q7

October 28, 2019

0.1 Question 7 (8 marks): Illustrate the trade-off between the number of iterations and the smoothing parameter μ for gradient descent with Armijo line-search and accelerated gradient with Armijo line-search. Do this by plotting the number of iterations (y-axis) vs magnitude of parameter μ (x-axis in ascending order). Start from a small μ and increase it gradually. Plot the result for both methods in the same plot. Use appropriate legends for the plot.

[ ]:
```
mus = [0.00001, 0.0001, 0.001, 0.01, 0.1, 1.0, 2.0, 5.0, 10.0]

lambda_ = 45
epsilon = 1.0e-2
gamma = 0.3
max_iterations = 200

fvals_arm = []
fvals_prac = []
for mu_val in mus:
    optimized_gd_arm_loop, f_vals_arm_loop =␣
 ↪gradient_descent_arm(x0=noisy_image_vec,
                                                                 ␣
 ↪lambda_=lambda_,
                                                                  mu=mu_val,
                                                                 ␣
 ↪epsilon=epsilon,
                                                                  gamma=gamma,
                                                                 ␣
 ↪max_iterations=max_iterations)

    optimized_gd_acc_prac_loop, f_vals_acc_prac_loop =␣
 ↪accelerated_gd_practical(x0=noisy_image_vec,
                                                                                ␣
 ↪lambda_=lambda_,
                                                                                ␣
 ↪mu=mu_val,
                                                                                ␣
 ↪epsilon=epsilon,
```

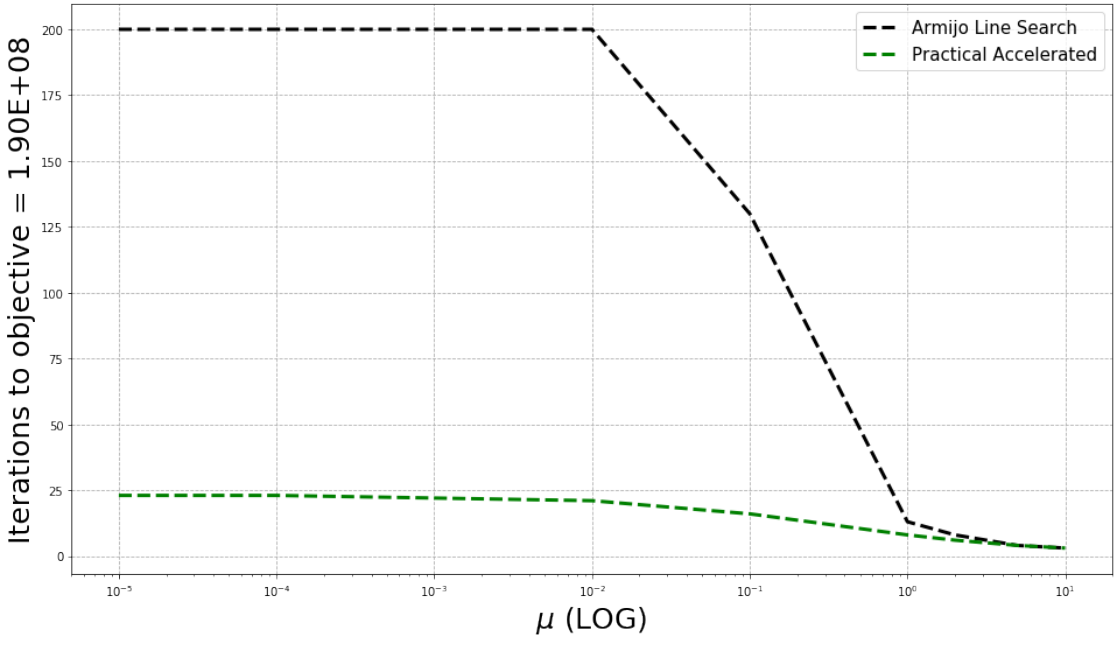
```
↪max_iterations=max_iterations,

↪gamma=gamma)
    fvals_arm.append(f_vals_arm_loop)
    fvals_prac.append(f_vals_acc_prac_loop)
```

[63]:
```
f_cutoff = 1.9e8
iter_arm = [np.where(fval <= f_cutoff)  for fval in fvals_arm]
iter_prac = [np.where(fval <= f_cutoff)  for fval in fvals_prac]

iter_arm = [max_iterations if len(i[0]) == 0 else i[0][0] for i in iter_arm]
iter_prac = [max_iterations if len(i[0]) == 0 else i[0][0] for i in iter_prac]

fig = plt.figure(figsize=(16, 9))
ax = fig.add_subplot(1, 1, 1)
ax.plot(mus, iter_arm, label=("Armijo Line Search"), linewidth=3.0, color␣
↪="black", linestyle='--')
ax.plot(mus, iter_prac, label=("Practical Accelerated"), linewidth=3.0, color␣
↪="green", linestyle='--')
# ax.set_yscale('log')
ax.set_xscale('log')
ax.legend(prop={'size': 15},loc="upper right")
plt.xlabel("$\mu$ (LOG)", fontsize=25)
plt.ylabel(f"Iterations to objective = {f_cutoff:.2E}", fontsize=25)
ax.grid(linestyle='dashed')
# plt.show()
plt.savefig('foo.pdf')
```



[ ]: