MLISP: Machine Learning in Signal Processing

Problem set 1

Prof. V. I. Morgenshtern

Problem 1: Gradients and Hessians [Ref: Stanford CS229 class]

Do not use lecture notes handouts or discussion handouts for this exercise. Recall that a matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ is symmetric if $\mathbf{A}^\mathsf{T} = \mathbf{A}$, that is, $A_{ij} = A_{ji}$ for all i, j. Also recall the gradient $\nabla f(\mathbf{x})$ of a function $f : \mathbb{R}^n \to \mathbb{R}$, is the n-vector of partial derivatives

$$\nabla f(\mathbf{x}) = \begin{bmatrix} \frac{\partial}{\partial x_1} f(x) \\ \vdots \\ \frac{\partial}{\partial x_n} f(x) \end{bmatrix}.$$

The Hessian $\nabla^2 f(x)$ of a function $f : \mathbb{R}^n \to \mathbb{R}$ is the $n \times n$ symmetric matrix of twice partial derivatives,

$$\nabla^2 f(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2}{\partial x_1^2} f(x) & \frac{\partial}{\partial x_1 \partial x_2} f(x) & \cdots & \frac{\partial}{\partial x_1 \partial x_n} f(x) \\ \vdots & & & \\ \frac{\partial^2}{\partial x_n \partial x_1} f(x) & \frac{\partial^2}{\partial x_n \partial x_2} f(x) & \cdots & \frac{\partial^2}{\partial x_n^2} f(x) \end{bmatrix}.$$

1. Let $f(x) = \frac{1}{2}\mathbf{x}^\mathsf{T}\mathbf{A}\mathbf{x} + \mathbf{b}^\mathsf{T}\mathbf{x}$, where $\mathbf{A}$ is a symmetric matrix and $\mathbf{b} \in \mathbb{R}^n$ is a vector. What is $\nabla f(\mathbf{x})$?

2. Let $f(x) = g(h(x))$, where $g : \mathbb{R} \to \mathbb{R}$ is differentiable and $h : \mathbb{R}^n \to \mathbb{R}$ is differentiable. What is $\nabla f(\mathbf{x})$?

3. Let $f(x) = \frac{1}{2}\mathbf{x}^\mathsf{T}\mathbf{A}\mathbf{x} + \mathbf{b}^\mathsf{T}\mathbf{x}$, where $\mathbf{A}$ is a symmetric matrix and $\mathbf{b} \in \mathbb{R}^n$ is a vector. What is $\nabla^2 f(\mathbf{x})$?

4. Let $f(x) = g(\mathbf{a}^\mathsf{T}\mathbf{x})$, where $g : \mathbb{R} \to \mathbb{R}$ is continuously differentiable and $\mathbf{a} \in \mathbb{R}^n$ is a vector. What are $\nabla f(\mathbf{x})$ and $\nabla^2 f(\mathbf{x})$?

Problem 2: Positive definite matrices [Ref: Stanford CS229 class]

A matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ is positive semi-definite (PSD), denoted $\mathbf{A} \succeq 0$, if $\mathbf{A} = \mathbf{A}^\mathsf{T}$ and $\mathbf{x}^\mathsf{T}\mathbf{A}\mathbf{x} \geq 0$ for all $\mathbf{x} \in \mathbb{R}^n$. A matrix $\mathbf{A}$ is positive definite, denoted $\mathbf{A} \succ 0$, if $\mathbf{A} = \mathbf{A}^\mathsf{T}$ and $\mathbf{x}^\mathsf{T}\mathbf{A}\mathbf{x} > 0$ for all $\mathbf{x} \neq \mathbf{0}$.

1. Let $\mathbf{z} \in \mathbb{R}^n$ be an n-vector. Show that $\mathbf{A} = \mathbf{z}\mathbf{z}^\mathsf{T}$ is positive semidefinite.

2. Let $\mathbf{z} \in \mathbb{R}^n$ be a non-zero n-vector. Let $\mathbf{A} = \mathbf{z}\mathbf{z}^\mathsf{T}$. What is the null-space of $\mathbf{A}$? What is the rank of $\mathbf{A}$?

3. Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be positive semidefinite and $\mathbf{B} \in \mathbb{R}^{m\times n}$ be arbitrary. Is $\mathbf{B}\mathbf{A}\mathbf{B}^\mathsf{T}$ PSD? If so, prove it. If not, give a counterexample with explicit $\mathbf{A}$, $\mathbf{B}$.

Problem 3: Eigenvectors, eigenvalues, and the spectral theorem [Ref: Stanford CS229 class]

The eigenvalues of an $n \times n$ matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ are the roots of the characteristic polynomial $p_\mathbf{A}(\lambda) = \det(\lambda\mathbf{I} - \mathbf{A})$, which may (in general) be complex. They are also defined as the the values $\lambda \in \mathbb{C}$ for which there exists a vector $\mathbf{x} \in \mathbb{C}^n$ such that $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$. We call such a pair $(\mathbf{x}, \lambda)$ an eigenvector, eigenvalue pair. In this question, we use the notation $\text{diag}(\lambda_1, \ldots, \lambda_n)$ to denote the diagonal matrix with diagonal entries $\lambda_1, \ldots, \lambda_n$, that is,

$$\text{diag}(\lambda_1, \ldots, \lambda_n) = \begin{bmatrix} \lambda_1 & 0 & 0 & & 0 \\ 0 & \lambda_2 & 0 & & 0 \\ \vdots & & & & \\ 0 & 0 & & 0 & \lambda_n \end{bmatrix}.$$

1. Suppose that the matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ is diagonalizable, that is, $\mathbf{A} = \mathbf{T}\Lambda\mathbf{T}^{-1}$ for an invertible matrix $\mathbf{T} \in \mathbb{R}^{n\times n}$, where $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)$ is diagonal. Use the notation $\mathbf{t}_i$ for the columns of $\mathbf{T}$, so that $\mathbf{T} = [\mathbf{t}_1 \cdots \mathbf{t}_n]$, where $\mathbf{t}_i \in \mathbb{R}^n$. Show that $\mathbf{A}\mathbf{t}_i = \lambda_i\mathbf{t}_i$, so that the eigenvalues/eigenvector pairs of $\mathbf{A}$ are $(\mathbf{t}_i, \lambda_i)$.

A matrix $\mathbf{U} \in \mathbb{R}^{n\times n}$ is orthogonal if $\mathbf{U}^\mathsf{T}\mathbf{U} = \mathbf{I}$. The spectral theorem, perhaps one of the most important theorems in linear algebra, states that if $\mathbf{A} \in \mathbb{R}^{n\times n}$ is symmetric, that is, $\mathbf{A} = \mathbf{A}^\mathsf{T}$, then $\mathbf{A}$ is diagonalizable by a real orthogonal matrix. That is, there are a diagonal matrix $\Lambda \in \mathbb{R}^{n\times n}$ and orthogonal matrix $\mathbf{U} \in \mathbb{R}^{n\times n}$ such that $\mathbf{U}^\mathsf{T}\mathbf{A}\mathbf{U} = \Lambda$, or, equivalently, $\mathbf{A} = \mathbf{U}\Lambda\mathbf{U}^\mathsf{T}$. Let $\lambda_i = \lambda_i(\mathbf{A})$ denote the ith eigenvalue of $\mathbf{A}$.

2. Let $\mathbf{A}$ be symmetric. Show that if $\mathbf{U} = [\mathbf{u}_1 \cdots \mathbf{u}_n]$ is orthogonal, where $\mathbf{u}_i \in \mathbb{R}^n$ and $\mathbf{A} = \mathbf{U}\Lambda\mathbf{U}^\mathsf{T}$, then $\mathbf{u}_i$ is an eigenvector of $\mathbf{A}$ and $\mathbf{A}\mathbf{u}_i = \lambda_i\mathbf{u}_i$, where $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)$.

3. Show that if $\mathbf{A}$ is positive semi-definite, then $\lambda_i(\mathbf{A}) \geq 0$ for each i.

Problem 4: Python and Jupyter notebooks

Python is quickly becoming a standard language for machine learning, data science, and artificial intelligence. Many of the exercises for this class are computer-based. Even though you are free to use the language of your choice to do these exercises, it is highly recommended that you use Python and work in Jupyter notebook. Jupyter notebook is a powerful interactive web-based interface for Python. In this exercise, assuming that you know Python, we ask to to install Jupyter notebook and familiarize yourself with it. Here is a tutorial you can use: `https://www.dataquest.io/blog/jupyter-notebook-tutorial`. In the notebook `read_data.ipynb` you will find a basic snippet of Python code to read the data necessary in the exercise below.

