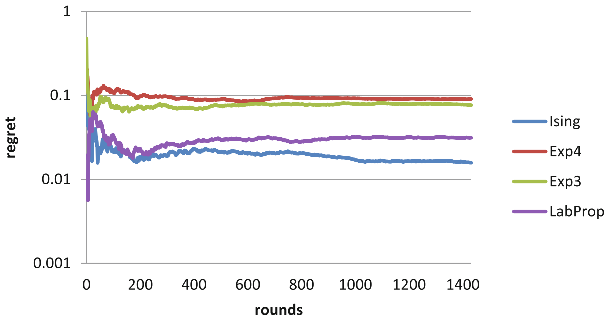
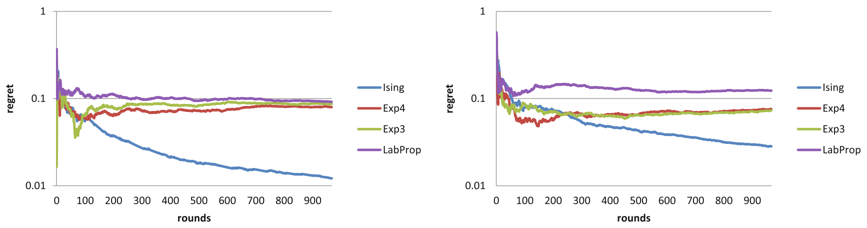


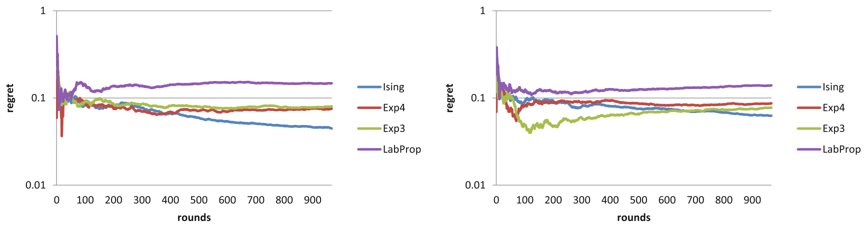
Fig. 7. Experiments on `ISOLET` with $K = 3, N = 1560, L = 128$

The following set of experiments in Fig. 8 and Fig. 9 test the robustness of our methods in presence of balanced noise. Our noise parameter nse is varied over the percentage range $s = 10, 20, 30, 40$. When noise is say x percent, we randomly eliminate the actions/edges in the graph (from existing connections) for which the noise is less than x percent, and add a balanced equal number of new actions (connections) to the graph. We see that the performance of `Ising` is the most robust across various noise levels. `LabProp` suffers with noise as it is heavily dependant on connectivity, and under performs in contrast to `Exp4` and `Exp3`. On the contrary, `Ising` uses the connectivity for side information, with its action selection unaffected with the introduction of noise. When the noise level increases, the performance of all the algorithms decrease uniformly.



(a) nse = 10 (b) nse = 20

Fig. 8. `USPS 1 vs. 2` Robustness Experiments with noise levels 10% and 20%



(a) nse = 30 (b) nse = 40

Fig. 9. `USPS 1 vs. 2` Robustness Experiments with noise levels 30% and 40%

5 Conclusion

There are real life scenarios where a core minimal subset of connections in a network is responsible for partitioning the graph. Such a core group could be a focus of targeted advertising or content-recommendation as that can have maximum influence on the network with a potential to go viral. Typically, there is a lot of available information in such settings that is potentially usable for detecting the changing partitioning set. We address such advertising and content recommendation challenges by casting the problem as an online Ising graph model of bandits with side information. We use the notion of *cut-size* as a regularity measure in the model to identify the partition and play the bandits game. The best case behaviour of the algorithm when there is a single partition is equivalent to the standard adversarial MAB. We show a polynomial algorithm where the label consistent "cut-size" can guide the sampling procedure. Further, we motivate a linear time exact algorithm for computing the max flow that also respects the label consistency. An interesting effect of the algorithm is that as long as the *cut-size* does not change, the learner keeps playing the same partition on the active action set (size smaller than the actual action set). The regret is then bounded by the number of times the cut changes during the entire game. This can be proven analytically, which we will like to pursue as future work.

