Feedback on project 1

Report - plots

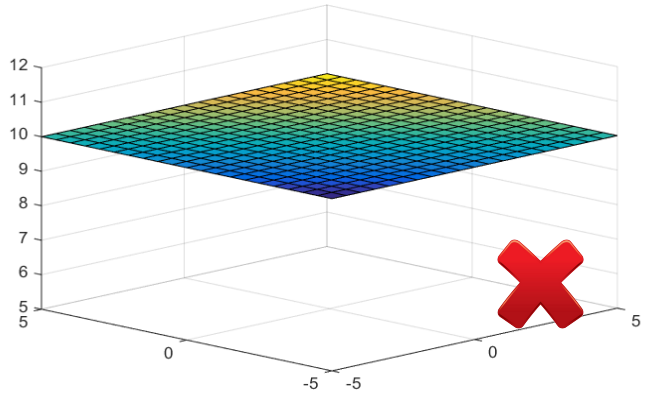
Checklist for a good plot :

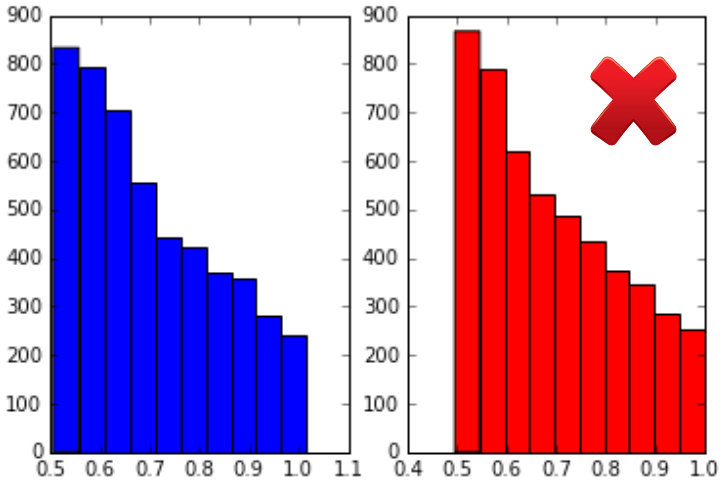
- Readable
  - Scale of axis
  - Font size
  - Lines are distint
    Use markers, color, dashes, ...

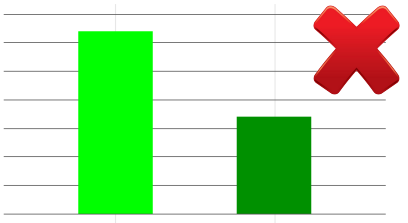
- Understandable
  - Labels
  - Legend
  - Title

- Context is provided (What, Why)

- Actually useful







Report - writing

- Use present tense, be factual.
    Avoid intros like
    « Since the beginning of time, physicists have tried to figure out the true model behind reality »
    Go to the point of the report : Your work.

- Don't try to cram everything in. Only relevant stuff.
    Avoid chronological epic tales of « We did X and then Y and then Z but Z did not work ».

- Use math notation only when necessary.
    Don't introduce notation when words are sufficient.
    Don't introduce notation you're not going to re-use.
    Use Φ, not phi (ϕ in Latex. Check [1])

- Have someone that did not write your report read it.
    It's easy to be convinced that your sentences make perfect sense after rewriting them 10 times.
    Check that your sentences generalize to other people as well.

- Reporting error from cross-validation :
    Don't : report with machine precision : 82.37182237182839
    Do : report mean and std dev. : 82.37 ± 0.02

[1] : detexify.kirelabs.org/classify.html

Report - Showing that you understand your work

Some stuff that got tried during project 1 :

- Outlier removal
- PCA
- Different basis functions
- Changing the link function
- Feature selection
- Neural Networks
- Other feature transformations
- Multiple models for differents missing data patterns

