Learning to Classify with Missing and Corrupted Features

Ofer Dekel[1] Ohad Shamir[2]

[1]Microsoft Research, Redmond

[2]Department of Computer Science and Engineering
The Hebrew University of Jerusalem

ICML, July 2008

Motivation

- Training phase data and classification phase data are often not the same...
  - Clean, high quality training data, vs. noisy, missing and corrupted data to be classified.

Motivation

- Training phase data and classification phase data are often not the same...
  - Clean, high quality training data, vs. noisy, missing and corrupted data to be classified.
- Many Real-life examples
  - Sensor Failure
  - Medical Testing
  - ...

Motivation

- Training phase data and classification phase data are often not the same...
  - Clean, high quality training data, vs. noisy, missing and corrupted data to be classified.
- Many Real-life examples
  - Sensor Failure
  - Medical Testing
  - ...
- Goal: devising classifiers which are robust to classification phase noise.

Problem Setting

- Instances drawn i.i.d from some $\mathcal{X} \times \{\pm 1\}$, $\mathcal{X} \subseteq \mathbb{R}^n$.

Problem Setting

- Instances drawn i.i.d from some $\mathcal{X} \times \{\pm 1\}$, $\mathcal{X} \subseteq \mathbb{R}^n$.
- Linear margin-based classifiers: given instance $\mathbf{x}$, predict by $\text{sign}(\langle \mathbf{w}, \mathbf{x} \rangle + b)$, where $\mathbf{w} \in \mathbb{R}^n$, $b \in \mathbb{R}$.

Problem Setting

- Instances drawn i.i.d from some $\mathcal{X} \times \{\pm 1\}$, $\mathcal{X} \subseteq \mathbb{R}^n$.
- Linear margin-based classifiers: given instance $\mathbf{x}$, predict by $\text{sign}(\langle \mathbf{w}, \mathbf{x} \rangle + b)$, where $\mathbf{w} \in \mathbb{R}^n$, $b \in \mathbb{R}$.
- A 'clean', uncorrupted training set is available for learning a classifier $(\mathbf{w}, b)$.

Problem Setting

Learned Classifier

w	3	-4	6	2	-5	7	1	3	-2	-1	9	0

b	0.5

Test Set

	Instances												Labels
x_1	7	6	5	-1	-4	14	48	3	0	-8	1	11	y_1 +1
x_2	-6	-9	14	1	5	7	-9	7	8	-1	2	0	y_2 -1
x_3	10	7	13	6	4	8	-8	0	-1	9	0	3	y_3 +1

Problem Setting

Learned Classifier

w	3	-4	6	2	-5	7	1	3	-2	-1	9	0

b	0.5

Test Set

	Instances												Labels
x_1	7		5		-4	14	48		0	-8		11	y_1 +1
x_2	-6	-9	14	1	5	7	-9	7	8	-1	2	0	y_2 -1
x_3	10	7	13	6	4	8	-8	0	-1	9	0	3	y_3 +1

Problem Setting

Learned Classifier

w	3	-4	6	2	-5	7	1	3	-2	-1	9	0

b	0.5

Test Set

	Instances												Labels
x_1	7		5		-4	14	48		0	-8		11	y_1 +1
x_2	-6	-9		1	5			7	8	-1		0	y_2 -1
x_3	10	7	13	6	4	8	-8	0	-1	9	0	3	y_3 +1

Problem Setting

Learned Classifier

w	3	-4	6	2	-5	7	1	3	-2	-1	9	0

b	0.5

Test Set

	Instances											Labels
x_1	7		5		-4	14	48		0	-8		11	y_1 +1
x_2	-6	-9		1	5			7	8	-1		0	y_2 -1
x_3		7		6	4		-8	0		9	0	3	y_3 +1

Problem Setting

- In this talk, we'll focus on feature deletion: instead of $y_{\text{pred}} = \text{sign}(\sum_{j=1}^{n} w_j x_j + b)$,

$$y_{\text{pred}} = \text{sign}(\sum_{j \in J} w_j x_j + b)$$

  - *See paper for a more general feature corruption scenario.*

Problem Setting

- Adversary's power must be reasonably bounded for learning to be possible.