References

1. Alamgir, M., von Luxburg, U.: Phase transition in the family of p-resistances. In: Shawe-Taylor, J., Zemel, R.S., Bartlett, P.L., Pereira, F.C.N., Weinberger, K.Q. (eds.) NIPS, pp. 379–387 (2011)
2. Amin, K., Kearns, M., Syed, U.: Graphical models for bandit problems (2012). arXiv preprint arXiv:1202.3782
3. Auer, P., Cesa-Bianchi, N., Freund, Y., Schapire, R.E.: Gambling in a rigged casino: the adversarial multi-armed bandit problem. In: Proceedings of the 36th Annual Symposium on Foundations of Computer Science, 1995, pp. 322–331. IEEE (1995)
4. Belkin, M., Matveeva, I., Niyogi, P.: Regularization and semi-supervised learning on large graphs. In: Shawe-Taylor, J., Singer, Y. (eds.) COLT 2004. LNCS (LNAI), vol. 3120, pp. 624–638. Springer, Heidelberg (2004)
5. Belkin, M., Niyogi, P.: Semi-supervised learning on riemannian manifolds. Mach. Learn. **56**(1–3), 209–239 (2004)
6. Blum, A., Chawla, S.: Learning from labeled and unlabeled data using graph mincuts. In: ICML, pp. 19–26 (2001)
7. Di Castro, D., Gentile, C., Mannor, S.: Bandits with an edge. In: CoRR, abs/1109.2296 (2011)
8. Ford, L.R., Fulkerson, D.R.: Maximal Flow through a Network. Canadian Journal of Mathematics **8**, 399–404 (1956). http://www.rand.org/pubs/papers/P605/
9. Gentile, C., Li, S., Zappella, G.: Online clustering of bandits (2014). arXiv preprint arXiv:1401.8257
10. Gentile, C., Orabona, F.: On multilabel classification and ranking with bandit feedback. The Journal of Machine Learning Research **15**(1), 2451–2487 (2014)
11. Herbster, M.: Exploiting cluster-structure to predict the labeling of a graph. In: Freund, Y., Györfi, L., Turán, G., Zeugmann, T. (eds.) ALT 2008. LNCS (LNAI), vol. 5254, pp. 54–69. Springer, Heidelberg (2008)
12. Herbster, M., Lever, G.: Predicting the labelling of a graph via minimum p-seminorm interpolation. In: Proceedings of the 22nd Annual Conference on Learning Theory (COLT 2009) (2009)
13. Herbster, M., Lever, G.: Predicting the labelling of a graph via minimum p-seminorm interpolation. In: COLT (2009)
14. Herbster, M., Lever, G., Pontil, M.: Online prediction on large diameter graphs. In: Advances in Neural Information Processing Systems, pp. 649–656 (2009)
15. Herbster, M., Pontil, M., Wainer, L.: Online learning over graphs. In: Proceedings of the 22nd international conference on Machine learning ICML 2005, pp. 305–312. ACM, New York (2005)
16. Nadler, B., Srebro, N., Zhou, X.: Statistical analysis of semi-supervised learning: the limit of infinite unlabelled data. In: NIPS, pp. 1330–1338 (2009)
17. Trevisan, L.: Lecture 15:cs261:optimization (2011). http://theory.stanford.edu/trevisan/cs261/lecture15.pdf
18. Valko, M., Munos, R., Kveton, B., Kocák, T.: Spectral bandits for smooth graph functions. In: 31th International Conference on Machine Learning (2014)
19. Zhu, X., Ghahramani, Z.: Towards semi-supervised classification with markov random fields. Tech. Rep. CMU-CALD-02-106, Carnegie Mellon University (2002)
20. Zhu, X., Ghahramani, Z., Lafferty, J.D.: Semi-supervised learning using gaussian fields and harmonic functions. In: ICML, pp. 912–919 (2003)