# Frank-Wolfe algorithm: effect of dimensionality and condition number on step schedule effectiveness

December 26, 2022

# Frank-Wolfe algorithm

- Problem:

$$\min_{x \in \mathcal{Q}} f(x)$$

- $\mathcal{Q}$ is compact

**Require:** Initial guess $x_0$, tolerance $\delta > 0$
**for** $t = 0, 1, 2, \ldots$ **do**

$$y^k = \arg\min_{y \in \mathcal{Q}} \langle \nabla f(x^k), y \rangle$$

$$x^{k+1} = (1 - \gamma_k) x^k + \gamma_k y^k$$

**end for**
**return** $x^k$

**Goa**l: determine the effect of data dimensionality and matrix condition number (insert definition) on effectiveness of step size schedules

**How to choose step size $\gamma_k$?**

# Step size

- Sublinear

$$\gamma_k = \frac{2}{k+2}$$

- Demyanov-Rubinov

$$\gamma = \min\left\{\frac{\langle -\nabla f(x^k), y^k - x^k\rangle}{L\|y^k - x^k\|^2}, 1\right\}$$

- Backtracking (Pedregosa et al, 2020)

$$\gamma_k = \min\left\{\frac{\langle -\nabla f(x^k), y^k - x^k\rangle}{M_k\|y^k - x^k\|^2}, 1\right\}$$

- Armijo
  Set $h^k = h_0$
  While $f(\theta^k - h^k g^k) > f(\theta^k) - c_1 h^k \langle \nabla f(\theta^k), g^k\rangle$ do $h^k = h^k \rho$

# Backtracking (Pedregosa et al, 2020)

- Step-size update rule

$$\gamma_k = \min\left\{\frac{\langle -\nabla f(x^k), y^k - x^k\rangle}{M_k\|y^k - x^k\|^2}, 1\right\}$$

- $Q_t(\gamma, M_t) \stackrel{\text{def}}{=} f(\boldsymbol{x}_t) - \gamma g_t + \frac{\gamma^2 M_t}{2}\|\boldsymbol{d}_t\|^2$
- $M_k$ update: While $f(\boldsymbol{x}_t + \gamma_t \boldsymbol{d}_t) > Q_t(\gamma_t, M_t)$ do
  $M_t = \tau M_t$

# Setups

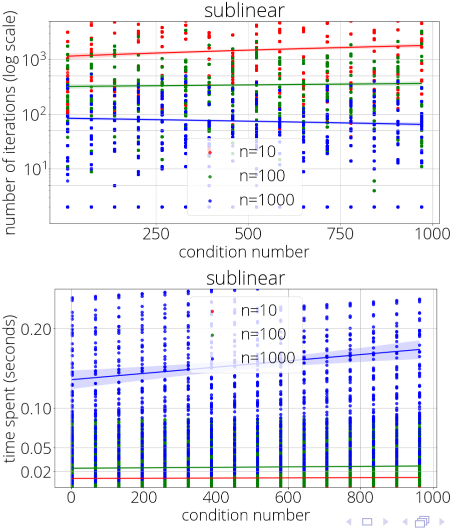
Our primary variables to measure: speed and iterations

- Toy quadratic problem $f(x) = x^T Ax - b^T x$. We control the condition number of $A$.
- LASSO logistic regression, we use 4 binary datasets:
  - Covtype (predicting forest cover type from cartographic variables, 581012 objects, 54 features)
  - Gisette (separation of handwritten numbers 4 and 9, 7000 objects, 5000 features)
  - Madelon (separate artificially created points, 2600 objects, 500 features)
  - RCV1 (predicting newswire articles class, 697641 objects, 47236 features)
- LASSO and Linear regression with synthetic datasets of well- and ill-conditioned regression problems of different dimensionalities

