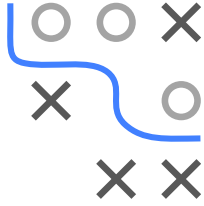
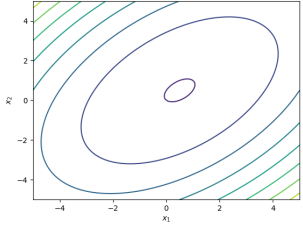
Optimization in Machine Learning



First order methods
GD on quadratic forms



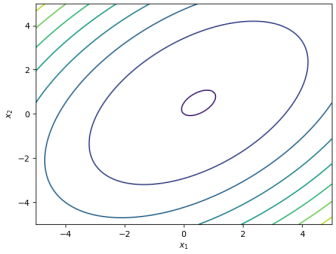
Learning goals

- Eigendecomposition of quadratic forms
- GD steps in eigenspace

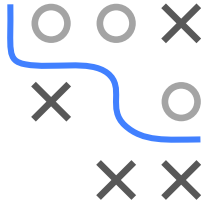
QUADRATIC FORMS & GD

- We consider the quadratic function $q(\boldsymbol{x}) = \boldsymbol{x}^T\mathbf{A}\boldsymbol{x} - \mathbf{b}^T\boldsymbol{x}$
- We assume that the Hessian $\mathbf{H} = 2\mathbf{A}$ has full rank
- We assume q is convex, so $\mathbf{A}$ is psd
- The optimal solution is $\boldsymbol{x}^* = \frac{1}{2}\mathbf{A}^{-1}\mathbf{b}$
- With $\nabla q(\boldsymbol{x}) = 2\mathbf{A}\boldsymbol{x} - \mathbf{b}$, gradient descent iterates as

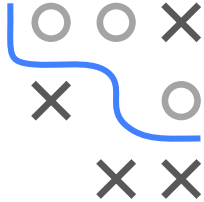
$$\boldsymbol{x}^{[t+1]} = \boldsymbol{x}^{[t]} - \alpha(2\mathbf{A}\boldsymbol{x}^{[t]} - \mathbf{b})$$



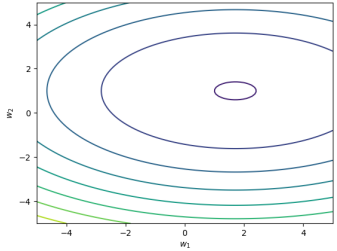
▸ Click for source



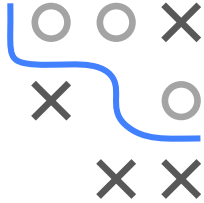
EIGENDECOMPOSITION OF QUADRATIC FORMS



- We want to work in the coordinate system induced by q
- **Recall:** That system is given by the eigenvectors of $\mathbf{H} = 2\mathbf{A}$
- Eigendecomposition: $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$
- $\mathbf{V}$ contains eigenvectors $\boldsymbol{v}_i$ and $\mathbf{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_n)$ stores eigenvalues
- Change of basis: $\boldsymbol{w}^{[t]} = \mathbf{V}^T(\boldsymbol{x}^{[t]} - \boldsymbol{x}^*)$



GD STEPS IN EIGENSPACE



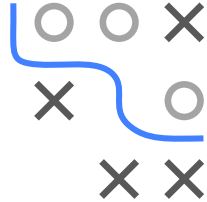
- With $\boldsymbol{w}^{[t]} = \mathbf{V}^T(\boldsymbol{x}^{[t]} - \boldsymbol{x}^*)$, a single GD step starts with

$$\boldsymbol{x}^{[t+1]} = \boldsymbol{x}^{[t]} - \alpha(2\mathbf{A}\boldsymbol{x}^{[t]} - \mathbf{b})$$

- This becomes

$$\boldsymbol{w}^{[t+1]} = \boldsymbol{w}^{[t]} - 2\alpha\boldsymbol{\Lambda}\boldsymbol{w}^{[t]}$$

