VMLS Cheatsheet[1-9] - meanmachin3

1 Vectors

Vectors

•An ordered finite list of numbers.
•Block or stacked vectors($a = [b, c, d]$), Subvectors ($a_{r:s} = (a_r, ..., a_s)$), Zero vectors (all elements equal to zero), Unit vectors(($e_i = 1$)), Ones vector(1_n) & Sparsity($nnz(x)$)

Vector addition

•*Commutative*: $a + b = b + a$
•*Associative*: $(a + b) + c = a + (b + c)$
•$a + 0 = 0 + a = a$
•$a - a = 0$

1.1 Scalar-vector multiplication

$(-2)(1, 9, 6) = (-2, -18, -12)$
•*Commutative*: $\alpha a = a\alpha$
•*Left-distributive*: $(\beta + \gamma)a = \beta a + \gamma a$
•*Right-distributive*: $a(\beta + \gamma) = a\beta + a\gamma$

Linear combinations: $\beta_1 a_1 + ... + \beta_m a_m$
•With Unit vectors: $b = b_1 e_1 + ... + b_n e_n$
•If $\beta_1 + ... + \beta_m = 1$, linear combination is said to be *affine combination*

1.2 Inner product

$a^T b = a_1 b_1 + a_2 b_2 + ... + a_n b_n$ **Properties**:
•*Commutativity*:$a^T b = b^T a$
•*Scalar multiplication Associativity*: $(\gamma a)^T b = \gamma(a^T b)$
•Vector addition Distributivity: $(a + b)^T c = a^T c + b^T c$.

General examples:

•Unit vector: $e_i^T a = a_i$
•Sum: $\mathbf{1}^T a = a^1 + ... + a^n$
•Average: $(\mathbf{1}/n)^T a = (a^1 + ... + a^n)/n$
•Sum of squares: $a^T a = a_1^2 + ... + a_n^2$
•Selective sum: If $b_i = 1 or 0$, $b^T a$ is the sum of elements for which $b_i = 1$,

Block vectors

$a^T b = a_1^T b_1 + ... + a_k^T b_k$

1.3 Complexity of vector computations

Space: 8n bytes
Complexity of vector operations: $x^T y = 2n - 1$ flops (n scalar multiplications and $n - 1$ scalar additions)
Complexity of sparse vector operations: If x is sparse, then computing ax requires $\mathbf{nnz}(x)$ flops, If x and y are sparse, computing x + y requires no more than min$\mathbf{nnz}(x)$, $\mathbf{nnz}(y)$. computing $x_T y$ requires no more than 2 min$\mathbf{nnz}(x)$, $\mathbf{nnz}(y)$ flops

2 Linear functions

2.1 Linear functions

$f : R^n \to R$ means f is a function mapping n-vectors to numbers
Superposition & linearity: $f(\alpha x + \beta y) = \alpha f(x) + \beta f(y)$
•$f(\alpha_1 x_1 + ... + \alpha_k x_k) = \alpha_1 f(x_1) + ... + \alpha_k f(x_k)$
A function that satisfies superposition is called *linear*

Linear function satisfies

•*Homogeneity*: For any n-vector x and any scalar α, $f(\alpha x) = \alpha f(x)$
•*Additivity*: For any n-vectors x and y, $f(x + y) = f(x) + f(y)$

Affine functions $f : R_n \to R$ is affine if and only if it can be expressed as $f(x) = a^T x + b$ for some n-vector a and scalar b, which is sometimes called the *offset* •Any *affine* scalar-valued function satisfies the following variation on the super-position property: $f(\alpha x + \beta y) = \alpha f(x) + \beta f(y)$, where $\alpha + \beta = 1$

2.2 Taylor approximation

The (first-order) Taylor approximation of f near (or at) the point z:
$\hat{f}(x) = f(z) + \frac{\partial f}{\partial x_1}(z)(x_1 - z_1) + ... + \frac{\partial f}{\partial x_n}(z)(x_n - z_n)$
Alternatively, $\hat{f}(x) = f(z) + \nabla f(z)^T(x - z)$

2.3 Regression model

Regression model is (the affine function of x) $\hat{y} = x^T \beta + v$

3 Norm and distance

3.1 Norm

Euclidean norm (or just norm) is
$||x|| = \sqrt{x_1^2 + x_2^2 + ... + x_n^2} = \sqrt{x^T x}$

Properties

•homogeneity: $||\beta x|| = |\beta| ||x||$
•triangle inequality: $||x + y|| \le ||x|| + ||y||$
•non negativity: $||x|| \ge 0$
•definiteness: $||x|| = 0$ only if x = 0
positive definiteness = non negativity + definiteness

$\mathbf{rms}(x) = \sqrt{\frac{x_1^2 + ... + x_n^2}{n}} = \frac{||x||}{\sqrt{n}}$

•*Norm of a sum*:
$||a+b||^2 = (x+y)^T(x+y) = ||x||^2 + 2x^T y + ||b||^2$

Norm of block vectors $||(a, b, c)|| = \sqrt{||a||^2 + ||b||^2 + ||c||^2} = ||(||a||, ||b||, ||c||)||$

Chebyshev inequality k of its entries satisfy $|x_i| \ge a$,
then $\frac{k}{n} \le (\frac{\mathbf{rms}(x)}{a})^2$

3.2 Distance

$\mathbf{dist}(a, b) = ||a - b||$
Triangle Inequality: $||a - c||^2 = ||(a - b) + (b - c)|| \le ||a - b|| + ||b - c||$
z_j is the nearest neighbor of x if $||x - z_j|| \le ||x - z_i||, i = 1, .., m$

3.3 Standard Deviation

de-meaned vector: $\tilde{x} = x - \mathbf{avg}(x)\mathbf{1}$
standard deviation:
$\mathbf{std}(x) = \mathbf{rms}(\tilde{x}) = \frac{||x - (\mathbf{1}^T x/n)\mathbf{1}||}{\sqrt{n}}$
$\mathbf{rms}(x)^2 = \mathbf{avg}(x)^2 + \mathbf{std}(x)^2$
By Chebyshev inequality, $|x_i - \mathbf{avg}(x)| \ge \alpha\mathbf{std}(x)$ then $k/n \le (std(x)/a)^2$. (This inequality is only interesting for $a > std(x)$)
Cauchy–Schwarz inequality: $|a^T b| \le ||a|| ||b||$

3.4 Angle

angle between two nonzero vectors a, b defined as
$\angle(a, b) = arccos(\frac{a^T b}{||a|| ||b||})$
$a^T b = ||a|| ||b|| cos(\angle(a, b))$

Classification of angles

$\theta = \pi/2$: $a \perp b$
$\theta = 0$: $a^T b = ||a|| ||b||$
$\theta = \pi = 180°$: $a^T b = -||a|| ||b||$
$\theta \le \pi/2 = 90° = a^T b \ge 0$
$\theta \ge \pi/2 = 90° = a^T b \le 0$

Correlation Coeficient (ρ) $\rho = \frac{\tilde{a}^T \tilde{b}}{||\tilde{a}|| ||\tilde{b}||}$

With $u = \tilde{a}/\mathbf{std}(a)$ & $u = \tilde{b}/\mathbf{std}(b)$,
$\rho = u^T v/n$ where $||u|| = ||v|| = n$

$std(a + b) = \sqrt{std(a)^2 2 + 2\rho std(a) std(b) + std(b)^2}$

Properties of standard deviation

•$\mathbf{std}(x + a1) = \mathbf{std}(x)$
•$\mathbf{std}(ax) = |a| std(x)$

Standardization $z = \frac{1}{\mathbf{std}(x)}(x - \mathbf{avg}(x)\mathbf{1})$

3.5 Complexity

•*norm*: 2n
•***rms***: 2n
•***dist***(a,b): 3n
•$\angle(a, b)$: 6n

4 Clustering

4.1 Clustering

4.2 A clustering Objective

$G_j \subset \{i | c_i = j\}$ where G_j is set of all indices i for which $c_i = j$
•Group representatives: n-vectors $z_1, .., z_k$
•Clustering objective is
$J^{clust} = \frac{1}{N}\sum_{i=1}^{N} ||x_i - Z_{c_i}||^2$
•mean square distance from vectors to associated representative
•goal: choose clustering c_i and representatives z_j to minimize $J_c lust$

