Gradients & Regression
A system of linear equations

Find \( x_1, x_2, x_3 \) such that

\[
\begin{align*}
    a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \\
    a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2 \\
    a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3
\end{align*}
\]

Can also be written as

\[
\begin{pmatrix}
    a_{11} & a_{12} & a_{13} \\
    a_{21} & a_{22} & a_{23} \\
    a_{31} & a_{32} & a_{33}
\end{pmatrix}
\begin{pmatrix}
    x_1 \\
    x_2 \\
    x_3
\end{pmatrix}
=
\begin{pmatrix}
    b_1 \\
    b_2 \\
    b_3
\end{pmatrix}
\]

Or as: \( Ax = b \)
To solve, invert the matrix

\[ Ax = b \iff x = A^{-1}b \]

- Inverse might not exist
- System can be
  - under-determined (infinite set of solutions)
  - Or over determined (no solution).
Approximately solving over-determined systems

There is no $x$ that satisfies $Ax = b$

Instead, find $x$ that minimizes $\|Ax - b\|_2$

• How to find the minimum?
• In one dimensional problem: set derivative to zero.
• In multi-dimensional case, set gradient to zero.
A function of two variables

The surface defined by $f(x,y) = 9 - x^2 - y^2$. 
Restricting the function to the variable $x$
Computing the partial derivative \( \frac{\partial}{\partial x} \)
Gradient = the partial derivative wrt all coordinates
Computing the gradient symbolically

\[ \nabla f = \left\langle \frac{\partial}{\partial x} f, \frac{\partial}{\partial y} f \right\rangle \]

example: \[ f(x, y) = 9 - x^2 - y^2 \]
\[ \nabla f = \left\langle -2x, -2y \right\rangle \]
Setting the gradient to zero we find that the maximum is at \( \langle x, y \rangle = \langle 0, 0 \rangle \)
Exactly minimizing square error

There is no \( x \) that satisfies \( Ax = b \)

Instead, find \( x \) that minimizes \( \| Ax - b \|_2^2 \)

Find \( x \) such that

\[
\nabla_x \| Ax - b \|_2^2 = 0
\]

\[
\nabla_x \| Ax - b \|_2^2 = 2 A^T (Ax - b) = 0
\]

\[
\x = (A^T A)^{-1} A^T b
\]

Pseudo-inverse of \( A \)
When the number of examples is large

- The size of the matrix $A$
  is number of variables $X$ number of examples
- Exact solution is not practical.
- The alternative: stochastic gradient descent.
Review: the gradient

$f : \mathbb{R}^d \rightarrow \mathbb{R}$ is a smooth function from $\mathbb{R}^d$ to $\mathbb{R}$

The gradient of $f$ at the point $\bar{x}$, denoted $\nabla f(\bar{x})$

is a vector pointing in the direction of steepest ascend (increase) of $f$

The gradient $\nabla f(\bar{x})$ can be calculated using partial derivatives:

$$\nabla f(\bar{x}) = \left\langle \frac{\partial f(\bar{x})}{\partial x_1}, \frac{\partial f(\bar{x})}{\partial x_2}, \ldots, \frac{\partial f(\bar{x})}{\partial x_d} \right\rangle$$
Optimization by Gradient Ascent

- Start at a randomly chosen starting point
  - Take a small step in the direction of the gradient
  - Repeat
- Converges to a local maximum (gradient zero).
- Which local maximum depends on starting point
Deterministic & Stochastic Gradient Descent

Find $\mathbf{x}$ that minimizes $\| \mathbf{A}\mathbf{x} - \mathbf{b} \|_2^2 = \sum_{i=1}^{N} (a_i \mathbf{x} - b_i)^2$

$$\nabla_\mathbf{x} \| \mathbf{A}\mathbf{x} - \mathbf{b} \|_2^2 = \nabla_\mathbf{x} \sum_{i=1}^{N} (a_i \mathbf{x} - b_i)^2 = \sum_{i=1}^{N} 2(a_i \mathbf{x} - b_i) a_i$$

