Classification Models

Goran Banjac

Large-Scale Convex Optimization
ETH Zurich

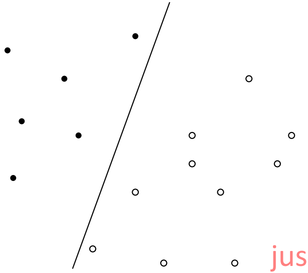
May 19, 2020

Linear discrimination

- separate two sets of points $\{x_1, \ldots, x_I\}$, $\{z_1, \ldots, z_J\}$ by a hyperplane:

$$a^T x_i + b > 0, \quad i = 1, \ldots, I, \qquad a^T z_j + b < 0, \quad j = 1, \ldots, J$$

if there exits, (separable), it is not unique.



- homogeneous in (a, b), hence equivalent to

just scaling the coefficients a and b.

$$a^T x_i + b \geq 1, \quad i = 1, \ldots, I, \qquad a^T z_j + b \leq -1, \quad j = 1, \ldots, J$$

- a set of linear inequalities in (a, b)

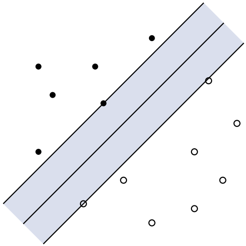
linear programming problems actually...

Robust linear discrimination

- Euclidean distance between hyperplanes

$$\mathcal{H}_1 = \{x \mid a^T x + b = 1\}$$
$$\mathcal{H}_2 = \{x \mid a^T x + b = -1\}$$

is $\mathrm{dist}(\mathcal{H}_1, \mathcal{H}_2) = 2/\|a\|_2$



- separate two sets of points by maximum margin by solving

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|a\|_2 \\ \text{subject to} & a^T x_i + b \geq 1, \quad i = 1, \ldots, I \\ & a^T z_j + b \leq -1, \quad j = 1, \ldots, J \end{array}$$

support vector machines...

Dual problem

- Lagrange dual of the maximum margin separation problem:

$$\boxed{\begin{array}{ll} \text{maximize} & \mathbf{1}^T\mu + \mathbf{1}^T\nu \\ \text{subject to} & 2\left\| \sum_{i=1}^I \mu_i x_i - \sum_{j=1}^J \nu_j z_j \right\|_2 \leq 1 \\ & \mathbf{1}^T\mu = \mathbf{1}^T\nu, \quad \mu \succeq 0, \quad \nu \succeq 0 \end{array}}$$

- from duality, optimal value is inverse of maximum margin of separation
- change variables to $\theta_i = \mu_i/\mathbf{1}^T\mu$, $\eta_j = \nu_j/\mathbf{1}^T\nu$, $t = 1/(\mathbf{1}^T\mu + \mathbf{1}^T\nu)$
- invert objective to minimize $1/(\mathbf{1}^T\mu + \mathbf{1}^T\nu) = t$

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \left\| \sum_{i=1}^I \theta_i x_i - \sum_{j=1}^J \eta_j z_j \right\|_2 \leq t \\ & \theta \succeq 0, \quad \mathbf{1}^T\theta = 1, \quad \eta \succeq 0, \quad \mathbf{1}^T\eta = 1 \end{array}$$

- optimal value is distance between convex hulls

Labeled data

- assigning a label to each point, we can represent data points as (x_i, y_i) where $y_i \in \{-1, 1\}$
  - $y_i = -1$: $a^T x_i + b \geq 1$
  - $y_i = 1$: $a^T x_i + b \leq -1$
- this allows us to rewrite both constraints as

$$\boxed{y_i(a^T x_i + b) \leq -1}$$

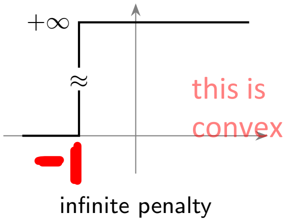
- the linear discrimination problem can then be written as

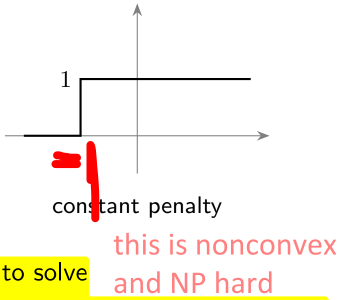
$$\text{minimize} \quad \sum_{i=1}^{N} \mathcal{I}_{[-\infty,-1]}\left(y_i(a^T x_i + b)\right)$$

this is convex; but if it is non-separable problem, then it is infeasible.

Non-separable sets

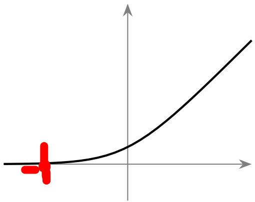
- if the points with different labels are not linearly separable, then the optimization problem becomes infeasible
- a natural extension would be to find a hyperplane that minimizes the number of misclassified points





- unfortunately, such problem is very hard to solve
- instead, we use convex loss functions that approximately minimize the number of misclassified points

Logistic regression



- logistic regression uses the *logistic* loss function

$$l(u) = \log(1 + e^u)$$

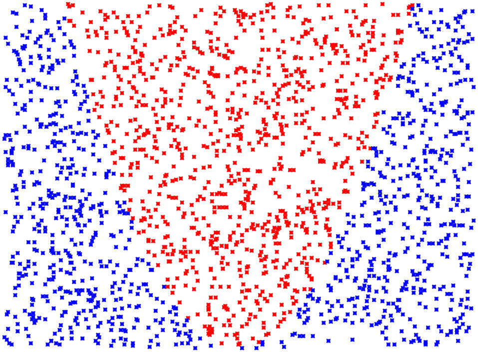
- training problem:

$$\text{minimize} \quad \sum_{i=1}^{N} \log\left(1 + e^{y_i(a^T x_i + b)}\right)$$