Problem Setting

- Adversary's power must be reasonably bounded for learning to be possible.
- Suppose each feature j has a fixed feature value $v_j \geq 0$.

Problem Setting

- Adversary's power must be reasonably bounded for learning to be possible.
- Suppose each feature j has a fixed feature value $v_j \geq 0$.
- Assumption: The adversary must leave intact a subset $J \subseteq \{1, \ldots, n\}$ of features, such that $V(J) := \sum_{j \in J} v_j \geq P$.

Previous work

- A similar setting considered in [Globerson & Roweis 2006],[Teo et. al. 2007]
  - Test-time feature deletion (with features values $v_j \equiv 1$).
  - A general framework for learning with invariances.

Theory - LP Formulation

- Worst-case "empirical risk" on training set:

$$\frac{1}{m}\sum_{i=1}^{m}\mathbf{1}\Big(\min_{J\,:\,V(J)\geq P} y_i\left(b+\textstyle\sum_{j\in J} w_j x_{i,j}\right)\ \leq\ 0\Big),$$

Theory - LP Formulation

- Worst-case "empirical risk" on training set:

$$\frac{1}{m}\sum_{i=1}^{m}\mathbf{1}\Big(\min_{J\,:\,V(J)\geq P} y_i\left(b+\textstyle\sum_{j\in J} w_j x_{i,j}\right)\ \leq\ 0\Big),$$

- An SVM-like formulation:

$$\begin{aligned}\min_{\mathbf{w},b,\boldsymbol{\xi}}\quad & \tfrac{1}{m\gamma}\textstyle\sum_{i=1}^{m}\xi_i \\ \text{s.t.}\quad & \forall\, i\in[m] \qquad \forall J\ :\ V(J)\geq P \\ & \qquad\qquad y_i\left(b+\textstyle\sum_{j\in J} w_j x_{i,j}\right)\ \geq\ \frac{\gamma V(J)}{P}-\xi_i\ \ , \\ & \forall\, i\in[m]\ \ \xi_i\ \geq\ 0\ , \quad \|\mathbf{w}\|_\infty\ \leq\ C\ \ .\end{aligned}$$

Theory - LP Formulation

- Worst-case "empirical risk" on training set:

$$\frac{1}{m}\sum_{i=1}^{m}\mathbf{1}\Big(\min_{J\,:\,V(J)\geq P} y_i\left(b+\textstyle\sum_{j\in J} w_j x_{i,j}\right)\ \leq\ 0\Big),$$

- An SVM-like formulation:

$$\begin{aligned}\min_{\mathbf{w},b,\boldsymbol{\xi}}\quad & \tfrac{1}{m\gamma}\textstyle\sum_{i=1}^{m}\xi_i \\ \text{s.t.}\quad & \forall\, i\in[m] \qquad \forall J\ :\ V(J)\geq P \\ & \qquad\qquad y_i\left(b+\textstyle\sum_{j\in J} w_j x_{i,j}\right)\ \geq\ \frac{\gamma V(J)}{P}-\xi_i\ \ , \\ & \forall\, i\in[m]\ \ \xi_i\ \geq\ 0\ , \qquad \|\mathbf{w}\|_\infty\ \leq\ C\ .\end{aligned}$$

Theory - LP Formulation

- Worst-case "empirical risk" on training set:

$$\frac{1}{m}\sum_{i=1}^{m}\mathbf{1}\Big(\min_{J\,:\,V(J)\geq P} y_i\left(b+\textstyle\sum_{j\in J} w_j x_{i,j}\right)\ \leq\ 0\Big),$$

- An SVM-like formulation:

$$\begin{aligned}\min_{\mathbf{w},b,\boldsymbol{\xi}}\quad & \tfrac{1}{m\gamma}\textstyle\sum_{i=1}^{m}\xi_i \\ \text{s.t.}\quad & \forall\, i\in[m] \qquad \forall J\ :\ V(J)\geq P \\ & \qquad\qquad y_i\left(b+\textstyle\sum_{j\in J} w_j x_{i,j}\right)\ \geq\ \frac{\gamma V(J)}{P}-\xi_i\ , \\ & \forall\, i\in[m]\ \ \xi_i\ \geq\ 0\,, \quad \|\mathbf{w}\|_\infty\ \leq\ C\ .\end{aligned}$$