# Condition number experiment set-up

- $\text{minimize}\{f(x) = x^T Ax - b^T x\}$ w.r.t $||x||_2 \leq 1$
- $b \sim \mathcal{N}(0, I_n)$, $A = \text{diag}(a_1, ..., a_n)$, $a_i \sim \text{Uniform}(1, \kappa)$, $A_{00} = 1, A_{nn} = \kappa$, where $\kappa$ is the condition number.
- For stability purpose, we consider $\frac{1}{\text{trace}(A)} A$. This does not impact the condition number
- We use `copt` Frank-Wolfe optimization procedure.
- For each $n$, we run 32 experiments for $\kappa \in [1, 1000]$

# Condition number: sublinear case, num iterations and time spent



# Condition number: DR vs backtracking, iterations num

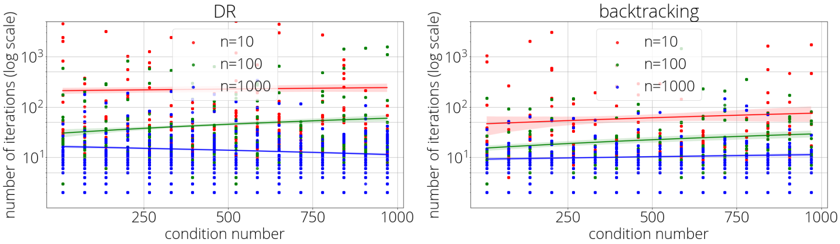
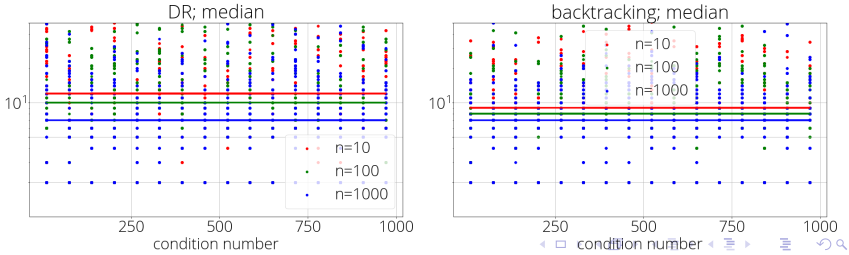
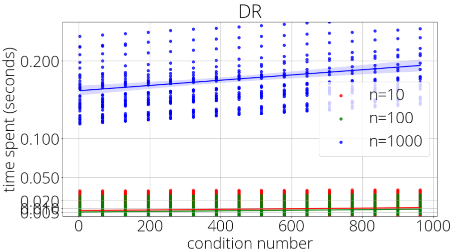
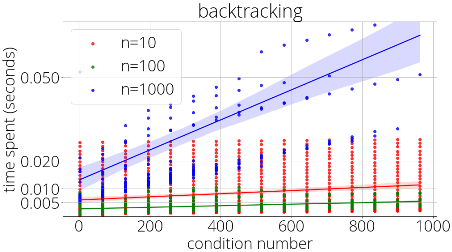



Figure: Regression (Top), median (Bottom)



