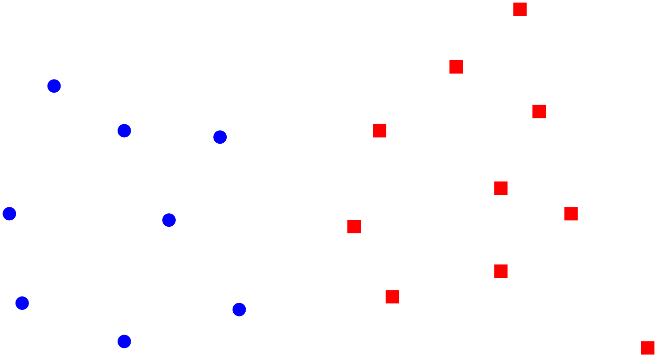
Selective sampling algorithms for cost-sensitive multiclass prediction

Alekh Agarwal

Microsoft Research

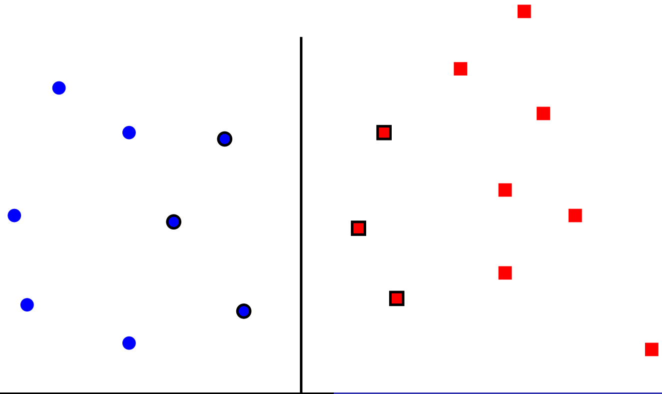
Why active learning?

- Standard setting - receive randomly sampled examples



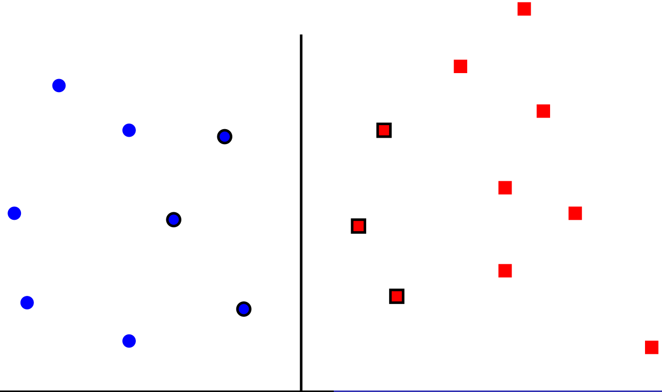
Why active learning?

- Standard setting - receive randomly sampled examples
- Not all data points are equally informative!



Why active learning?

- Standard setting - receive randomly sampled examples
- Not all data points are equally informative!
- Labelled data points are expensive, unlabelled cheap
  - Object recognition - images need **human labelling**
  - Protein interaction prediction - **lab test** for each protein pair
  - Web ranking - **human editors** to label relevant pages

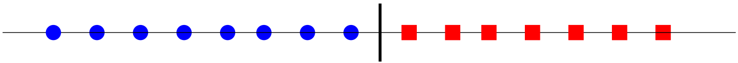


What is active learning?

- Sequentially query points with label uncertainty

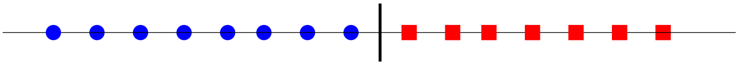
What is active learning?