Theory - LP Formulation

- Worst-case "empirical risk" on training set:

$$\frac{1}{m}\sum_{i=1}^{m}\mathbf{1}\Big(\min_{J\,:\,V(J)\geq P} y_i\left(b+\textstyle\sum_{j\in J} w_j x_{i,j}\right)\ \leq\ 0\Big),$$

- An SVM-like formulation:

$$\begin{aligned}\min_{\mathbf{w},b,\boldsymbol{\xi}}\quad & \tfrac{1}{m\gamma}\textstyle\sum_{i=1}^{m}\xi_i \\ \text{s.t.}\quad & \forall\, i\in[m] \qquad \forall J\ :\ V(J)\geq P \\ & \qquad\qquad y_i\left(b+\textstyle\sum_{j\in J} w_j x_{i,j}\right)\ \geq\ \frac{\gamma V(J)}{P}-\xi_i\ \ , \\ & \forall\, i\in[m]\ \ \xi_i\ \geq\ 0\ , \quad \|\mathbf{w}\|_\infty\ \leq\ C\ \ .\end{aligned}$$

- Problem: exponential-sized LP!

A Compact LP Formulation

A Compact LP Formulation

Derive a compact LP formulation, inspired by [Carr & Lancia 2000]

- For any $(\mathbf{w}, b, \mathbf{x}_i, y_i, \xi_i)$, does it satisfy the constraints

$$\forall J : V(J) \geq P \quad y_i \left(b + \sum_{j \in J} w_j x_{i,j} \right) \geq \frac{\gamma V(J)}{P} - \xi_i \quad ?$$

A Compact LP Formulation

Derive a compact LP formulation, inspired by [Carr & Lancia 2000]

- For any $(\mathbf{w}, b, \mathbf{x}_i, y_i, \xi_i)$, does it satisfy the constraints

$$\forall J : V(J) \geq P \quad y_i \left(b + \sum_{j \in J} w_j x_{i,j} \right) \geq \frac{\gamma V(J)}{P} - \xi_i \quad ?$$

- Rewrite as an integer program

$$\min_{\boldsymbol{\tau} \in \{0,1\}^n} \quad \left[y_i b + \sum_{j=1}^n \tau_j \left(y_i w_j x_{i,j} - \frac{\gamma v_j}{P} \right) \right] \geq -\xi_i$$

$$\text{s.t.} \quad \sum_{j=1}^n \tau_j v_j \geq P$$

A Compact LP Formulation

Derive a compact LP formulation, inspired by [Carr & Lancia 2000]

- For any $(\mathbf{w}, b, \mathbf{x}_i, y_i, \xi_i)$, does it satisfy the constraints

$$\forall J : V(J) \geq P \quad y_i \left(b + \sum_{j \in J} w_j x_{i,j} \right) \geq \frac{\gamma V(J)}{P} - \xi_i \quad ?$$

- Rewrite as an integer program

$$\min_{\boldsymbol{\tau} \in \{0,1\}^n} \quad \left[y_i b + \sum_{j=1}^n \tau_j \left(y_i w_j x_{i,j} - \frac{\gamma v_j}{P} \right) \right] \geq -\xi_i$$

$$\text{s.t.} \quad \sum_{j=1}^n \tau_j v_j \geq P$$

- Relax into a LP

$$\min_{\boldsymbol{\tau} \in [0,1]^n} \quad \left[y_i b + \sum_{j=1}^n \tau_j \left(y_i w_j x_{i,j} - \frac{\gamma v_j}{P} \right) \right] \geq -\xi_i$$

$$\text{s.t.} \quad \sum_{j=1}^n \tau_j v_j \geq P$$

A Compact LP Formulation

- After a duality transform and some more technical work, original constraint set is replaced by $O(mn)$ constraints:

$$\begin{aligned}
\forall\, i \in [m] \quad & P\lambda_i \;-\; \textstyle\sum_{j=1}^{n} \alpha_{i,j} + y_i b \;\geq -\xi_i \\
\forall\, i \in [m]\; \forall\, j \in [n] \quad & y_i w_j x_{i,j} - \frac{\gamma v_j}{P} \;\geq\; \lambda_i v_j - \alpha_{i,j} \\
\forall\, i \in [m]\; \forall\, j \in [n] \quad & \alpha_{i,j} \geq 0\ , \\
\forall\, i \in [m] \quad & \lambda_i \geq 0 \;\text{ and }\; \xi_i \geq 0 \\
& \|\mathbf{w}\|_\infty \leq\; C\ ,
\end{aligned}$$

Approximation Guarantee

Assume for simplicity $\|\mathbf{x}_i\| \leq 1$ for all i.