All interesting, but sometime lacking proper motivation for why

---

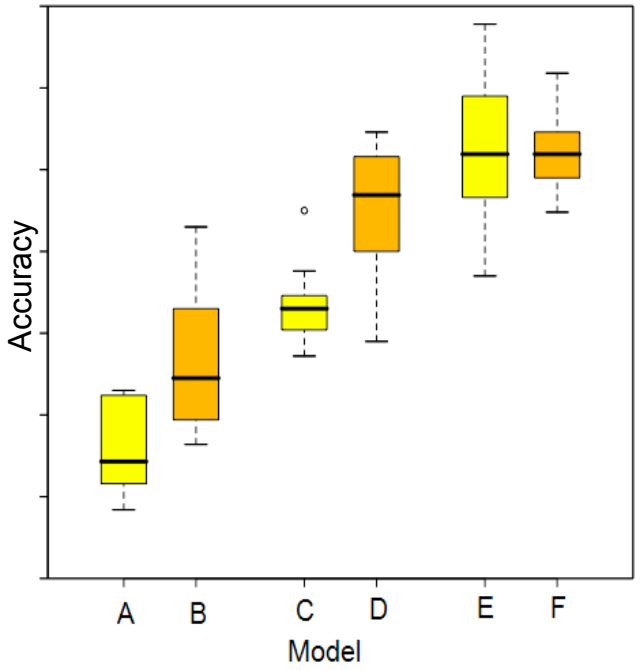
Checklist for a good presentation of a method in the report :

- What : Explain the problem that you are trying to solve.
- Why : Explain why this problem is important. Why are you fixing X instead of Y ?
- How : Explain what you do to solve it. How does your method solve X ?
- Results : Show that it works by comparing error.

Report - Show that your method works

- Identify core steps in your procedure that lead to your results
- Show how much they matter by showing how much they improve your model

Model	Accuracy
A: Baseline (Random guess)	0.743 ± 0.001
B: Simple logistic regression	0.782 ± 0.002
C: Handling NaN	0.797 ± 0.002
D: Polynomial Expansion (D=7)	0.817 ± 0.011
E: Regularized (D=9, λ=10^{-2})	0.821 ± 0.007
F: Magic Sauce™	0.839 ± 0.014



Optimization problem : Logistic regression is impossible

Possible cause : Bad computations (especially Newton's method)

$$\mathbf{H}(\mathbf{w}) = \mathbf{X}\mathbf{S}\mathbf{X}^\top$$

If you build S and compute XSX^T, is gives you
[DxN][NxN][NxD] -> $O(DN^2)$

But S is a diagonal matrix ! Use either :
- Elementwise multiplication
- Sparse matrices

-> gives you O(DN)

Other possible cause : Bad choice of optimization method

Logistic regression is not Newton's method

Machine learning setup :

- Cost function (MSE, MAE, log loss, ...)
- Model (linear $w^T x$, basis expansion $\Phi^T x$, ...)
- Link function (sigmoid, multinomial, tanh, exp, ...)
- Optimization procedure
  - Specialized (Least squares)
  - General (Gradient descent)

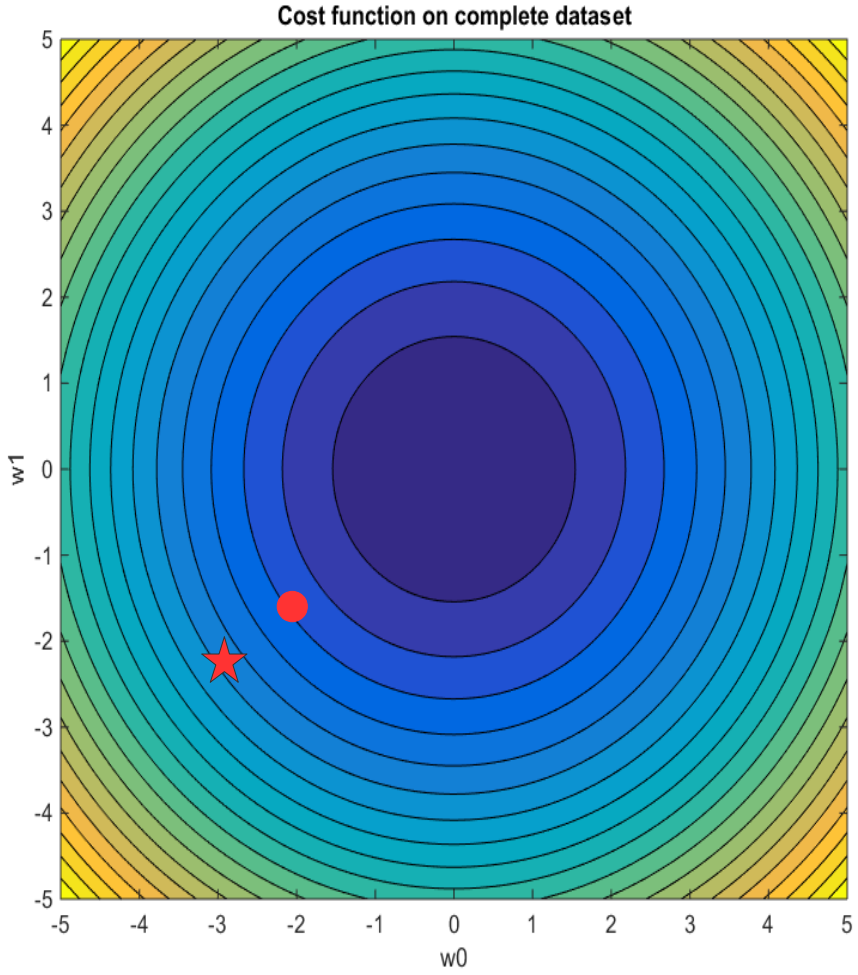
All of this is (mostly) interchangeable !