- Sequentially query points with label uncertainty
  - Like random search vs. binary search

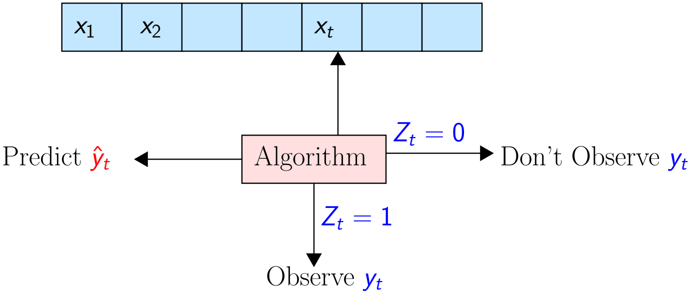


What is active learning?

- Sequentially query points with label uncertainty
  - Like random search vs. binary search
  - Example: sampling near decision boundary for linear separators



Online selective sampling paradigm



Filter examples online, querying only a subset of labels. Examples *not revisited*

Prior work

- Bulk of work in the binary setting
- Agnostic active learning
  - Atlas, Balcan, Beygelzimer, Cohn, Dasgupta, Hanneke, Hsu, Ladner, Langford, ...
- Linear selective sampling: Cesa-Bianchi, Gentile and co-authors

Prior work

- Bulk of work in the binary setting
- Agnostic active learning
  - Atlas, Balcan, Beygelzimer, Cohn, Dasgupta, Hanneke, Hsu, Ladner, Langford, ...
- Linear selective sampling: Cesa-Bianchi, Gentile and co-authors
- Empirical work in the multiclass setting: Jain and Kapoor (2009), Joshi et al. (2012), ...
- Relatively little theoretical work

This talk

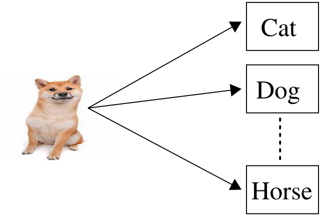
- Efficient algorithm in a multiclass GLM setting
- Analysis of regret and label complexity
- Sharp rates under Tsybakov-type noise condition
- Regret ranges between $\widetilde{O}(1/\sqrt{N_T})$ (noisy) to $\widetilde{O}(\exp(-c_0 N_T))$ (hard-margin)

This talk

- Efficient algorithm in a multiclass GLM setting
- Analysis of regret and label complexity
- Sharp rates under Tsybakov-type noise condition
- Regret ranges between $\widetilde{O}(1/\sqrt{N_T})$ (noisy) to $\widetilde{O}(\exp(-c_0 N_T))$ (hard-margin)
- Safety guarantee under model mismatch
- Numerical simulations

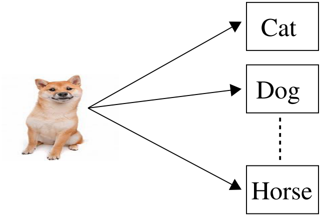
Multiclass prediction

- $x \in \mathbb{R}^d$, $y \in \{1, 2, \ldots, K\}$
- Only one label per example



Multiclass prediction

- $x \in \mathbb{R}^d$, $y \in \{1, 2, \ldots, K\}$
- Only one label per example
- Cost matrix $C \in \mathbb{R}^{K \times K}$
- Penalty C_{ij} for predicting label j when true label is i

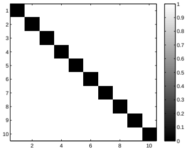


	Cat	Dog	Horse
Cat	0	1	10
Dog	1	0	10
Horse	10	10	0

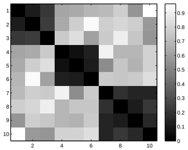
Structured cost matrices

- Often have block- or tree-structured cost matrices in applications

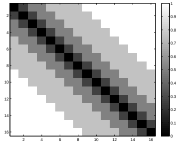
0/1



Block



Tree



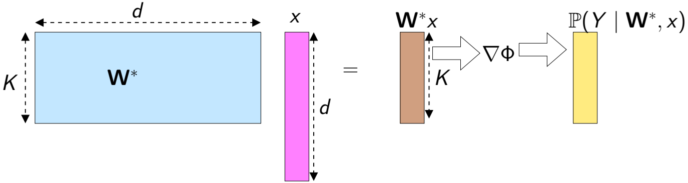
Multiclass GLM

- Weight matrix $\mathbf{W}^* \in \mathbb{R}^{K \times d}$
- Convex function $\Phi : \mathbb{R}^K \mapsto \mathbb{R}$