Theorem

Approximation Guarantee

Assume for simplicity $\|\mathbf{x}_i\| \leq 1$ for all i.

Theorem

- *If $\forall j, v_j \in \{0, 1\}$, compact formulation and exponential formulation have equal optimal values (i.e. average hinge loss).*

Approximation Guarantee

Assume for simplicity $\|\mathbf{x}_i\| \leq 1$ for all i.

Theorem

- *If $\forall j, v_j \in \{0, 1\}$, compact formulation and exponential formulation have equal optimal values (i.e. average hinge loss).*
- *Otherwise, the difference in the optimal values is at most C/γ.*

Generalization Bounds

$\ell_\gamma(\mathbf{w}, \mathbf{x}, y)$ - indicator for margin of at least $\gamma V(J)/P$ on adversarially treated $(\mathbf{x}, y)$.

Theorem

Let $\hat{R}_\gamma = \frac{1}{m}\sum_{i=1}^{m} \ell_\gamma(\mathbf{w}, \mathbf{x}_i, y_i)$.
The expected $0/1$ *loss on the feature-deleted test set is upper bounded (w.p.* $1-\delta$*) by*

$$\hat{R}_\gamma + O\left(\sqrt{\frac{\hat{R}_\gamma\left(\ln\left(\frac{m}{\delta}\right) + n\left(\ln\left(\frac{C}{\gamma}\right)+1\right)\right)}{m}}\right).$$

Online-to-batch Algorithm

Online-to-batch Algorithm

- Corrupted instances presented sequentially to the algorithm.

Online-to-batch Algorithm

- Corrupted instances presented sequentially to the algorithm.
- On round i, algorithm predicts $\text{sign}(b_i + \sum_{j \in J} w_{i,j} x_{i,j})$

Online-to-batch Algorithm

- Corrupted instances presented sequentially to the algorithm.
- On round i, algorithm predicts $\text{sign}(b_i + \sum_{j \in J} w_{i,j} x_{i,j})$
- Suffers hinge loss:

$$\xi(\mathbf{w}, b, \mathbf{x}, y) := \left[\max_{J \,:\, V([n] \setminus J) \leq N} \frac{\gamma V(J)}{P} - y \left(b + \sum_{j \in J} w_j x_j \right) \right]_+ ,$$

Online-to-batch Algorithm

- Corrupted instances presented sequentially to the algorithm.
- On round i, algorithm predicts $\text{sign}(b_i + \sum_{j \in J} w_{i,j} x_{i,j})$
- Suffers hinge loss:

$$\xi(\mathbf{w}, b, \mathbf{x}, y) := \left[\max_{J \,:\, V([n] \setminus J) \leq N} \frac{\gamma V(J)}{P} - y \left(b + \sum_{j \in J} w_j x_j \right) \right]_+ ,$$

Online-to-batch Algorithm

- Corrupted instances presented sequentially to the algorithm.
- On round i, algorithm predicts $\text{sign}(b_i + \sum_{j \in J} w_{i,j} x_{i,j})$
- Suffers hinge loss:

$$\xi(\mathbf{w}, b, \mathbf{x}, y) := \left[\max_{J \,:\, V([n] \setminus J) \leq N} \frac{\gamma V(J)}{P} - y \left(b + \sum_{j \in J} w_j x_j \right) \right]_+ ,$$

- Update step:

$$j \in [n] \quad w_{i+1,j} = \begin{cases} \left[w_{i,j} + y_i \tau x_{i,j} \right]_{\pm C} & \text{if } j \in J_i \\ w_{i,j} & \text{otherwise} \end{cases}$$

$$b_{i+1} = \left[b_i + y_i \tau \right]_{\pm C}$$

Online-to-batch Algorithm

- Choosing τ according to the cumulative loss bound: $\frac{1}{\gamma m}\sum_{i=1}^{m}\xi(\mathbf{w}_i, b_i, \mathbf{x}_i, y_i)$ upper bounded by

$$\leq \frac{1}{\gamma m}\sum_{i=1}^{m}\xi(\mathbf{w}^\star, b^\star, \mathbf{x}_i, y_i) \;+\; \frac{C}{\gamma}\sqrt{\frac{2(n+1)}{m}}.$$

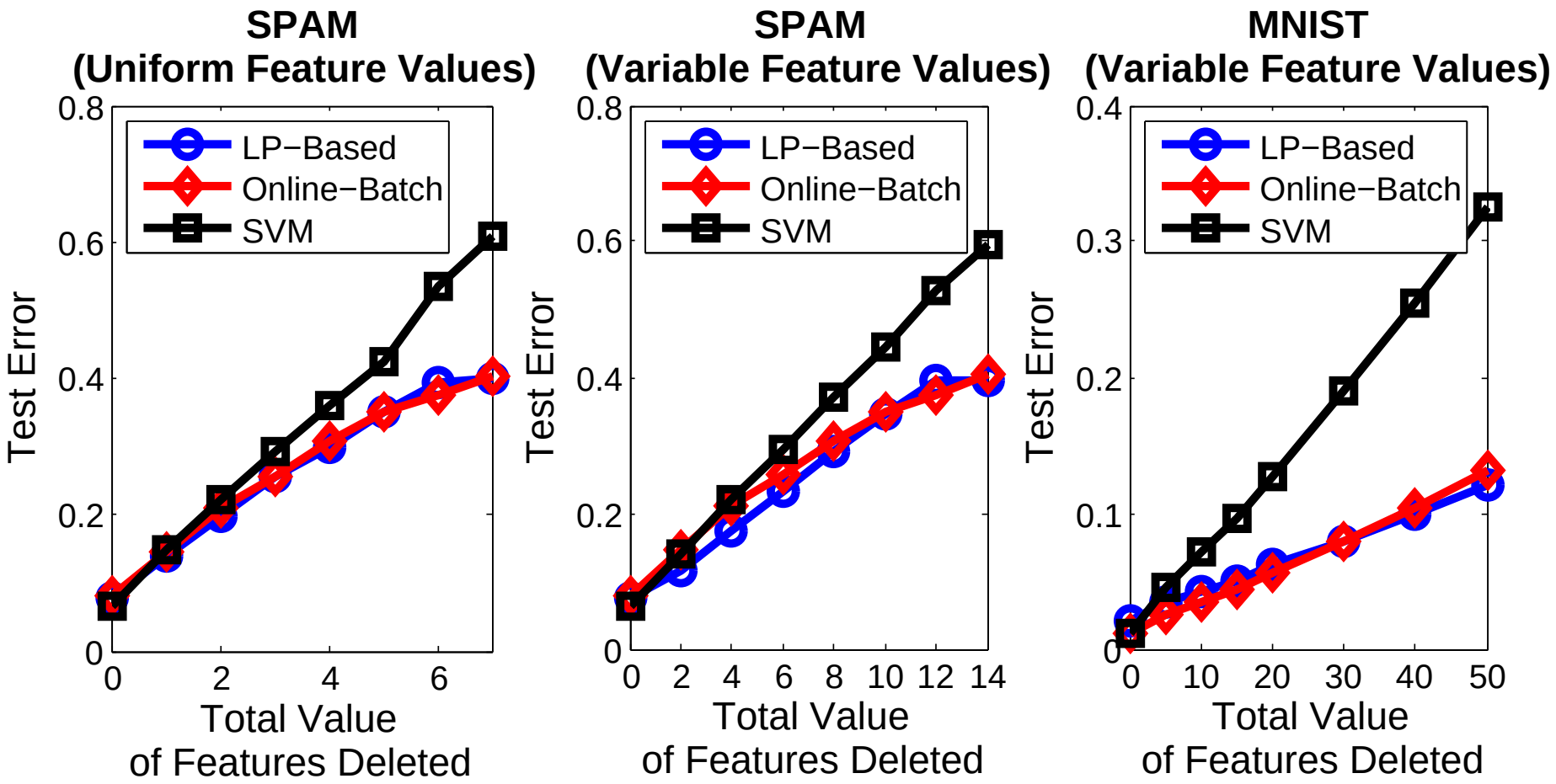
Online-to-batch Algorithm

