Automatic Polynomial Expansions

What?

- `vw` can build polynomial decision surfaces:

$$x \mapsto 2 \cdot \mathbb{1}[x_1 x_2 - 3x_1 x_4 x_5 + 7x_2 x_4 x_9 \geq 0] - 1.$$

What?

- `vw` can build polynomial decision surfaces:

$$x \mapsto 2 \cdot \mathbb{1}[x_1 x_2 - 3 x_1 x_4 x_5 + 7 x_2 x_4 x_9 \geq 0] - 1.$$

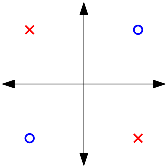
- Just add `--stage_poly` to your command line.

Why?

- Polynomial features facilitate many problems (e.g., see kaggle forums).

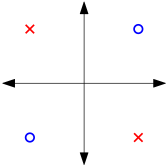
Why?

- Polynomial features facilitate many problems (e.g., see kaggle forums).



Why?

- Polynomial features facilitate many problems (e.g., see kaggle forums).

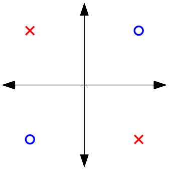


- `vw` already had (manual) polynomials of low degree:

```
vw -q ff (..)
vw --cubic fff (..)
```

Why?

- Polynomial features facilitate many problems (e.g., see kaggle forums).



- `vw` already had (manual) polynomials of low degree:

```
vw -q ff (..)
vw --cubic fff (..)
```

- `--stage_poly` vs. `--ksvm`: different bias.

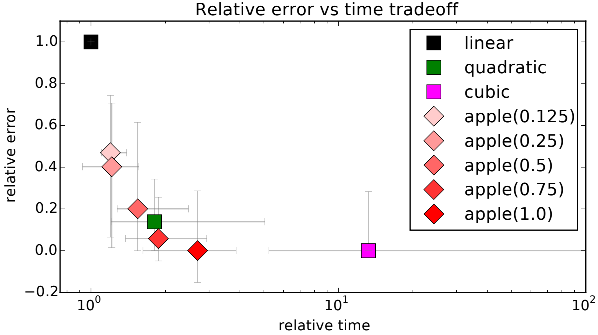
Does it work?

Theory. Loss minimization guarantee.

Does it work?

Theory. Loss minimization guarantee.

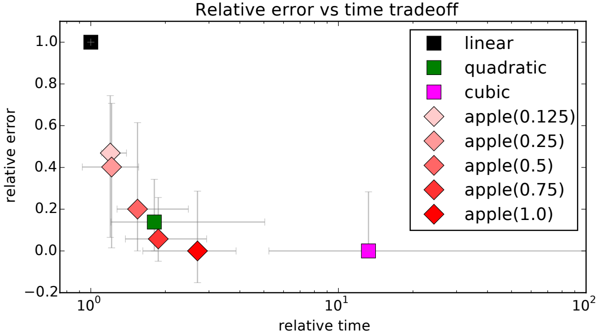
Practice.



Does it work?

Theory. Loss minimization guarantee.

Practice.



More Info. NIPS 2014:

`Scalable Nonlinear Learning with Adaptive Polynomial Expansions. Alekh Agarwal, Alina Beygelzimer, Daniel Hsu, John Langford, Matus Telgarsky.`

Usage

```
--stage_poly
```

Usage

```
--stage_poly
```

```
--sched_exponent arg1
```

Usage

```
--stage_poly

--sched_exponent arg1

--batch_sz arg2
```

Usage

```
--stage_poly

--sched_exponent arg1

--batch_sz arg2

--batch_sz_no_doubling
```

Hacking

- Different types of features.

Hacking

- Different types of features.
- Different support search.

More info @ github wiki & NIPS paper.

More info @ github wiki & NIPS paper.