Definition (Multiclass GLM)

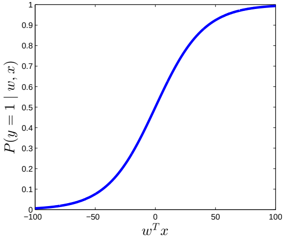
For every $x \in \mathbb{R}^d$, the class conditional probabilities follow the model

$$\mathbb{P}(Y = i \mid \mathbf{W}^*, x) = (\nabla\Phi(\mathbf{W}^* x))_i$$



Multiclass GLM intuition

Binary: $K = 2$. Φ is convex $\iff$ link function is monotone increasing. E.g.: logistic, linear, . . .



Example: multiclass logistic

- Define $\Phi(v) = \log(\sum_{i=1}^K \exp(v_i))$
- Obtain $(\nabla\Phi(v))_i = \exp(v_i)/(\sum_{j=1}^K \exp(v_j))$
- Yields the multinomial logit noise model

$$\mathbb{P}(Y = i \mid \mathbf{W}, x) = \frac{\exp(x^T \mathbf{W}^i)}{\sum_{j=1}^K \exp(x^T \mathbf{W}^j)}.$$

Loss function

- Given Φ, define the loss

$$\ell(\mathbf{W}x, y) = \Phi(\mathbf{W}x) - y^T\mathbf{W}x.$$

- Convex since Φ is convex
- *Fisher consistent:* $\mathbf{W}^*$ minimizes $\mathbb{E}[\ell(\mathbf{W}x, y) \mid \mathbf{W}^*, x]$ for each x

Loss function

- Given Φ, define the loss

$$\ell(\mathbf{W}x, y) = \Phi(\mathbf{W}x) - y^T\mathbf{W}x.$$

- Convex since Φ is convex
- *Fisher consistent:* $\mathbf{W}^*$ minimizes $\mathbb{E}[\ell(\mathbf{W}x, y) \mid \mathbf{W}^*, x]$ for each x

$$\begin{aligned}
\mathbb{E}[\nabla\ell(\mathbf{W}x, y) \mid \mathbf{W}^*, x] &= \mathbb{E}[\nabla\Phi(\mathbf{W}x) \mid \mathbf{W}^*, x] - \mathbb{E}[\nabla y^T\mathbf{W}x \mid \mathbf{W}^*, x] \\
&= \nabla\Phi(\mathbf{W}x)x^T - \mathbb{E}[y \mid \mathbf{W}^*, x]x^T \\
&= \nabla\Phi(\mathbf{W}x)x^T - \nabla\Phi(\mathbf{W}^*x)x^T
\end{aligned}$$

Score function

- Given a cost matrix C and Φ, define

$$S_{\mathbf{W}}^{x}(i) = -\sum_{j=1}^{K} \underbrace{C(j,i)}_{\text{cost of } i} \underbrace{(\nabla\Phi(\mathbf{W}x))_j}_{\text{probability of } j} .$$

- Negative expected cost of predicting j, when $\mathbf{W}$ is the true weight matrix
- Maximum score $\iff$ minimum expected cost

Score function

- Given a cost matrix C and Φ, define

$$S_{\mathbf{W}}^{x}(i) = -\sum_{j=1}^{K} \underbrace{C(j,i)}_{\text{cost of } i} \underbrace{(\nabla\Phi(\mathbf{W}x))_j}_{\text{probability of } j}.$$