- Choosing τ according to the cumulative loss bound: $\frac{1}{\gamma m}\sum_{i=1}^{m}\xi(\mathbf{w}_i, b_i, \mathbf{x}_i, y_i)$ upper bounded by

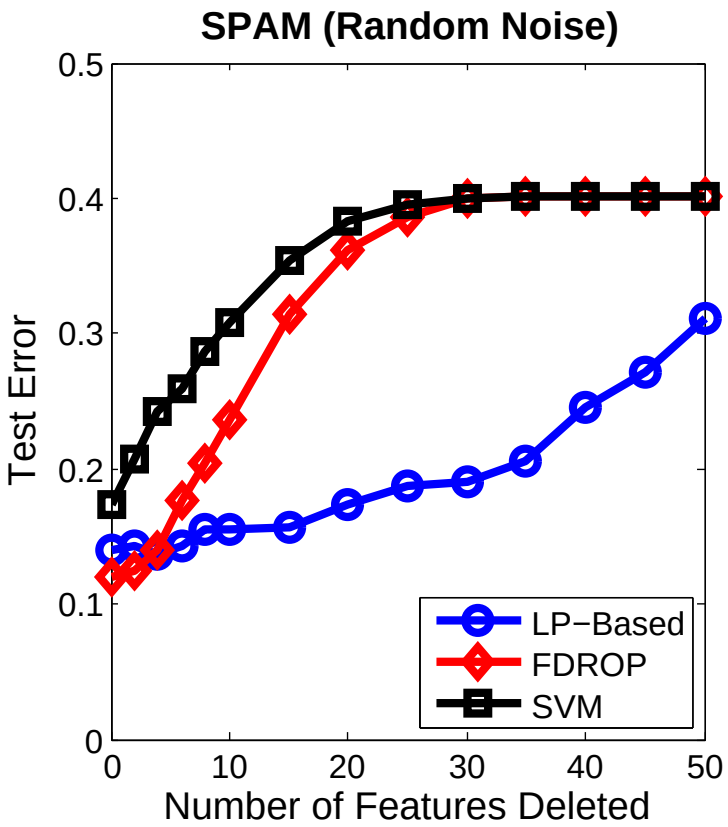
$$\leq \frac{1}{\gamma m}\sum_{i=1}^{m}\xi(\mathbf{w}^\star, b^\star, \mathbf{x}_i, y_i) \;+\; \frac{C}{\gamma}\sqrt{\frac{2(n+1)}{m}}.$$

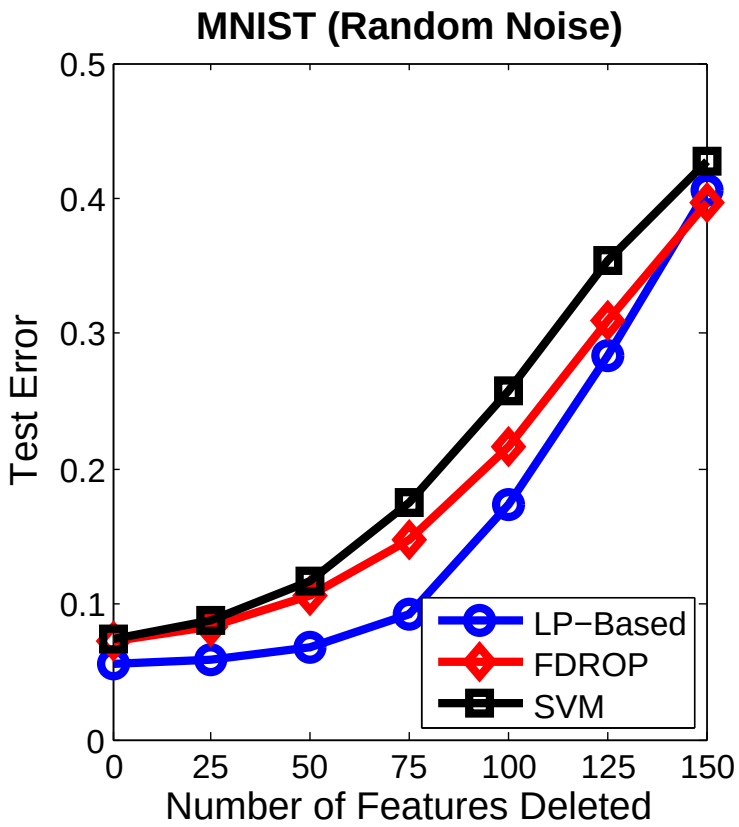
- Online to batch procedure: Run online algorithm on a randomly permuted sequence of the training set, and choose average hypothesis: $\bar{\mathbf{w}} = \frac{1}{m}\sum_{i=1}^{m}\mathbf{w}_{i-1}$ and $\bar{b} = \frac{1}{m}\sum_{i=1}^{m} b_{i-1}$.

Experiments

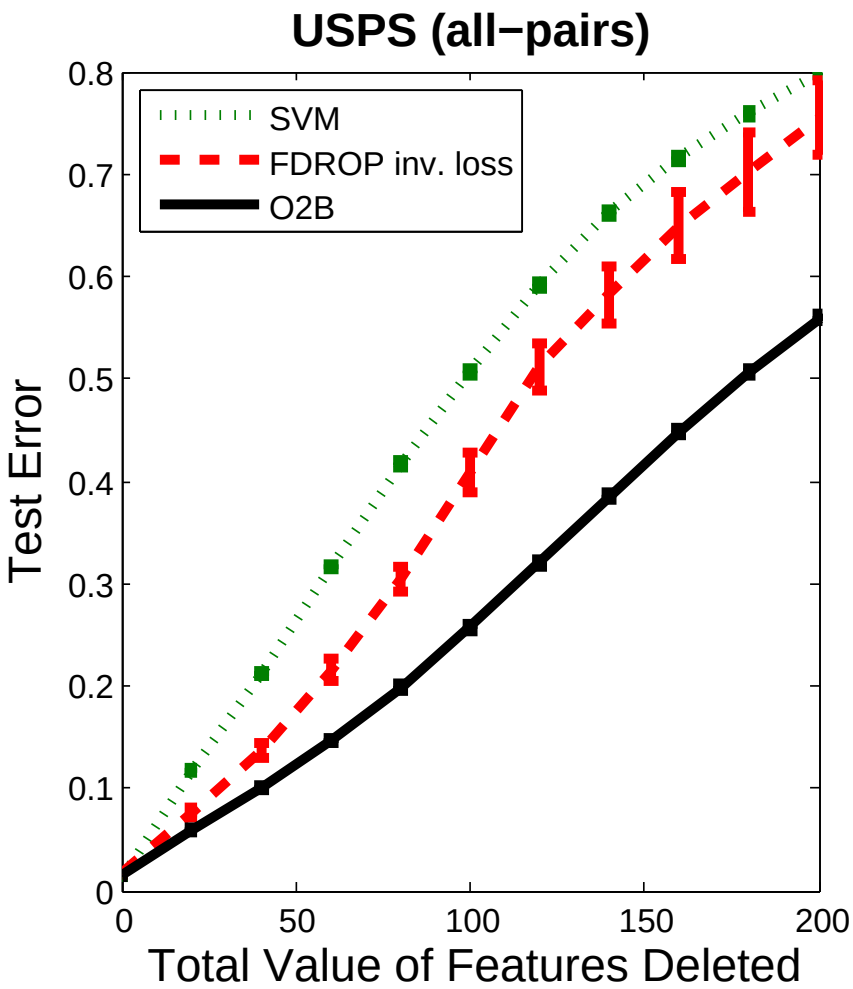


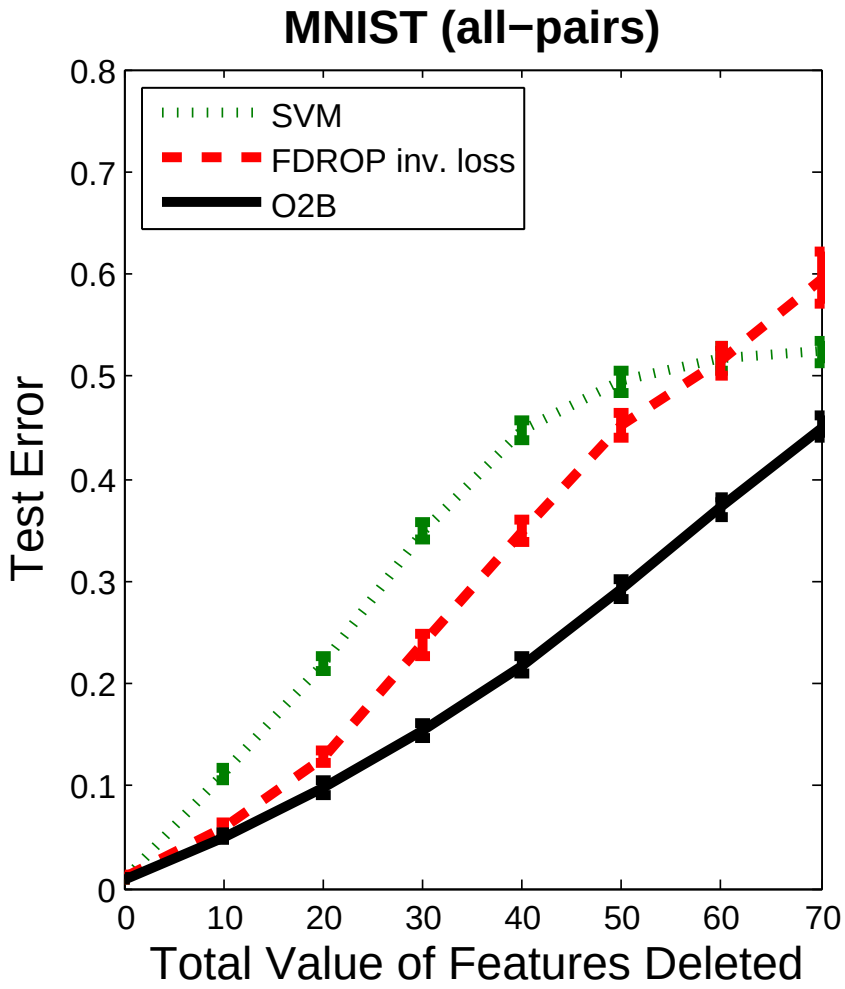
Experiments





Experiments





Conclusions

- Presented 2 algorithms which successfully withstand adversarial feature manipulation at the classification phase.

Conclusions

- Presented 2 algorithms which successfully withstand adversarial feature manipulation at the classification phase.
- Takes into account differing feature values.

Conclusions

- Presented 2 algorithms which successfully withstand adversarial feature manipulation at the classification phase.
- Takes into account differing feature values.
- Approximation and generalization guarantees.

Conclusions

- Presented 2 algorithms which successfully withstand adversarial feature manipulation at the classification phase.
- Takes into account differing feature values.
- Approximation and generalization guarantees.
- ∞-norm provides robustness at the expense of sparsity.

Thank You!

Withstanding Feature Corruption

- Our formulation allows us to handle feature corruption as well.
- Assume corrupted features are replaced by samples from an arbitrary bounded noise distribution.

Withstanding Feature Corruption

- Our formulation allows us to handle feature corruption as well.
- Assume corrupted features are replaced by samples from an arbitrary bounded noise distribution.

Observation

Correct prediction boils down to achieving an $\Omega(\sqrt{n})$ margin on the uncorrupted features.

Withstanding Feature Corruption

- Our formulation allows us to handle feature corruption as well.
- Assume corrupted features are replaced by samples from an arbitrary bounded noise distribution.

Observation

Correct prediction boils down to achieving an $\Omega(\sqrt{n})$ margin on the uncorrupted features.

- But margin scales linearly with n with feature-redundant data.
- Conclusion: Eventually, feature corruption scenario may not be much harder than feature deletion.