• Taking a step in direction opposite of gradient moves $\mathbf{x}$ towards the minimum.
• **Deterministic gradient Descent:** sum over all examples and then take a step.
• **Stochastic Gradient Descent:** take a small step after each example.
• **Mini-Batch:** Take a step after summing $M>1$ examples.
LinearRegressionWithSGD

\[ x_{t+1} = x_t - \eta \sum_{i=1}^{M} (a_{t,i} \cdot x_t - b_t) a_{t,i} \]

LinearRegressionWithSGD(data, it, s, miniB, init)

- **data** – The training data, an RDD of LabeledPoint.
- **iterations** – The number of iterations (default: 100).
- **step** – The step parameter used in SGD (default: 1.0).
- **miniBatchFraction** – Fraction of data to be used for each SGD iteration (default: 1.0).
- **initialWeights** – The initial weights (default: None).
Learning rate and initial weights

• SGD is guaranteed to converge to a local minimum, if the learning rate (step) is sufficiently small.
• If step size too large – SGD can diverge.
• If step size too small – convergence will take many iterations.
• Initial weights can help start the process close to the minimum.
Why Minibatch?

- Updating separately in each executor will cause the estimate of $x$ for different partitions to diverge.
- Alternatively, communicating each update to all executors creates a communication and synchronization bottleneck.
- **Minibatch**: each partition calculates a sum using a fraction of it’s partition. The sums are combined and all executors receive the same updated $x$
- Smaller mini-batches – faster convergence, but more communication.
Mini-Batch SGD

Sum and update state vector
Learning rate matters!

\[ \eta_t = t, \text{ it is too big} \]

too small \( \eta_t \), after 100 iterations
Stochastic gradient descent

Batch gradient descent

- Data set: set-1 (100 examples, 2 gaussians)
- Network: 1 linear unit, 2 inputs, 1 output, 2 weights, 1 bias
- Learning rate: $\eta = 2.5$
- Hessian largest eigenvalue: $\lambda_{\max} = 0.84$
- Maximum admissible learning rate: $\eta_{\max} = 2.38$

- Log MSE (J/\tau)
- Epochs

Stochastic gradient descent

- Data set: set-1 (100 examples, 2 gaussians)
- Network: 1 linear unit, 2 inputs, 1 output, 2 weights, 1 bias
- Learning rate: $\eta = 0.2$
- Hessian largest eigenvalue: $\lambda_{\max} = 0.84$
- Maximum admissible learning rate (for batch): $\eta_{\max} = 2.38$

- Log MSE (J/\tau)
- Epochs

Training set and Test set

• We are usually interested in finding models that fit well **unseen** data.

• To evaluate the effectiveness of the learning algorithm we separate the data randomly into two parts:
  – Training set: used to find best model
  – Test set: used to see if model generalizes well.
Regularization

- When the data is high dimensional and noisy, decreasing the training error too much will often cause the test error to increase.
- This is called overfitting.
- One way to avoid overfitting is to “regularize” the trained model.

Find \( \mathbf{x} \) that minimizes \( \| \mathbf{Ax} - \mathbf{b} \|_2^2 + \lambda \| \mathbf{x} \| \)

L2: Ridge Regression: \( \| \mathbf{x} \|_2^2 = \sum_i x_i^2 \)

L1: Lasso: \( \| \mathbf{x} \|_1 = \sum_i |x_i| \)
Additional Parameters for LinearRegressionWithSGD

- **regParam** – The regularizer parameter (default: 0.0).
- **regType** –
  The type of regularizer used for training our model.
  - **Allowed values**:
    - “l1” for using L1 regularization (lasso),
    - “l2” for using L2 regularization (ridge),
    - None for no regularization
      (default: None)
- **intercept** – Boolean parameter which indicates the use or not of the augmented representation for training data (i.e. whether bias features are activated or not, default: False).
- **validateData** – Boolean parameter which indicates if the algorithm should validate data before training. (default: True)
- **convergenceTol** – A condition which decides iteration termination. (default: 0.001)