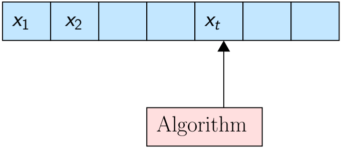
- Negative expected cost of predicting j, when $\mathbf{W}$ is the true weight matrix
- Maximum score $\iff$ minimum expected cost
- Bayes predictor predicts $\arg\max_{i=1}^{K} S_{\mathbf{W}^*}^{x}(i)$

CS-Selectron algorithm with general query function

- **Input:** Query function Q, cost matrix C, parameter $\gamma > 0$
- **Initialize:** $\mathbf{W}_1 = 0$, $M_1 = \gamma I / \gamma_\ell$
- For $t = 1, 2, \ldots T$

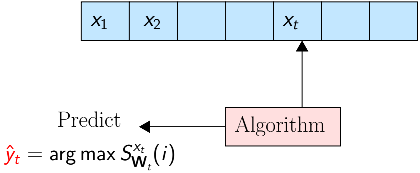
CS-Selectron algorithm with general query function

- **Input:** Query function Q, cost matrix C, parameter $\gamma > 0$
- **Initialize:** $\mathbf{W}_1 = 0$, $M_1 = \gamma I/\gamma_\ell$
- For $t = 1, 2, \ldots T$
  - **Observe** x_t, $H_{t+1} = H_t \cup \{x_t\}$



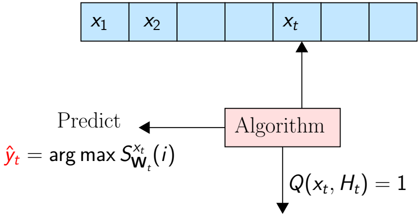
CS-Selectron algorithm with general query function

- For $t = 1, 2, \ldots T$
  - **Observe** x_t, $H_{t+1} = H_t \cup \{x_t\}$
  - **Predict** $\hat{y}_t = \arg\max_{i=1,2,\ldots,K} S_{\mathbf{W}_t}^{x_t}(i)$



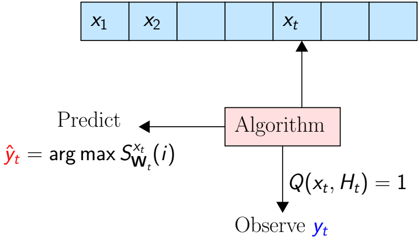
CS-Selectron algorithm with general query function

- For $t = 1, 2, \ldots T$
  - **Observe** x_t, $H_{t+1} = H_t \cup \{x_t\}$
  - **Predict** $\hat{y}_t = \arg\max_{i=1,2,\ldots,K} S^{x_t}_{\mathbf{W}_t}(i)$
  - If $\mathbf{Q}(\mathbf{x_t}, \mathbf{H_t}) = \mathbf{1}$, then



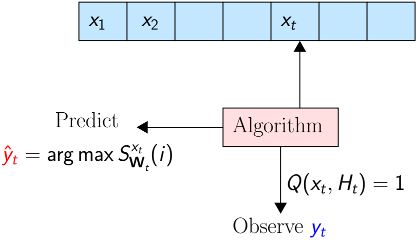
CS-Selectron algorithm with general query function

- For $t = 1, 2, \ldots T$
  - **Observe** x_t, $H_{t+1} = H_t \cup \{x_t\}$
  - **Predict** $\hat{y}_t = \arg\max_{i=1,2,\ldots,K} S^{x_t}_{\mathbf{W}_t}(i)$
  - If $\mathbf{Q}(\mathbf{x_t}, \mathbf{H_t}) = \mathbf{1}$, then
    - **Query** label y_t



