A Regularization Approach to Metrical Task Systems

Jacob Abernethy, Peter Bartlett, Niv Buchbinder and **Isabelle Stanton**

UC Berkeley and Microsoft Research

October 7, 2010

A Congested Highway

Consider driving down a congested n lane highway. You can observe the current congestion of each lane, and can not instantaneously switch to any other lane.

	1	**2**	**3**	$\cdots$	$n-1$	n
1)	1	0	1	$\cdots$	1	1

A Congested Highway

Consider driving down a congested n lane highway. You can observe the current congestion of each lane, and can not instantaneously switch to any other lane.

	1	**2**	**3**	$\cdots$	$n-1$	n
1)	1	0	1	$\cdots$	1	1
2)	1	0	1	$\cdots$	0	1

A Congested Highway

Consider driving down a congested n lane highway. You can observe the current congestion of each lane, and can not instantaneously switch to any other lane.

	1	**2**	**3**	$\cdots$	$n-1$	n
1)	1	0	1	$\cdots$	1	1
2)	1	0	1	$\cdots$	0	1
3)	1	0	0	$\cdots$	1	1

A Congested Highway

Consider driving down a congested n lane highway. You can observe the current congestion of each lane, and can not instantaneously switch to any other lane.

	1	**2**	**3**	$\cdots$	$n-1$	n
1)	1	0	1	$\cdots$	1	1
2)	1	0	1	$\cdots$	0	1
3)	1	0	0	$\cdots$	1	1
4)	0	1	1	$\cdots$	0	0

A Congested Highway

Consider driving down a congested n lane highway. You can observe the current congestion of each lane, and can not instantaneously switch to any other lane.

	1	**2**	**3**	$\cdots$	$n-1$	n
1)	1	0	1	$\cdots$	1	1
2)	1	0	1	$\cdots$	0	1
3)	1	0	0	$\cdots$	1	1
4)	0	1	1	$\cdots$	0	0
5)	0	1	0	$\cdots$	1	0

A Congested Highway

Consider driving down a congested n lane highway. You can observe the current congestion of each lane, and can not instantaneously switch to any other lane.

	1	**2**	**3**	$\cdots$	$n-1$	n
1)	1	0	1	$\cdots$	1	1
2)	1	0	1	$\cdots$	0	1
3)	1	0	0	$\cdots$	1	1
4)	0	1	1	$\cdots$	0	0
5)	0	1	0	$\cdots$	1	0
6)	1	1	0	$\cdots$	1	0
Experts Cost	4	3	3	$\cdots$	4	3

A Congested Highway

Consider driving down a congested n lane highway. You can observe the current congestion of each lane, and can not instantaneously switch to any other lane.

	1	**2**	**3**	$\cdots$	$n-1$	n
1)	1	0	1	$\cdots$	1	1
2)	1	0	1	$\cdots$	0	1
3)	1	0	0	$\cdots$	1	1
4)	0	1	1	$\cdots$	0	0
5)	0	1	0	$\cdots$	1	0
6)	1	1	0	$\cdots$	1	0
Experts Cost	4	3	3	$\cdots$	4	3
MTS Cost				$\cdots$		2

MTS, More Formally

Given n states, and a metric δ between them, a series of cost vectors $\mathbf{c}^1, \mathbf{c}^2, \cdots \mathbf{c}^t$ arrives. An algorithm sees the cost vector and then decides on a state.

The *cost* of some algorithm A is the total expected "servicing cost" plus the total "moving" cost, i.e.

$$\text{cost}_A(\mathbf{c}^1, \ldots, \mathbf{c}^T) := \sum_{t=1}^{T} \left(\mathbf{p}^t \cdot \mathbf{c}^t + \text{dist}_\delta(\mathbf{p}^t, \mathbf{p}^{t-1})\right)$$

The Work Vector and OPT

Definition

The *work vector* tracks the cost of the least expensive path that ends at state i at time t. Given a sequence of cost vectors $\mathbf{c}^1, \ldots, \mathbf{c}^T$, it is defined as:

$$\mathbf{W}^0 := \langle 0, \infty, \ldots, \infty \rangle \qquad W_i^t := \min_{j \in [n]} \left\{ W_j^{t-1} + \delta(i,j) + c_j^t \right\}$$

Definition

$$\mathrm{OPT}(\mathbf{c}^1, \ldots, \mathbf{c}^T) = \min_{i \in [n]} W_i^T$$

The Work Vector Example

	1	2	3	$\cdots$	$n-1$	n
1	1	0	1	$\cdots$	1	1
$\mathbf{w}^1$	1	2	2	$\cdots$	2	2
2	1	0	1	$\cdots$	0	1
$\mathbf{w}^2$	2	2	3	$\cdots$	2	3
3	1	0	0	$\cdots$	1	1
$\mathbf{w}^3$	3	2	3	$\cdots$	3	3

Competitive Ratio

Definition

An algorithm A has competitive ratio C if, for all cost sequences $\mathbf{c}^1, \mathbf{c}^2, \ldots, \mathbf{c}^T$ of any length T

$$\frac{\mathrm{cost}_A(\mathbf{c}^1, \mathbf{c}^2, \ldots, \mathbf{c}^T)}{\mathrm{cost}_{\mathrm{OPT}}(\mathbf{c}^1, \mathbf{c}^2, \ldots, \mathbf{c}^T)} \quad \leq \quad C$$

Why can't we use regret?