Gradient vs. Newton method : Choose Accuracy vs. Computation complexity (in D)

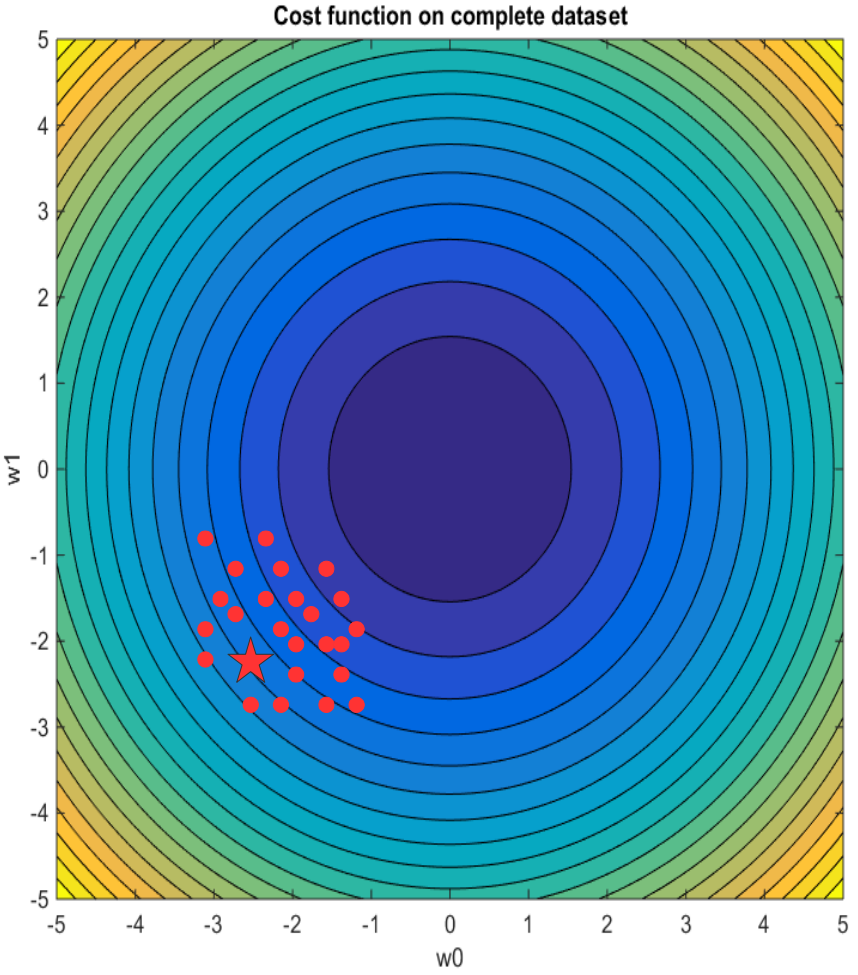
First order	- Use the gradient	- O(ND)
Second order (Newton's method)	- Use the Hessian	- $O(ND^2)$