CS-Selectron algorithm with general query function

- For $t = 1, 2, \ldots T$
  - **Observe** x_t, $H_{t+1} = H_t \cup \{x_t\}$
  - **Predict** $\hat{y}_t = \arg\max_{i=1,2,\ldots,K} S^{x_t}_{\mathbf{W}_t}(i)$
  - If $\mathbf{Q(x_t, H_t) = 1}$, then
    - **Query** label y_t
    - **Update** $\mathbf{W}_t, M_t$ and H_t



$$Z_t = 1,\; H_{t+1} = H_t \cup \{y_t\},\; M_{t+1} = M_t + x_t x_t^T$$

$$\mathbf{W}_{t+1} = \arg\min_{\mathbf{W} \in \mathcal{W}} \left\{ \sum_{s=1}^{t} Z_s \ell(\mathbf{W} x_s, y_s) + \gamma \|\mathbf{W}\|_F^2 \right\}.$$

Algorithm intuition

- Low-regret algorithm on queried examples
- Update ensures $\|\mathbf{W}_t - \mathbf{W}^*\|_{M_t}$ is small
- Query function ensures low regret on rounds with no queries

Query function: BBQ$_\epsilon$rule

- Variant of Cesa-Bianchi et al. (2009)

$$Q(x_t, H_t) = \mathbb{1}\left\{\eta_\epsilon \|x_t\|^2_{M_t^{-1}} \geq \epsilon^2\right\}$$

Query function: BBQ$_\epsilon$rule

- Variant of Cesa-Bianchi et al. (2009)

$$Q(x_t, H_t) = \mathbb{1}\left\{\eta_\epsilon \|x_t\|^2_{M_t^{-1}} \geq \epsilon^2\right\}$$

- Note: $\|\mathbf{W}^* x_t - \mathbf{W}_t x_t\|_2 \leq \|\mathbf{W}^* - \mathbf{W}_t\|_{M_t} \|x_t\|_{M_t^{-1}}$
- Queries points with large confidence intervals on the predictions

$Q(x_t, H_t) = 1$ $\qquad$ $Q(x_t, H_t) = 0$

Theoretical results: assumptions

Assumption

The function $\Phi(\cdot)$ *is* γ_ℓ*-strongly convex, that is for all* $u, v \in S \subseteq \mathbb{R}^K$, *we have*

$$\Phi(u) \geq \Phi(v) + \langle \nabla\Phi(v), (u - v) \rangle + \frac{\gamma_\ell}{2}\|u - v\|_2^2.$$

Theoretical results: assumptions

Assumption

The function $\Phi(\cdot)$ *is* γ_ℓ*-strongly convex, that is for all* $u, v \in S \subseteq \mathbb{R}^K$*, we have*

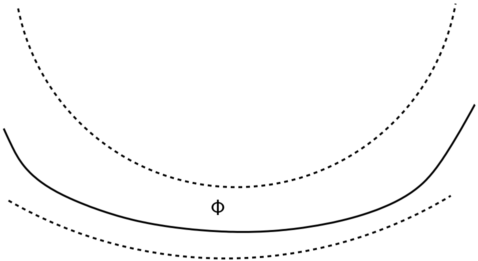
$$\Phi(u) \geq \Phi(v) + \langle \nabla\Phi(v), (u-v) \rangle + \frac{\gamma_\ell}{2}\|u-v\|_2^2.$$

Assumption

The function $\Phi(\cdot)$ *is* γ_u*-smooth, that is for all vectors* $u, v \in S \subseteq \mathbb{R}^K$*, we have*

$$\Phi(u) \leq \Phi(v) + \langle \nabla\Phi(v), (u-v) \rangle + \frac{\gamma_u}{2}\|u-v\|_2^2.$$

Theoretical results: assumptions



Theoretical results: assumptions

Assumption

$\forall x \in \mathcal{X}$, we have $\|x\|_2 \leq R$ and $\forall \mathbf{W} \in \mathcal{W}$, we have $\|\mathbf{W}^i\|_2 \leq \omega$ for all $i = 1, 2, \ldots, K$.