Bounds

- Any **deterministic** algorithm has a competitive ratio of at least $2n - 1$ on any metric [Borodin, Linial, Saks]
- The Work Function algorithm has a C.R. of $2n - 1$ [BLS]

Bounds

- Any **deterministic** algorithm has a competitive ratio of at least $2n-1$ on any metric [Borodin, Linial, Saks]
- The Work Function algorithm has a C.R. of $2n-1$ [BLS]
- The C.R. for a **randomized** algorithm is $\Omega(\log n/\log\log n)$ for any metric [Bartal, Bollobas, Mendel]
- Embedding a metric into a HST gives a $O(\log^2 n \log^2 \log n)$ competitive algorithm for any metric. [Bartal, Blum, Burch, Tompkins], [Fiat, Mendel]

Bounds

- Any **deterministic** algorithm has a competitive ratio of at least $2n-1$ on any metric [Borodin, Linial, Saks]
- The Work Function algorithm has a C.R. of $2n-1$ [BLS]
- The C.R. for a **randomized** algorithm is $\Omega(\log n / \log\log n)$ for any metric [Bartal, Bollobas, Mendel]
- Embedding a metric into a HST gives a $O(\log^2 n \log^2 \log n)$ competitive algorithm for any metric. [Bartal, Blum, Burch, Tompkins], [Fiat, Mendel]
- Any randomized algorithm has a C.R. of at least H_n on the **uniform metric**.[BLS]
- Marking [BLS] has C.R. $2H_n$, Exponential [Irani, Seiden] has C.R. $\ln n + O(\sqrt{\log n})$

Our Algorithmic Framework

Our aim is to design *randomized* algorithms under the following assumptions

- the algorithm is *work-based*: $\mathbf{p}^t = \mathbf{p}(\mathbf{W}^t)$

Our Algorithmic Framework

Our aim is to design *randomized* algorithms under the following assumptions

- the algorithm is *work-based*: $\mathbf{p}^t = \mathbf{p}(\mathbf{W}^t)$

Conjecture

There is an optimal work-based MTS algorithm for any metric against an oblivious adversary.

Our Algorithmic Framework

Our aim is to design *randomized* algorithms under the following assumptions

- the algorithm is *work-based*: $\mathbf{p}^t = \mathbf{p}(\mathbf{W}^t)$
- the algorithm is *reasonable*: if $W_i = W_j + \delta(i,j)$ then $p_i(\mathbf{W}) = 0$

Our Algorithmic Framework

Our aim is to design *randomized* algorithms under the following assumptions

- the algorithm is *work-based*: $\mathbf{p}^t = \mathbf{p}(\mathbf{W}^t)$
- the algorithm is *reasonable*: if $W_i = W_j + \delta(i,j)$ then $p_i(\mathbf{W}) = 0$

We make the following assumptions in order to make the analysis easier.

- the algorithm is *conservative*: If state i receives cost, then the probability at state j does not decrease.
- the cost vectors are *elementary*: every $\mathbf{c}^t = \alpha \mathbf{e}_i$
- the cost vectors are *reasonable*: if $\mathbf{c}^t = \alpha \mathbf{e}_i$ then $\alpha \leq W_j^t - W_i^t + \delta(i,j)$

Hedge as a Regularization Problem

We can solve the Experts setting by using regularization.

$$\mathbf{p}^t = \operatorname*{argmin}_{\mathbf{p}\in\Delta_n} \left(R(\mathbf{p}) + \lambda \sum_{s=1}^{t-1} \mathbf{p}\cdot\mathbf{l}^s\right)$$

If $R(\mathbf{p}) = \sum_{i\in[n]} p_i \log p_i$ is the negative entropy function, solving this results in the exponential weighted majority algorithm:

$$p_i^t = \frac{\exp\left(-\lambda \sum_{s=1}^{t-1} l_i^s\right)}{\sum_j \exp\left(-\lambda \sum_{s=1}^{t-1} l_j^s\right)}$$

Adapting MTS into a Regularized Objective

Our algorithm should be work-based, so we replace $\mathbf{l}^s$ with $\mathbf{W}$.

$$\mathbf{p}^t = \operatorname*{argmin}_{\mathbf{p}\in\Delta_n} \left(R(\mathbf{p}) + \lambda \mathbf{p}\cdot\mathbf{W}\right)$$

Why isn't this enough?

MTS as a Regularization Problem

We need a more sophisticated penalty function that goes to ∞ for supported states so that we can enforce *reasonableness*.