# Condition number: DR vs backtracking, time spent (sec.)

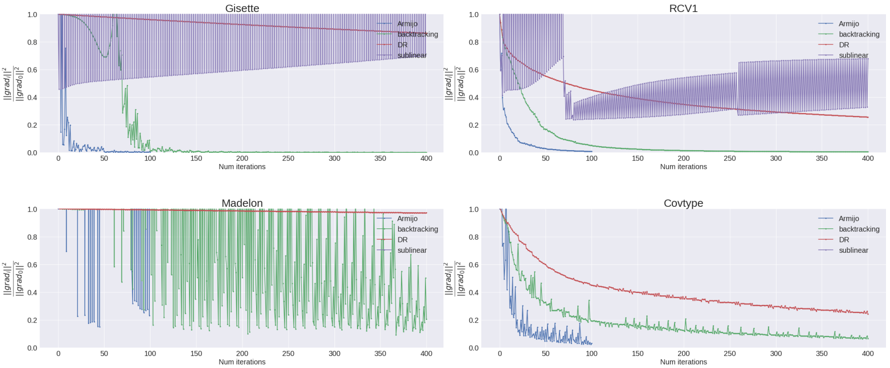


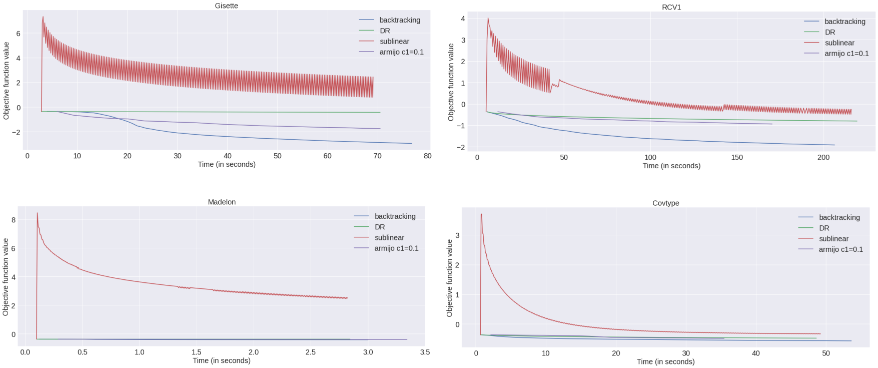
# Toy example: conclusion

- Seems there is no strong dependence on the condition number for any method
- In terms of iterations required, Bactracking is a little bit better (especially for smaller dimensions) than DR
- In terms of time spent, however, Backtracking outperforms DR significantly. This is expected since DR needs to estimate the global constant $L$ (which is harder for larger dimensions)
- Sublinear speed is comparable with backtracking, however it takes more iterations.
- Overall, **backtracking** balances the number of iterations and seconds per iterations.However, the **sublinear** method also performs well and benefits due to its simplicity.

