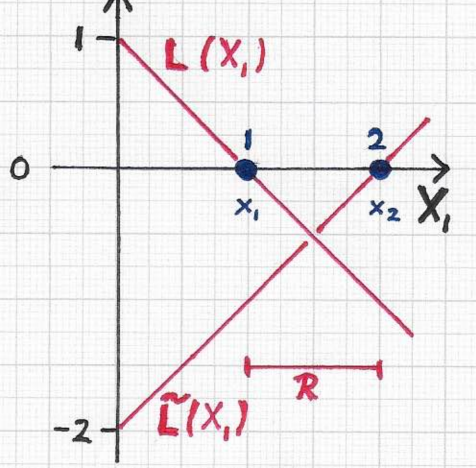
OBJECT AND MATERIAL EIGENFUNCTIONS - Cont'd.

• Laplacian eigenfunctions and neural networks:... The most important and most difficult aspect of the linear programming (and optimization) problem we must solve in the general multivariate setting is the CORRECT DEFINITION OF THE COST FUNCTION TO BE MINIMIZED, TOGETHER WITH A/THE SUFFICIENT SET OF EQUALITY AND INEQUALITY CONSTRAINTS AND CERTAIN, A-PRIORI-SPECIFIED VALUES OF PARAMETERS (to restrict the solution space in an allowable and desirable way). Again, we first consider the simple univariate problem studied before to fully explore this aspect. We use these definitions:



• $L(X_1) = c_0 + c_1 X_1$, $\tilde{L}(X_1) = \tilde{c}_0 + \tilde{c}_1 X_1$;

• cost function S:

$$S = (L(1)/L)^2 + (L(2)/L)^2 + (\tilde{L}(1)/\tilde{L})^2 + (\tilde{L}(2)/\tilde{L})^2$$
$$= ((L(1))^2 + (L(2))^2)/\nabla^2 + ((\tilde{L}(1))^2 + (\tilde{L}(2))^2)/\tilde{\nabla}^2$$
$$= (L(1))^2 + (L(2))^2 + (\tilde{L}(1))^2 + (\tilde{L}(2))^2$$

{since $\nabla^2 = 1$ for $c_1 = +/-1$ and $\tilde{\nabla}^2 = 1$ for $\tilde{c}_1 = +/-1$ }

$$= (c_0 + c_1)^2 + (c_0 + 2c_1)^2 + (\tilde{c}_0 + \tilde{c}_1)^2 + (\tilde{c}_0 + 2\tilde{c}_1)^2 \rightarrow \min ;$$

• inequalities: { $c_1 = -1$, $\tilde{c}_1 = 1$ specified a priori }

$L(1) = c_0 - 1$, $L(2) = c_0 - 2$, $\tilde{L}(1) = \tilde{c}_0 + 1$, $\tilde{L}(2) = \tilde{c}_0 + 2$

$\Rightarrow \quad c_0 \leq 1 \ \wedge \ c_0 \leq 2 \ \wedge \ \tilde{c}_0 \leq -1 \ \wedge \ \tilde{c}_0 \leq -2$

• equalities / specified values: $c_1^2 = 1 \wedge \tilde{c}_1^2 = 1 \wedge c_1 = -\tilde{c}_1$

• result: $(c_0, c_1, \tilde{c}_0, \tilde{c}_1) = (1, -1, -2, 1)$.

...

OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

- Laplacian eigenfunctions and neural networks:... We can and must view and describe this simple example in the general multivariate context. For this purpose, we use a needed multi-index notation. We discuss the example and **make** references to the necessary multivariate generalizations:

1) We must construct **2** (any number of) univariate (multivariate) linear polynomials L_1 and L_2 $(L_1, L_2, \ldots)$, using a 1-dimensional (N-dimensional) domain X_1 $((X_1, X_2, \ldots, X_N))$:

$$L_1(X_1) = c_0^1 + c_1^1 X_1 \quad , \quad L_2(X_2) = c_0^2 + c_1^2 X_1$$

$$\left(L_i(X_1, \ldots, X_N) = c_0^i + c_1^i X_1 + \ldots + c_N^i X_N \quad , \quad i = 1 \ldots K \right).$$

2) The linear polynomials must have values that are negative or **zero** for **2** (n) given X_1-values (locations/sites/points $\mathbf{x}_j = (X_1^j, \ldots, X_N^j)^T$, $j = 1 \ldots n$):

$$L_1(1) = c_0^1 + c_1^1 \leq 0 \quad \wedge \quad L_1(2) = c_0^1 + c_1^1 2 \leq 0$$

$$\wedge \; L_2(1) = c_0^2 + c_1^2 \leq 0 \quad \wedge \quad L_2(2) = c_0^2 + c_1^2 2 \leq 0$$

$$\left(L_i(\mathbf{x}_j) = c_0^i + c_1^i X_1^j + \ldots + c_N^i X_N^j = c_0^i + \sum_{d=1}^{N} c_d^i X_d^j \leq 0, \quad i = 1 \ldots K, j = 1 \ldots n \right).$$

3) The square of the first derivative (the length of the gradient vector) of all polynomials must be **1**:

$$(c_1^1)^2 = 1 \quad \wedge \quad (c_1^2)^2 = 1$$

$$\left(\langle \nabla L_i, \nabla L_i \rangle = (c_1^i)^2 + \ldots + (c_N^i)^2 = 1 \;, \; i = 1 \ldots K \right).$$

4) The first derivative values of the linear polynomials must have different signs (GENERALIZATION?):

$$c_1^1 = -c_1^2 .$$

•••

OBJECT AND MATERIAL EIGENFUNCTIONS - Cont'd.

• Laplacian eigenfunctions and neural networks: ... **5)** The cost function to be minimized for the determination of all coefficients is the sum of all squared values of all polynomials' values when evaluated at all X_1-values (locations/sites/points x_j, $j=1...n$):

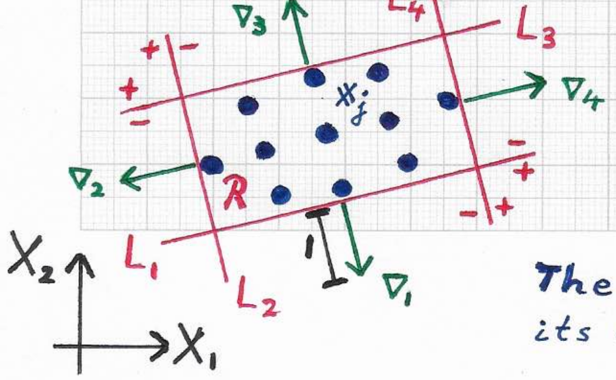
$$S = (c_0^1 + c_1^1)^2 + (c_0^1 + 2c_1^1)^2 + (c_0^2 + c_1^2)^2 + (c_0^2 + 2c_1^2)^2 \rightarrow min$$

$$\left(S = \sum_{i=1}^{K} \sum_{j=1}^{n} (L_i(x_j))^2 \rightarrow min \right).$$

6) Result: $(c_0^1, c_1^1, c_0^2, c_1^2) = (-2, 1, 1, -1)$ $\left((c_0^1, c_1^1, ..., c_N^K) \right)$.

$$\boxed{\begin{array}{l} L_i(X) = c_0^i + \sum_{d=1}^{N} c_d^i X_d \\ L_i(x_j) \leq 0 \\ \langle \nabla L_i, \nabla L_i \rangle = 1 \\ \text{Condition(s) for } \{\nabla L_i\} \\ \sum_{i,j} (L_i(x_j))^2 \rightarrow min \end{array}}$$

The relevant facts of the linear programming/optimization problem are summarized in the left box. The needed generalized condition(s) for the unit-length gradient vectors of the linear polynomials L_i is (are) not immediately obvious. The general set $\{\nabla L_i\}_{i=1}^{K}$ must be carefully defined.



We consider the bivariate case next, to describe desirable computational (and geometrical) generalizations of the univariate case.

The left figure shows a given set $\{x_j\}$, its associated region R, with lines and gradients.

...

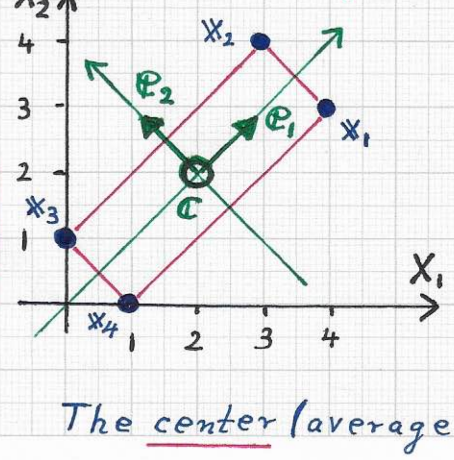
OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks: ... The simple bivariate example illustrated on the previous page makes several important points. The **region R** is bounded by a rectangle that has a data-defined orientation, defined by the given set $\{x_j\}_{j=1}^n$. The four **rectangle** boundary edges are implied by the four linear, bivariate polynomials — that all have negative values (or have value zero) in R and have normalized, unit gradient vectors "pointing away from R, i.e., indicating maximal increase directions. In summary:

$$L_i(\mathbf{x}) = L_i(X_1, X_2) = c_0^i + c_1^i X_1 + c_2^i X_2 \ , \ i=1\ldots4 \ ;$$
$$\nabla_i = \nabla L_i(X_1, X_2) = (c_1^i, c_2^i)^T, \ \langle \nabla_i, \nabla_i \rangle = 1, \ i=1\ldots4 ;$$
$$L_i(x_j) = c_0^i + c_1^i X_1^j + c_2^i X_2^j \le 0 \ , \ x_j = (X_1^j, X_2^j), \ i=1\ldots4, \ j=1\ldots n ;$$