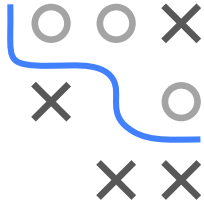
GD STEPS IN EIGENSPACE: PROOF



- A single GD step satisfies $\boldsymbol{x}^{[t+1]} = \boldsymbol{x}^{[t]} - \alpha(2\mathbf{A}\boldsymbol{x}^{[t]} - \mathbf{b})$
- Multiplying with $\mathbf{V}^T$ and subtracting $\boldsymbol{x}^*$ yields

$$\begin{aligned}
\mathbf{V}^T(\boldsymbol{x}^{[t+1]} - \boldsymbol{x}^*) &= \mathbf{V}^T(\boldsymbol{x}^{[t]} - \boldsymbol{x}^*) - \alpha\mathbf{V}^T(2\mathbf{A}\boldsymbol{x}^{[t]} - \mathbf{b}) \\
\boldsymbol{w}^{[t+1]} &= \boldsymbol{w}^{[t]} - \alpha\mathbf{V}^T(2\mathbf{A}(\boldsymbol{x}^{[t]} - \boldsymbol{x}^*) + \underbrace{2\mathbf{A}\boldsymbol{x}^* - \mathbf{b}}_{=0}) \\
&= \boldsymbol{w}^{[t]} - 2\alpha\boldsymbol{\Lambda}\mathbf{V}^T(\boldsymbol{x}^{[t]} - \boldsymbol{x}^*) \\
&= \boldsymbol{w}^{[t]} - 2\alpha\boldsymbol{\Lambda}\boldsymbol{w}^{[t]}
\end{aligned}$$

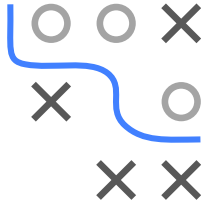
GD STEPS IN EIGENSPACE



Therefore, for each coordinate i:

$$\begin{aligned} w_i^{[t+1]} &= w_i^{[t]} - 2\alpha\lambda_i w_i^{[t]} \\ &= (1 - 2\alpha\lambda_i) w_i^{[t]} \\ &= (1 - 2\alpha\lambda_i)^{t+1} w_i^{[0]}. \end{aligned}$$

GD ERROR IN ORIGINAL SPACE



- Move back to the original space:

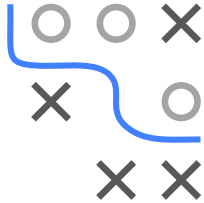
$$\boldsymbol{x}^{[t]} - \boldsymbol{x}^* = \mathbf{V}\boldsymbol{w}^{[t]} = \sum_{i=1}^{d} (1 - 2\alpha\lambda_i)^t w_i^{[0]} \boldsymbol{v}_i$$

- This gives the difference in $\boldsymbol{x}$-space in closed form
- The difference vector is written in the eigenbasis, initial error components $w_i^{[0]}$ decay with rate $1 - 2\alpha\lambda_i$
- Smaller absolute values yield faster decay; values close to 1 slow convergence
- Eigenvectors with larger eigenvalues typically converge fastest, especially early on

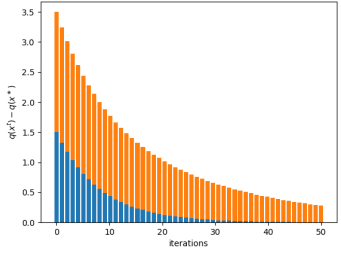
GD REGRET

- We can also analyze the regret in y-space:

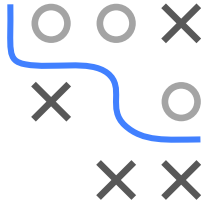
$$q(\boldsymbol{x}^{[t]}) - q(\boldsymbol{x}^*) = \frac{1}{2}\sum_{i=1}^{d}(1-2\alpha\lambda_i)^{2t}\lambda_i(w_i^{[0]})^2$$



- This is straightforward to derive; formulas only become slightly lengthy
- Rates become more similar once the slow modes dominate



OPTIMAL STEP SIZE



- Convergence requires $|1 - 2\alpha\lambda_i| < 1$ for all i
- Therefore $0 < \alpha\lambda_i < 1$
- The slowest component governs the rate:

$$\max_{i=1,\ldots,n} |1 - 2\alpha\lambda_i| = \max(|1 - 2\alpha\lambda_1|, |1 - 2\alpha\lambda_n|)$$

- The optimum occurs at $\alpha = \frac{1}{\lambda_1 + \lambda_n}$
- The optimal rate equals $\frac{\lambda_n/\lambda_1 - 1}{\lambda_n/\lambda_1 + 1}$
- Convergence is driven by the condition number $\kappa = \frac{\lambda_n}{\lambda_1}$
- $\kappa = 1$ is ideal and yields convergence in one step