Theoretical results: setup

- Bound label complexity N_T and regret:

$$R_T = \sum_{t=1}^{T} (\mathbb{E}_t[C(Y_t, \hat{y}_t)] - \mathbb{E}_t[C(Y_t, y_t^*)])$$

Theoretical results: setup

- Bound label complexity N_T and regret:

$$R_T = \sum_{t=1}^{T} (\mathbb{E}_t[C(Y_t, \hat{y}_t)] - \mathbb{E}_t[C(Y_t, y_t^*)])$$

- Determined by fraction of *hard examples*

$$T_\epsilon = \{1 \leq t \leq T \ : \ S_{\mathbf{W}^*}^{x_t}(y_t^*) - S_{\mathbf{W}^*}^{x_t}(y_t^{'}) \leq \epsilon\}.$$

Main results: BBQ$_\epsilon$rule

Theorem (BBQ$_\epsilon$ query rule)

With probability at least $1 - 2\delta$,

$$R_T = \widetilde{O}\left(\epsilon T_\epsilon + \psi(C, \Phi)\,\frac{d}{\epsilon}\log\frac{1}{\delta}\right),$$

and label complexity is at most

$$N_T = \widetilde{O}\left(\frac{\gamma_u^2 d^2 K}{\gamma_\ell^2 \epsilon^2}\log\frac{1}{\delta}\right)$$

- Result holds for arbitrary sequence x_t

Query function: DGS rule

- BBQ_ϵ doesn't use the labels at all for querying!
- Variant of Dekel et al. (2010)
- Define

$$y_t^* = \arg\max_{i=1,\ldots,K} S^{x_t}_{\mathbf{W}^*}(i), \quad y_t^{'} = \arg\max_{i \neq y_t^*} S^{x_t}_{\mathbf{W}^*}(i)$$
$$\hat{y}_t = \arg\max_{i=1,\ldots,K} S^{x_t}_{\mathbf{W}_t}(i), \quad y_t^{''} = \arg\max_{i \neq \hat{y}_t} S^{x_t}_{\mathbf{W}_t}(i).$$

Query function: DGS rule

- BBQ$_\epsilon$ doesn't use the labels at all for querying!
- Variant of Dekel et al. (2010)
- Define

$$y_t^* = \arg\max_{i=1,\ldots,K} S_{\mathbf{W}^*}^{x_t}(i), \quad y_t^{'} = \arg\max_{i \neq y_t^*} S_{\mathbf{W}^*}^{x_t}(i)$$
$$\hat{y}_t = \arg\max_{i=1,\ldots,K} S_{\mathbf{W}_t}^{x_t}(i), \quad y_t^{''} = \arg\max_{i \neq \hat{y}_t} S_{\mathbf{W}_t}^{x_t}(i).$$

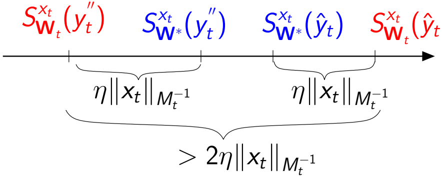
- Set $Q(x_t, H_t) = \mathbb{1}\left\{S_{\mathbf{W}_t}^{x_t}(\hat{y}_t) - S_{\mathbf{W}_t}^{x_t}(y_t^{''}) \leq 2\eta_{DGS} \|x_t\|_{M_t^{-1}}\right\}$

Query function: DGS rule

- BBQ_ϵ doesn't use the labels at all for querying!
- Variant of Dekel et al. (2010)
- Define

$$y_t^* = \arg\max_{i=1,\ldots,K} S_{\mathbf{W}^*}^{x_t}(i), \quad y_t^{'} = \arg\max_{i \neq y_t^*} S_{\mathbf{W}^*}^{x_t}(i)$$
$$\hat{y}_t = \arg\max_{i=1,\ldots,K} S_{\mathbf{W}_t}^{x_t}(i), \quad y_t^{''} = \arg\max_{i \neq \hat{y}_t} S_{\mathbf{W}_t}^{x_t}(i).$$