GENERALIZATION of univariate case: $\nabla_3 = -\nabla_1$, $\nabla_4 = -\nabla_2$

"COST FUNCTION": compute minimal region R that contains all points x_j, i.e., the smallest rectangle region boundary with orientation vectors ∇_1, ∇_2.



The left figure shows a simple example. The points $x_1 = (4,3)^T, \ldots, x_4 = (1,0)^T$ are given. We understand "data-defined orientation" as "principal directions" (principal components) of the set $\{x_j\}$, as defined by its covariance matrix. The center (average) of the point set is $\mathbf{c} = (2,2)^T$, and is subtracted from the coordinates of all given points x_j.

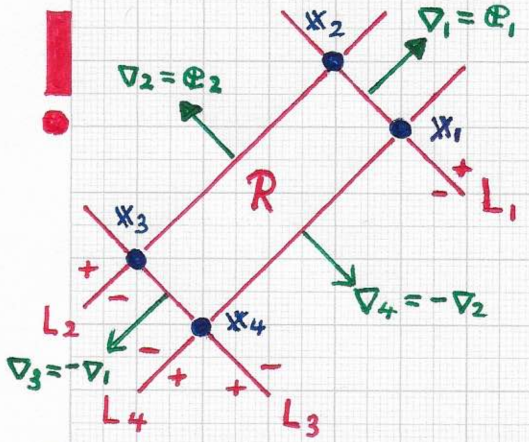
...

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

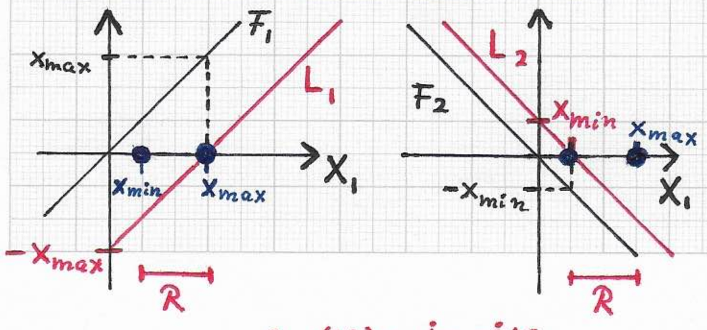
• Laplacian eigenfunctions and neural networks:... After this so-called "mean subtraction" step, one obtains the following point matrix P, where points are written as columns:

$$P = \begin{bmatrix} 2 & 1 & -2 & -1 \\ 1 & 2 & -1 & -2 \end{bmatrix} \Rightarrow P \cdot P^T = \begin{bmatrix} 2 & 1 & -2 & -1 \\ 1 & 2 & -1 & -2 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 1 & 2 \\ -2 & -1 \\ -1 & -2 \end{bmatrix} = \begin{bmatrix} 10 & 8 \\ 8 & 10 \end{bmatrix} = C .$$

The matrix C (covariance) has eigenvalues defined by its characteristic polynomial, i.e., $(10-\lambda)^2 - 64 = 0$. The resulting two eigenvalues are $\lambda_1 = 18$ and $\lambda_2 = 2$, having the associated normalized, unit eigenvectors $\mathbb{e}_1 = (\sqrt{2}/2, \sqrt{2}/2)^T$ and $\mathbb{e}_2 = (-\sqrt{2}/2, \sqrt{2}/2)^T$, respectively. The eigenvectors are included in the figure on the previous page. These eigenvectors define the needed four gradients/gradient vectors of the four linear polynomials L_i as shown in the left figure. The requirement that R must have minimal area and must include all points $\mathbb{x}_j$ implies the the values of the polynomial constants $c_0^1, c_0^2, c_0^3, c_0^4$.



The left figures show how to determine the values of these constants in the univariate case. Here, we define $x_{min} = \min\{x_j\}$, $x_{max} = \max\{x_j\}$, $F_1 = 0 + 1X_1$ and $F_2 = 0 - 1X_1$. Thus, $L_1 = -x_{max} + 1X_1$ and $L_2 = x_{min} - 1X_1$. ...



$$L_i(X_1) = c_0^i + c_1^i X_1$$