Definition

For any metric δ on $[n]$ and any $1-$Lipschitz vector $\mathbf{W}$ with respect to δ, we say that $f_i(\mathbf{W}, \lambda)$ is a Lipschitz penalty function if $f_i(\mathbf{W}, \lambda) \to \infty$ as $\min_j \left(W_j - W_i + \delta(i,j)\right) \to 0$

MTS as a Regularization Problem

We need a more sophisticated penalty function that goes to ∞ for supported states so that we can enforce *reasonableness*.

Definition

For any metric δ on $[n]$ and any $1-$Lipschitz vector $\mathbf{W}$ with respect to δ, we say that $f_i(\mathbf{W},\lambda)$ is a Lipschitz penalty function if $f_i(\mathbf{W},\lambda) \to \infty$ as $\min_j \Big(W_j - W_i + \delta(i,j)\Big) \to 0$

We can now pose MTS as a regularization problem:

$$\mathbf{p}(\mathbf{W}) = \operatorname*{argmin}_{\mathbf{p}} \left(R(\mathbf{p}) + \sum_i p_i f_i(\mathbf{W},\lambda)\right)$$

We must select the right $R(\mathbf{p})$ and $f_i(\mathbf{W},\lambda)$.

Our Penalty Functions

We analyze the algorithms that result from the following penalty functions

$$\begin{array}{lrcl} \text{(Alg 1)} & f_i(\mathbf{W},\lambda) & = & -\lambda\log(1+W_{\min}-W_i) \\ \text{(Alg 2)} & f_i(\mathbf{W},\lambda) & = & -\log(e^{\lambda(1+W_{\min}-W_i)}-1) \end{array}$$

Here, we are using the uniform metric which suffers l_1 moving costs, so $R(\mathbf{p}) = \sum_{i\in[n]} p_i \log p_i$.

Our Penalty Functions

We analyze the algorithms that result from the following penalty functions

$$\begin{array}{lrcl}(\text{Alg } 1) & f_i(\mathbf{W}, \lambda) & = & -\lambda \log(1 + W_{\min} - W_i) \\ (\text{Alg } 2) & f_i(\mathbf{W}, \lambda) & = & -\log(e^{\lambda(1+W_{\min}-W_i)} - 1)\end{array}$$

Here, we are using the uniform metric which suffers l_1 moving costs, so $R(\mathbf{p}) = \sum_{i\in[n]} p_i \log p_i$.

- Alg 1 is $e \ln n + 1$ competitive

$$\frac{(1 + W_{\min} - W_i)^{\lambda}}{Z}$$

- Alg 2 is $\ln n + 2 \ln \ln n + 1$ competitive

$$\frac{(e^{\lambda(1+W_{\min}-W_i)} - 1)}{Z}$$

Proof Technique

Observation

The cost faced by an algorithm when given a cost vector $\epsilon \mathbf{e}_i$ *is*

$$\epsilon p_i(\mathbf{W} + \epsilon \mathbf{e}_i) + \sum_{j \in [n] \setminus \{i\}} [p_j(\mathbf{W} + \epsilon \mathbf{e}_i) - p_j(\mathbf{W})] \delta(i, j)$$

As $\epsilon \rightarrow 0$, *the instantaneous cost becomes*

$$p_i(\mathbf{W}) + \sum_{j \in [n] \setminus \{i\}} \frac{\partial p_j(\mathbf{W})}{\partial W_i} \delta(i, j)$$

Proof Technique

To bound the cost of an algorithm, we need a potential function $\Phi(\mathbf{W})$ such that

$$\frac{\partial \Phi(\mathbf{W})}{\partial W_i} \geq p_i(\mathbf{W}) + \sum_{j \in [n] \setminus \{i\}} \frac{\partial p_j(\mathbf{W})}{\partial W_i} \delta(i, j)$$

Proof Technique

To bound the cost of an algorithm, we need a potential function $\Phi(\mathbf{W})$ such that

$$\frac{\partial \Phi(\mathbf{W})}{\partial W_i} \geq p_i(\mathbf{W}) + \sum_{j \in [n] \setminus \{i\}} \frac{\partial p_j(\mathbf{W})}{\partial W_i} \delta(i,j)$$

For Alg 2,

$$\Phi(\mathbf{W}) = cW_{\min} - \frac{1+\lambda}{\lambda} \log \sum_{i=1}^{n} (e^{\lambda(1+W_{\min}-W_i)} - 1)$$

where $\lambda = \ln n + 2 \log logn$ and $c = \lambda + 1 + o(1)$.

Summary

To use our framework for a general metric you should:

1. Analyze the metric to understand the moving costs. Pick a good regularizer.
2. Pick a good Lipschitz Penalty Function for the metric.
3. Solve the Regularized Objective
4. Integrate $p_i(\mathbf{W}) + \sum_{j \in [n] \setminus \{i\}} \frac{\partial p_j(\mathbf{W})}{\partial W_i} \delta(i,j)$ for all i. The potential is any function who's derivative upperbounds this for all i.