Full batch vs. Stochastic gradient descent

Other Accuracy - Computation complexity tradeoff

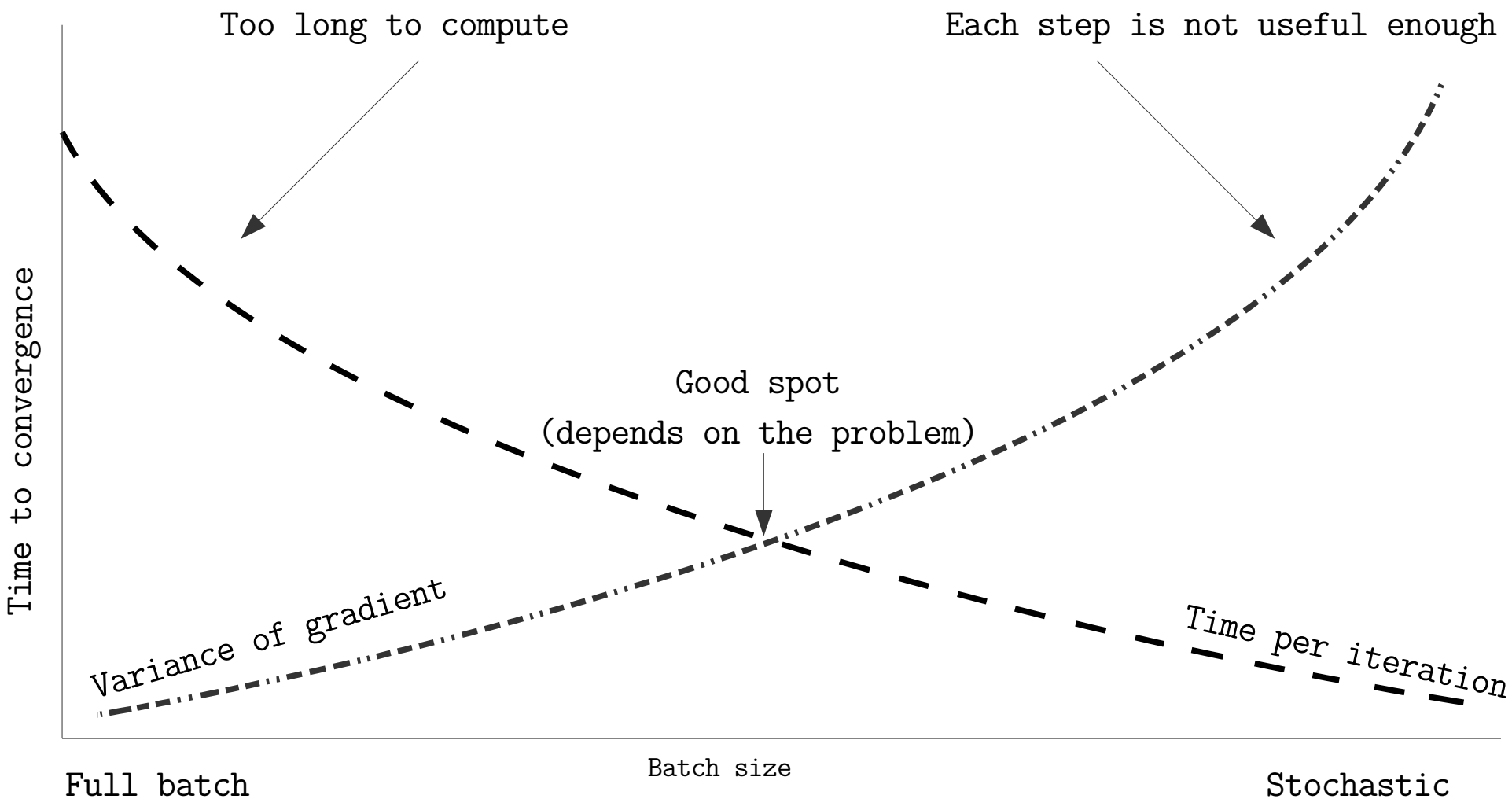