4.3 The k-means algorithm

---

given $x_1, .., x_N \in \mathbf{R}^n$ and $z_1, ..., z_k \in \mathbf{R}^n$
repeat
– *Update partition*: assign i to G_j, $j = argmin_{j'} ||x_i - z_{j'}||_2$
– *Update centroids*: $Z_j = \frac{1}{|G_j|}\sum_{i \in G_j} x_i$
until $z1, ..., zk$ stop changing

---

5 Linear Independence

$(a_1, ..., a_k)$ is linearly dependent if $\beta_1 a_1 + ... + \beta_k a_k = 0$, for some $\beta_1, ... \beta_k$,that are not all zero

5.1 Linear Independence

$(a_1, ..., a_k)$ is linearly independent if $\beta_1 a_1 + ... + \beta_k a_k = 0$ & $\beta_1 = ... = \beta_k = 0$
•Adding vector to linearly dependent makes new vector linearly dependent
•Removing vector from linearly independent makes new vector linearly independent

5.2 Basis

basis: A collection of n linearly independent(maximum possible size) n-vectors

Independence-dimension inequality

•*a linearly independent set of n-vectors can have at most n elements*
•*any set of n + 1 or more n-vectors is linearly dependent*

5.3 Orthonomal Vectors

$a_1, ..., a_k$ are (mutually) *orthogonal* if $a_i \perp a_j$ for i != j
They are *normalized* if $||a_i|| = 1$ for i=1,..,k
•*orthonormal* if *orthogonal* & *normalized*
•can be expressed using inner products
$a_i^T a_j = \begin{cases} 1, & \text{if } i = j \\ 0, & i \ne j \end{cases}$
•orthonormal sets of vectors are linearly independent
•$a_1, ..., a_n$ is an orthonormal basis, we have for any n-vector $x = (a_1^T x)a_1 + ... + (a_n^T x)a_n$

5.4 Gram–Schmidt(orthogonalization)

An algorithm to check if $a_1, ..., a_k$ are linearly independent

---

given n-vectors $a_1, ... a_n$
for i = 1,..,k
1.Orthogonalization:
$\tilde{q}_i = a_i - (q_1^T a_i)q_1 - ... - (q_{i-1}^T a_i)q_{i-1}$
2. Test for linear dependence:
if $\tilde{q} = 0$, quit
3.Normalization: $q_i = \tilde{q}_i/||\tilde{q}_i||$

•if G–S does not stop early (in step 2), $a_1, ..., a_k$ are linearly independent
•if G–S stops early in iteration $i = j$, then a_j is a linear combination of $a_1, .., a_{j-1}$ (so $a_1, .., a_k$ are linearly dependent)

Complexity: $2nk^2$

6 Matrices

6.1 Matrices

The set of real m × n matrices is denoted $R^{m \times n}$

6.2 Zero and identity matrices

•*Zero*: All elements equals 0.
•*Identity*: All elements equals 0 and diagonal element equals 1.
•*Sparse*: If many entries are 0
•*Diagonal*: off-diagonal entries are zero
•*Triangular*: *upper triangular* if $A_{ij} = 0$ for $i > j$, and it is *lower triangular* if $A_{ij} = 0$ for $i < j$
•*Adjacency Matrix*: For, $R = (1, 2), (1, 3), (2, 1), (2, 4), (3, 4), (4, 1)$
$A_{ij} = \begin{cases} 1, & (i, j) \in R \\ 0, & (i, j) \notin R \end{cases}$
A relation R on $1, ..., n$ is represented by the n×n matrix A with $A_{ij} = 1$, if there exists a edge else , $A_{ij} = 0$

6.3 Transpose, addition and norm

Block matrix Transpose

$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^T = \begin{bmatrix} A^T & C^T \\ B^T & D^T \end{bmatrix}$