# Logistic regression, convergence

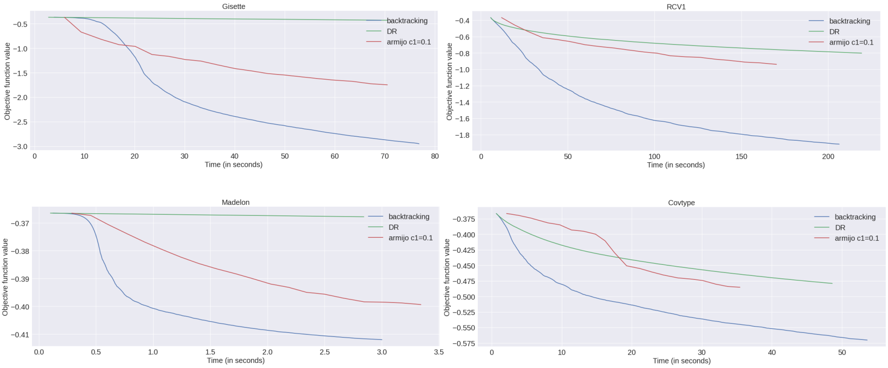


# Logistic regression, convergence 2 (logarithmic scale)



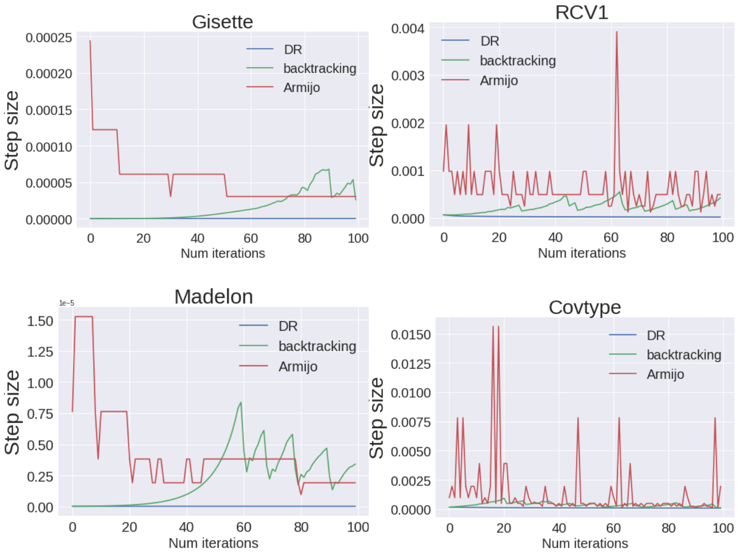

Sublinear step size results in worst convergence on all datasets. It exhibits ’zigzagging’ in objective function value. In all following experiments, Armijo is used with $c_1 = 0.1$.

# Logistic regression, convergence 3 (logarithmic scale)




On all datasets, exact step size significantly outperforms Armijo and DR steps sizes. However, Armijo requires more time per iteration that other methods

# Logistic regression, step size

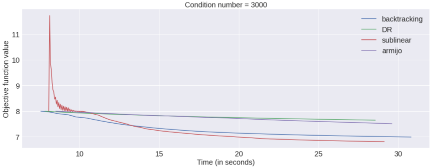


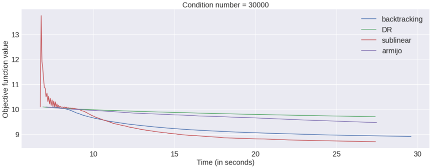
# Logistic regression: conclusion

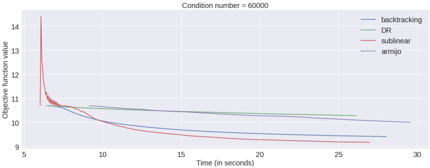
- Armijo is better in terms of number of iterations, but backtracking is better in terms of time
- Sublinear is the worst method in both terms, also unstable
- Sublinear has very big step size, DR has very small, in backtracking it has zigzage form and Armijo can be both small and big

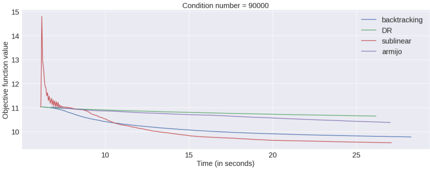
# Linear regression (logarithmic scale)

For probelm $Ax = b$, we generate matrix $A$ with given dimensionality and condition number [1] and construct linear regression problem. We fix dimensionality to 5000, but find no dependence of algorithm convergence on condition number



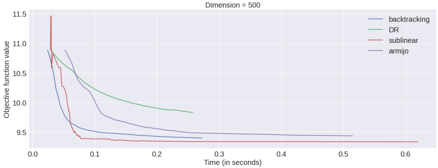



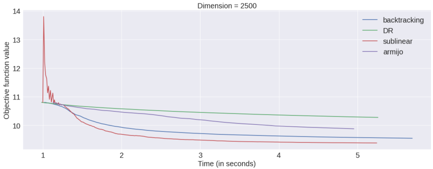



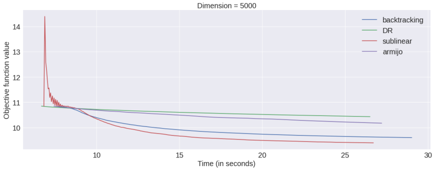

[1] See Bierlaire, M., Toint, P., and Tuyttens, D. (1991). On iterative algorithms for linear ls problems with bound constraints. Linear Algebra and Its Applications, 143, 111-143.

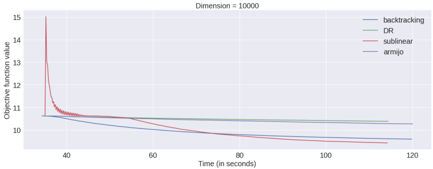
# Linear regression (logarithmic scale)

We fix condition number to 60000.