Full batch



Stochastic

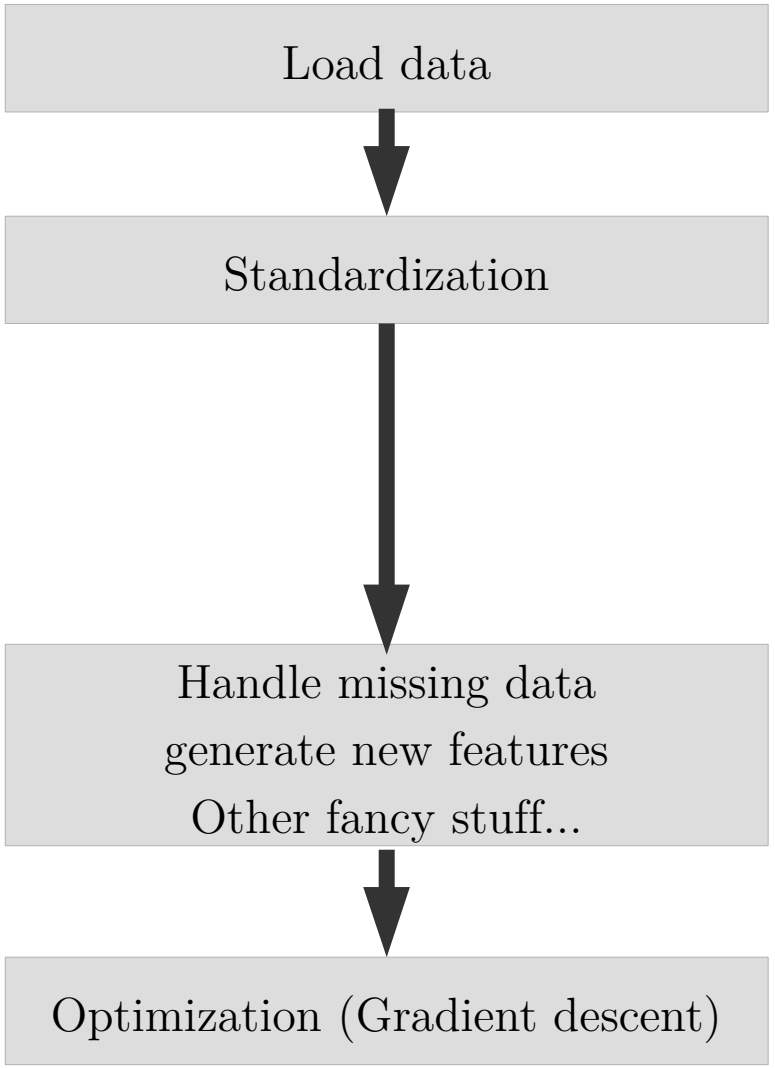
Batch size : Computation time vs. usefulness tradeoff