Problem 5: Linear regression and gradient descent

The file `house_prices.txt` contains a data set of house prices: the first column is the house size in square feet, the second column is the number of bedrooms, the third column is the price in USD.

1. Plot house prices vs. house sizes as a scatter plot.
2. Next, fit the linear regression to these data points. You should not use a software package to do this. Instead, follow the instructions:
   - Consider the linear model $h_{\boldsymbol{\theta}}(\mathbf{x}) = \theta_0 x_0 + \theta_1 x_1 = \boldsymbol{\theta}^\mathsf{T}\mathbf{x}$, where x_1 is the house size in the first column of `house_prices.txt`, $x_0 = 1$ by convention, $\mathbf{x} = [x_0, x_1]^\mathsf{T}$, and $\boldsymbol{\theta} = [\theta_0, \theta_1]^\mathsf{T}$. Define the cost function on the dataset:

     $$J(\boldsymbol{\theta}) = \frac{1}{2n}\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) - y^{(i)}\right)^2. \tag{1}$$

     Above, n is the number of rows in `house_prices.txt`, $y^{(i)}$ are the house prices in the third column of the file, $\mathbf{x}^{(i)} = [1\ x_1^{(i)}]^\mathsf{T}$ and $x_1^{(i)}$ is the house size in the first column and ith row of the file. Plot $J(\boldsymbol{\theta})$ as a function of $\boldsymbol{\theta}$ using the contour plot. You should see a figure similar to 'Contour plot of $J(\cdot)$' in the lecture notes of Lecture 1.
   - Start with some initial value $\boldsymbol{\theta}_0$ and run the steps of the gradient descent algorithm as explained in Lecture 1. Plot the location of each new $\boldsymbol{\theta}_j$ on the scatter plot you made above. *[Hint: make your gradient descent algorithm easily adjustable to different amounts of x and theta values, so you can reuse it in later problems]*
   - Make changes to the learning rate of the gradient descent algorithm. Observe how the path of the algorithm changes. Make sure that the algorithm converges to the true minimum of the function $J(\boldsymbol{\theta})$. *[Hint: use different alphas for the two x dimensions, to get a better convergence rate. To calculate the convergence rate divide the new cost by the old cost after each iteration]*
   - Use the closed form solution for $\boldsymbol{\theta}$:

     $$\hat{\boldsymbol{\theta}} = (\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y}. \tag{2}$$

     Above, $\mathbf{y} = [y^{(1)}, \cdots y^{(n)}]^\mathsf{T}$ and $\mathbf{X}$ is the data matrix whose rows are $\mathbf{x}^{(i)}$. Is the the same point that you have found above using gradient descent?
3. Consider the function $J(\theta_0, \theta_1) = \theta_0^2\cos(\theta_1)$ on the interval $[-1, 1] \times [-2\pi, 2\pi]$.
   - Plot the contour plot of this function.
   - Use gradient descent as above to find the minimum of this function. Plot the location of each new $\boldsymbol{\theta}_j$ on the scatter plot you made above.
   - Experiment with the learning rate and the initial point. Do you always find the same minimum?

Problem 6: High-dimensional regression (exam practice)

We have the following regression model

$$\mathtt{y} = \mathbf{X}\boldsymbol{\theta} + \mathtt{m}$$

where $\mathtt{y} \in \mathbb{R}^n$, $\mathbf{X} \in \mathbb{R}^{n \times p}$, $\boldsymbol{\theta} \in \mathbb{R}^p$, $\mathtt{m} \in \mathbb{R}^n$, $\mathtt{m}$ is zero-mean and $\mathbb{V}\mathrm{ar}\left[\mathtt{m}\right] = \sigma^2 \mathbf{I}$.

1. Calculate $\hat{\boldsymbol{\theta}}$, the least squares estimate of $\boldsymbol{\theta}$. Please do the derivation.
2. What assumption on $\mathbf{X} \in \mathbb{R}^{n \times p}$ do you need to invert $\mathbf{X}^\top \mathbf{X}$? What does this assumption mean for the relation between n and p?
3. Calculate the expected value of $\hat{\boldsymbol{\theta}}$.