Symmetric matrix: $A = A^T$

Properties of matrix addition

•Commutativity: $A + B = B + A$
•Associativity: $(A + B) + C = A + (B + C)$
•Addition with zero matrix: $A + 0 = 0 + A = A$
•Transpose of sum: $(A + B)^T = A^T + B^T$
If A is a matrix and β , γ are scalars $(\beta + \gamma)A = \beta A + \gamma A, (\beta\gamma)A = \beta(\gamma A)$

Matrix norm $||A|| = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{m} A_{ij}^2}$ matrix norm satisfies the properties of any norm

6.4 Matrix-vector multiplication

A is an $m \times n$ matrix and x is an n-vector, then the matrix-vector product $y = Ax$
$y_i = \sum_{k=1}^{n} A_{ik} x_k = A_{i1} x_1 + ... + A_{in} x_n$ for $i = 1...m$

•Row and column interpretations.

$y = Ax$ can be expressed as $y_i = b_i^T x, i = 1, .., m$ where $b_1^T, ..., b_m^T$ are rows of A
•$y = Ax$ could also be expressd in terms of column $y = x_1 a_1 + x_2 a_2 + ... + x_n a_n$

6.5 Complexity

addition: mn
sparse matrix addition: If A or B or both are sparse $min\{nnz(A), nnz(B)\}$
vector multiplication A_{mxn} with n-vector: $m(2n - 1) \approx 2mn$
Matrix Transpose: 0 flops

7 Matrix examples

7.1 Geometric transformations

•*Scaling*: $y = Ax$ with $A = aI$ stretches a vector by the factor $|a|$ (or shrinks it when $|a| < 1$), and it flips the vector (reverses its direction) if $a < 0$
•*Dilation*: $y = Dx$, where D is a diagonal matrix, $D = diag(d1, d2)$. Stretches the vector x by different factors along the two different axes. (Or shrinks, if $|d_i| < 1$, and flips, if $di < 0$.)
•*Rotation Matrix (counter clockwise)*: $y = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} x$
•*Reflection* Suppose that y is the vector obtained by reflecting x through the line that passes through the origin, inclined θ radians with respect to horizontal.

$y = \begin{bmatrix} cos(2\theta) & sin(2\theta) \\ sin(2\theta) & -cos(2\theta) \end{bmatrix} x$

• *Projection into a line* Projection of point x onto a set is the point in the set that is closest to x.

$y = \begin{bmatrix} (1/2)(1 + cos(2\theta)) & (1/2)sin(2\theta) \\ (1/2)sin(2\theta) & (1/2)(1 - cos(2\theta)) \end{bmatrix} x$

7.2 Selectors

An $m \times n$ selector matrix A is one in which each row is a unit vector (transposed):

$$\begin{bmatrix} e_{k_1}^T \\ . \\ . \\ e_{k_m}^T \end{bmatrix}$$

When it multiplies a vector, it simply copies the k_ith entry of x into the ith entry of $y = Ax$:

$y = (x_{k_1}, x_{k_2}, ..., x_{k_m})$

r:s matrix slicing

$A = [0_{m\times(r-1)} I_{m\times m} 0_{m\times(n-s)}]$

where $m = s - r + 1$

7.3 Incidence matrix

Directed graph: A *directed graph* consists of a set of *vertices* (or nodes), labeled 1,...,n, and a set of *directed edges* (or branches), labeled 1,...,m.

$A_{ij} = \begin{cases} 1, & \text{edge j points to node i} \\ -1, & \text{edge j points from node i} \\ 0, & \text{otherwise} \end{cases}$

7.4 Convolution

The convolution of an n-vector a and an m-vector b is the (n + m - 1)-vector denoted c = a * b

$c_k = \sum_{i+j=k+1} a_i b_j, k = 1, .., n + m - 1$

Properties of convolution

• symmetric: $a * b = b * a$

• associative: $(a * b) * c = a * (b * c)$

• $a * b = 0$ implies that either $a = 0$ or $b = 0$

• A basic property is that for fixed a, the convolution a * b is a linear function of b; and for fixed b, it is a linear function of a, $a * b = T(b)a = T(a)b$ where where T(b) is the (n + m - 1) × n matrix with entries