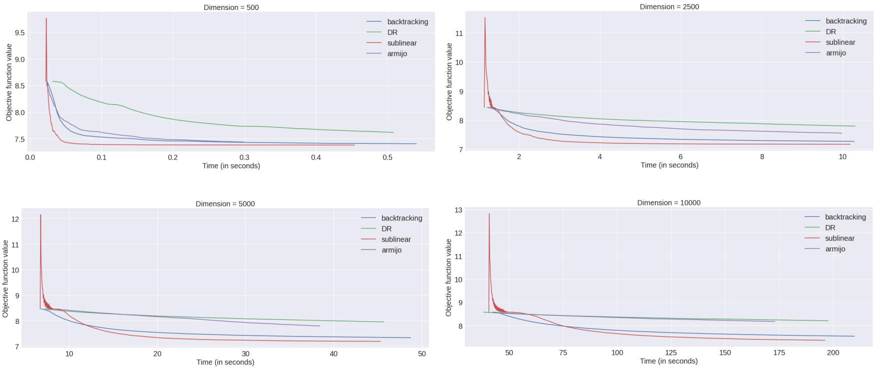



# Linear regression (logarithmic scale)

We fix condition number to 5000.

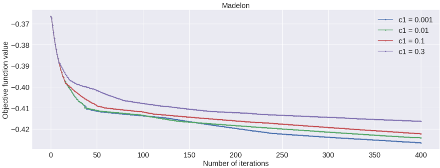


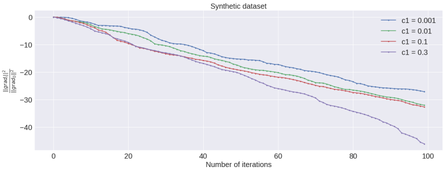

It can be seen that as dimensionality increases, performance of DR and Armijo step schedules becomes more similar, but permofmance of sublinear step degrades

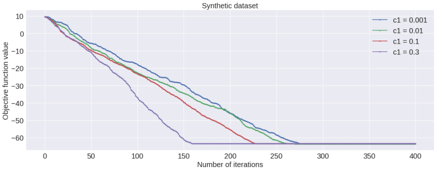
# Linear regression: conclusion

- Condition number in general does not impact convergence for different step sizes
- Sublinear seems to achieve the lowest objective function value
- Armijo is the slowest method
- As dimensionality increases, DR and Armijo have more similar performance

# Influence of Armijo parameters on convergence (logarithmic scale)

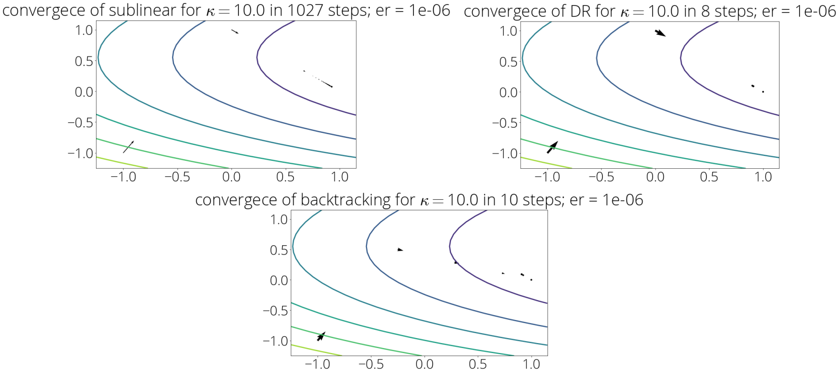







Dependance on $c_1$ coefficient is shown for logarithmic regression on Madelon dataset, and for synthetic LASSO problem with 5000 observations

# Appendix: condition number, 2D example



# Appendix: condition number, 2D example