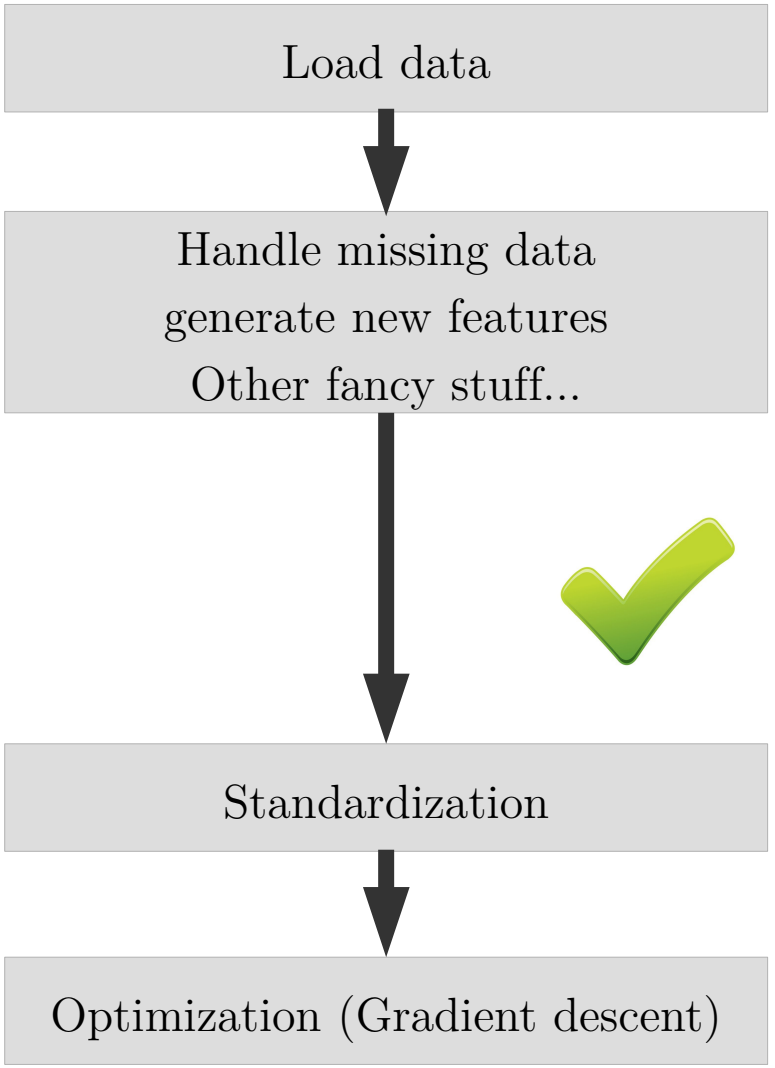


Full batch
- Long to compute
- Very precise measure of gradient

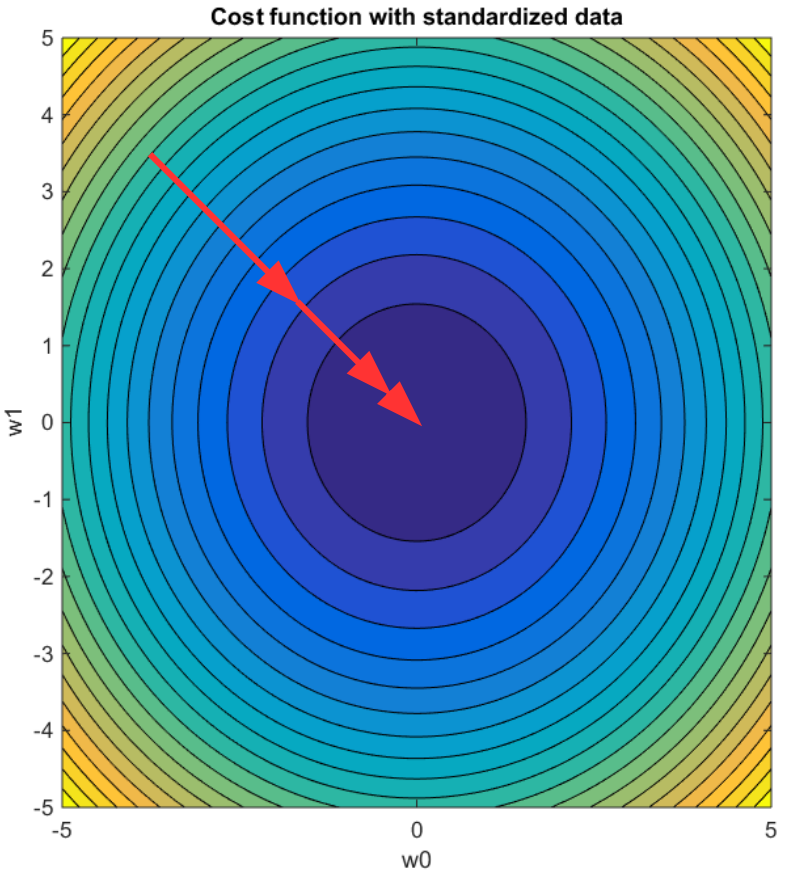
Stochastic
- Quick to compute
- Not precise