$T(b)_{ij} = \begin{cases} b_{i-j+1}, & 1 \le i - j + 1 \le m \\ 0, & \text{otherwise} \end{cases}$

Complexity of convolution

• $c = a * b$: 2mn flops

• $T(a) or T(b)a$: 2mn flops

• Convolution could be calculated faster using *fast Fourier transform (FFT)* : $5(m + n)log_2(m + n)flops$

8 Linear equations

8.1 Linear and affine functions

• Superposition condition: $f(\alpha x + \beta y) = \alpha f(x) + \beta f(y)$

• Such an f is called Linear

Matrix vector product function:

• A is $m \times n$ matrix such that $f(x) = Ax$

• f is linear: $f(\alpha x + \beta y) = A(\alpha x + \beta y) = \alpha f(x) + \beta f(y)$

• Converse is true: If $f : R^n \mapsto R^m$ is linear, then

$f(x) = f(x_1 e_1 + x_2 e_2 + ... x_n e_n)$

$= x_1 f(e_1) + x_2 f(e_2) + ... x_n f(e_n) = Ax$ with

$A = [f(e_1) + f(e_2) + ... f(e_n)]$

Affine Functions: $f : R^n \mapsto R^m$ is affine if it is a linear function plus a constant i.e $f(x) = Ax + b$ same as $f(\alpha x + \beta y) = \alpha f(x) + \beta f(y)$ holds for all x, y and α, β such that $\alpha + \beta = 1$

A and b can be calculated as

$A = [f(e_1) - f(0) \ \ f(e_2) - f(0) ... f(e_n) - f(0)]]$;

$b = f(0)$

• Affine functions sometimes incorrectly called linear functions

8.2 Linear function models

Price elasticity of demand $\delta_i^{price} = (p_i^{new} - p_i)/p_i$: fractional changes in prices $\delta_i^{dem} = (d_i^{new} - d_i)/d_i$: fractional change in demand Price demand elasticity model: $\delta^{dem} = E\delta^{price}$

Taylor series approximation

• The (first-order) Taylor approximation of f near (or at) the point z:

$\hat{f}(x) = f(z) + \frac{\partial f}{\partial x_1}(z)(x_1 - z_1) + ... + \frac{\partial f}{\partial x_n}(z)(x_n - z_n)$

• in compact notation:

$\hat{f}(x) = f(z) + Df(z)(x - z)$

8.3 Systems of linear equations

• set (or system) of m linear equations in n variables $x_1, .., x_n$:

$A_{11}x_1 + A_{12}x_2 + ... + A_{1n}x_n = b_1$

$A_{21}x_1 + A_{22}x_2 + ... + A_{2n}x_n = b_2$

.

.

$A_{m1}x_1 + A_{m2}x_2 + ... + A_{mn}x_n = b_m$

• systems of linear equations classified as

– under-determined if m < n (A wide)

– square if m = n (A square)

– over-determined if m > n (A tall)

9 Linear dynamical systems

9.1 Linear dynamical systems

$x_{t+1} = A_t x_t$, t = 1,2,. . .

• A_t are n × n dynamics matrices

• $(A_t)_{ij}(x_t)_j is contribution to (x_{t+1})_i from (x_t)_j$

• system is called time-invariant if $A_t = A$ doesn't depend on time

• can simulate evolution of xt using recursion $x_{t+1} = A_t x$

• linear dynamical system with input

$x_{t+1} = A_t x_t + B_t u_t + c_t$, t = 1,2,...

– u_t is an input m-vector

– B_t is n × m input matrix

– c_t is offset

K-Markov model:

$x_{t+1} = A_1 x_t + ... + A_K x_{t-K+1}$, t = K,K + 1,...

– next state depends on current state and K - 1 previous states

– also known as auto-regresssive model

– for K = 1, this is the standard linear dynamical system $x_{t+1} = Ax_t$

9.2 Population dynamics

9.3 Epidemic dynamics

9.4 Motion of a mass

9.5 Supply chain dynamics