- Set $Q(x_t, H_t) = \mathbb{1}\left\{S_{\mathbf{W}_t}^{x_t}(\hat{y}_t) - S_{\mathbf{W}_t}^{x_t}(y_t^{''}) \leq 2\eta_{DGS}\,\|x_t\|_{M_t^{-1}}\right\}$
- Note: $|S_{\mathbf{W}_t}^{x_t}(i) - S_{\mathbf{W}^*}^{x_t}(i)| \leq \eta_{DGS}\,\|x_t\|_{M_t^{-1}}$



Main results: DGS rule

Theorem (DGS query rule)

With probability at least $1 - 2\delta$,

$$R_T = \widetilde{O}\left(\inf_{\epsilon > 0} \left\{ \epsilon T_\epsilon + \frac{\gamma_u^2 d}{\gamma_\ell^2 \epsilon} \log \frac{1}{\delta} \right\} \right),$$

and for any $\epsilon > 0$*, label complexity is at most*

$$N_T = \widetilde{O}\left(T_\epsilon + \frac{\gamma_u^2 d^2 K}{\gamma_\ell^2 \epsilon^2} \right)$$

- Can optimize over ϵ for the best bound

Multiclass Tsybakov noise condition

Specialize to $0/1$ costs for ease of presentation, and i.i.d. x_t

Assumption (Multiclass Tsybakov noise condition)

There exist $\epsilon_0 > 0$, $\alpha > 0$ *and some* c *such that the distribution* $\mathbb{P}$ *over* $\mathbb{R}^d$ *satisfies for all* $0 \leq \epsilon \leq \epsilon_0$,

$$\mathbb{P}\left((\nabla\Phi(\mathbf{W}^*X))_{y^*(X)} - (\nabla\Phi(\mathbf{W}^*X))_{y'(X)} \leq \epsilon\right) \leq c\,\epsilon^{\alpha}.$$

Multiclass Tsybakov noise condition

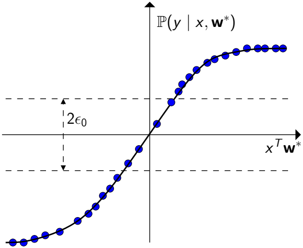
Specialize to $0/1$ costs for ease of presentation, and i.i.d. x_t

Assumption (Multiclass Tsybakov noise condition)

There exist $\epsilon_0 > 0$, $\alpha > 0$ and some c such that the distribution $\mathbb{P}$ over $\mathbb{R}^d$ satisfies for all $0 \leq \epsilon \leq \epsilon_0$,

$$\mathbb{P}\left((\nabla\Phi(\mathbf{W}^* X))_{y^*(X)} - (\nabla\Phi(\mathbf{W}^* X))_{y'(X)} \leq \epsilon \right) \leq c\,\epsilon^{\alpha}.$$

Ensures separation between class-conditional probabilities, controls T_ϵ. Pictorial illustration for the binary case



Results for low noise

Corollary

Under the multiclass Tsybakov condition, BBQ_ϵ rule yields with probability at least $1 - 2\delta$

$$\frac{R_T}{T} = \widetilde{O}\left(\left(\frac{\gamma_u^2 d^2 K}{\gamma_\ell^2 N_T}\right)^{\frac{1+\alpha}{2}} \right).$$

Results for low noise

Corollary

Under the multiclass Tsybakov condition, BBQ_ϵ rule yields with probability at least $1 - 2\delta$

$$\frac{R_T}{T} = \widetilde{O}\left(\left(\frac{\gamma_u^2 d^2 K}{\gamma_\ell^2 N_T}\right)^{\frac{1+\alpha}{2}} \right).$$

- Similar result for DGS rule
- $1/\sqrt{N_T}$ when $\alpha = 0$ and $\exp(-c_0 N_T)$ as $\alpha \to \infty$