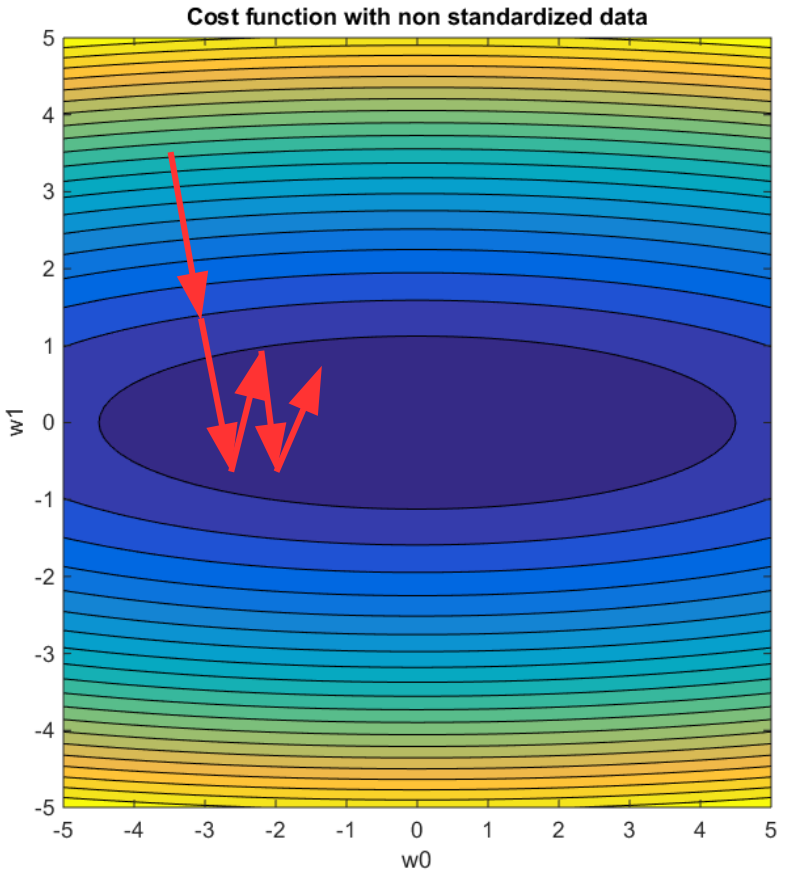
Standardization, when to do it ?