Results for low noise

Corollary

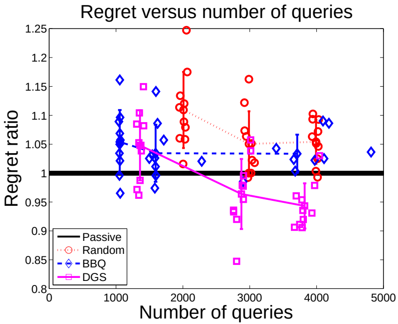
Under the multiclass Tsybakov condition, BBQ_ϵ rule yields with probability at least $1 - 2\delta$

$$\frac{R_T}{T} = \widetilde{O}\Big(\Big(\frac{\gamma_u^2 d^2 K}{\gamma_\ell^2 N_T}\Big)^{\frac{1+\alpha}{2}} \Big).$$

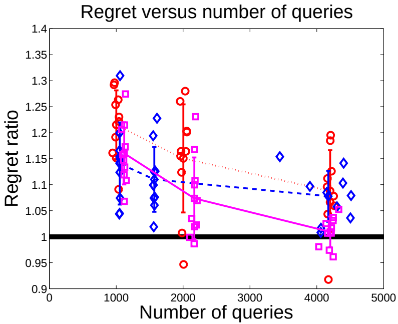
- Similar result for DGS rule
- $1/\sqrt{N_T}$ when $\alpha = 0$ and $\exp(-c_0 N_T)$ as $\alpha \to \infty$
- $R_T = \Omega(N_T^{-(1+\alpha)/2})$ under noise condition $\Rightarrow$ *optimality*

Numerical simulations

- Synthetic mixture of Gaussians data in $\mathbb{R}^{1000}$
- Evaluated BBQ, DGS, Random and Passive
- 0/1 cost matrix, multiclass logistic loss



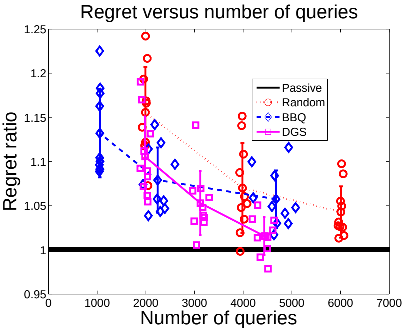
$K = 5$



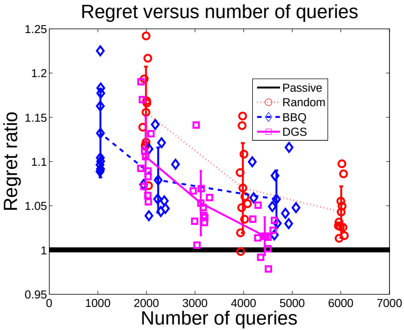
$K = 10$

Plots showing the ratio of active to passive regret, as a function of the number of queries

Model mismatch



Plot of regret ratio under model mismatch scenario



Plot of regret ratio under model mismatch scenario

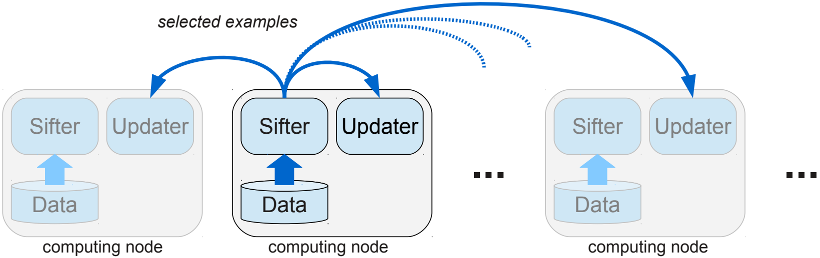
- Additional safety guarantee ensuring never worse than random under *model mismatch* in the paper

Conclusions

- Efficient active learning algorithm for cost-sensitive multiclass GLM
- Bounds on regret and label complexity
- Generalization of Tsybakov noise condition in binary case
- Optimal regret with the number of queries under noise condition
- Applications to communication efficient distributed learning

Para-active learning

- Sift for informative examples in parallel
- Update model on selected examples



Thank You