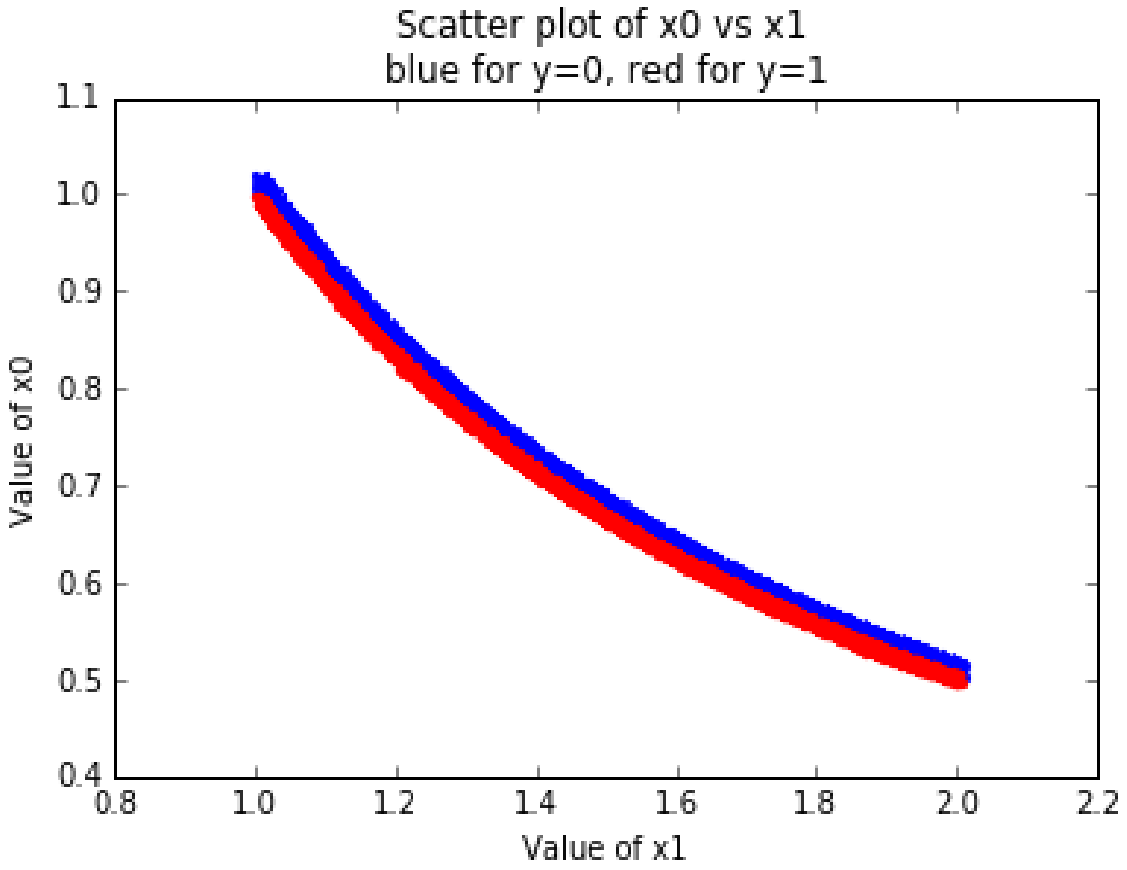


Why ? -> Effect of standardization





Correlation != ~~Causation~~ Useless



Using x0 and x1:

Test accuracy: 0.574 +- 0.002

Correlation (R^2) between x0 and x1:

0.966224848231

Using only x0:

Test accuracy: 0.505 +- 0.005

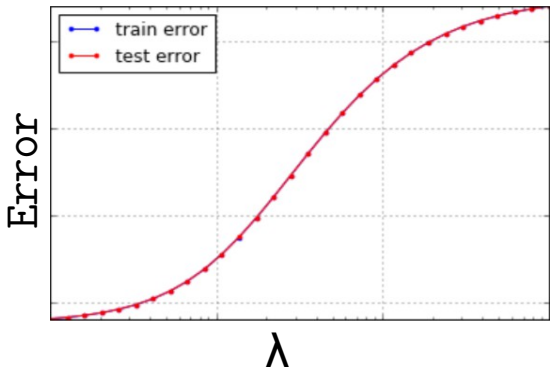
Using only x1:

Test accuracy: 0.508 +- 0.001

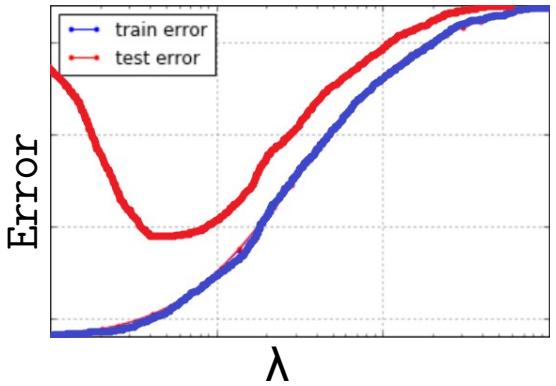
Using x0, x1 and x0*x1

Test accuracy: 0.972 +- 0.008

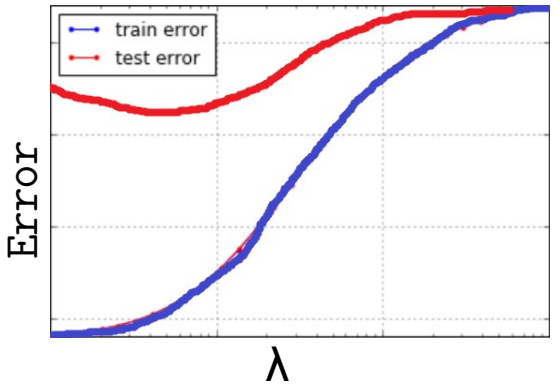
Diagnostic of model



Regularization has no effect on variance, because there is no variance.
Model (features) is either perfect or too simple



High variance, but regularization is helping.
Can improve both ways :
- reducing variance by removing not-that-useful features
- reducing bias by adding new features



High variance, regularization is not helping (enough).
The model is too complex.

Not set in stone - needs to be compared with other models on same problem !

Good luck for project 2 !