- convex in (a, b)
- problem formulation is slightly different when $y_i \in \{0, 1\}$

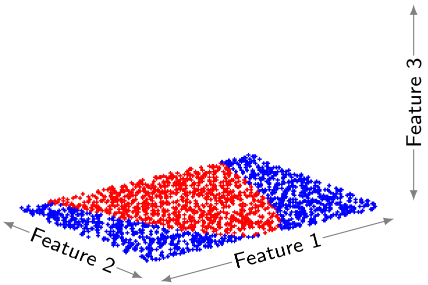
Nonlinear example

- logistic regression tries to separate data by a hyperplane
- introducing nonlinear features, we can approximate a nonlinear boundary with logistic regression



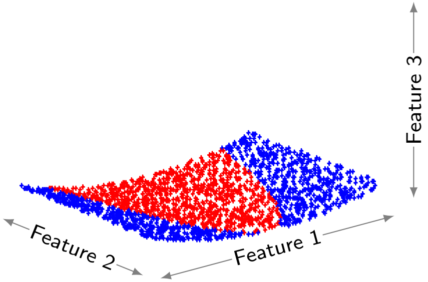
Nonlinear example

- the boundary seems to be linear in feature 2 and quadratic in feature 1
- add a third feature which is feature 1 squared



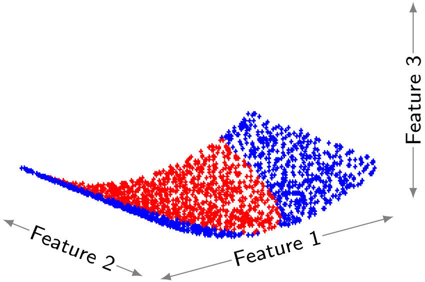
Nonlinear example

- the boundary seems to be linear in feature 2 and quadratic in feature 1
- add a third feature which is feature 1 squared



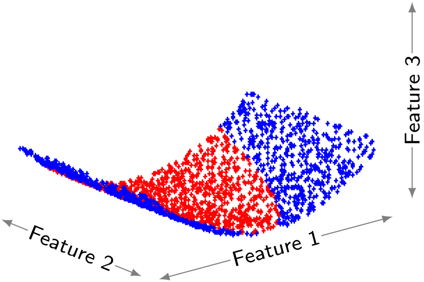
Nonlinear example

- the boundary seems to be linear in feature 2 and quadratic in feature 1
- add a third feature which is feature 1 squared



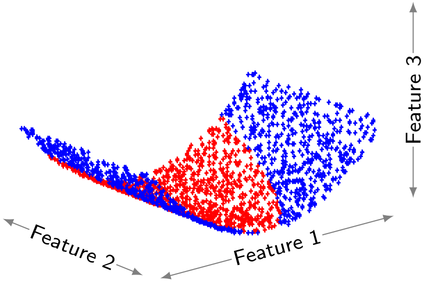
Nonlinear example

- the boundary seems to be linear in feature 2 and quadratic in feature 1
- add a third feature which is feature 1 squared



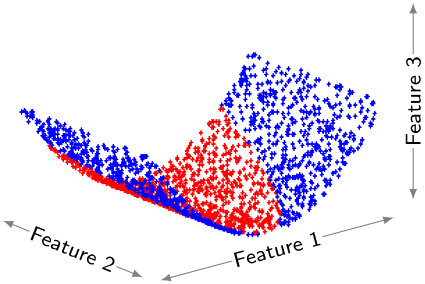
Nonlinear example

- the boundary seems to be linear in feature 2 and quadratic in feature 1
- add a third feature which is feature 1 squared



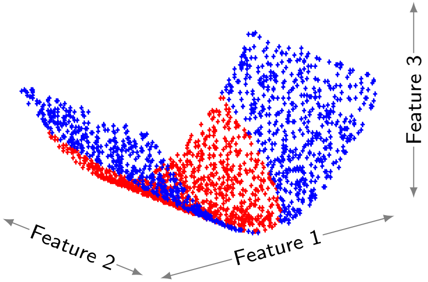
Nonlinear example

- the boundary seems to be linear in feature 2 and quadratic in feature 1
- add a third feature which is feature 1 squared



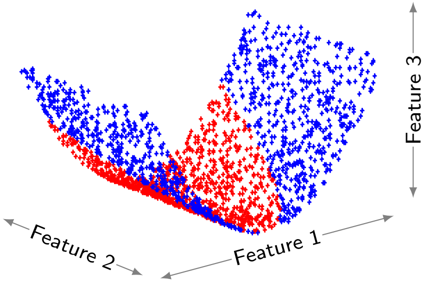
Nonlinear example

- the boundary seems to be linear in feature 2 and quadratic in feature 1
- add a third feature which is feature 1 squared



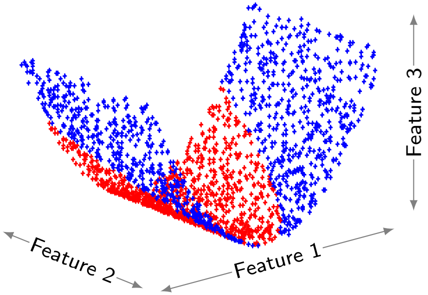
Nonlinear example

- the boundary seems to be linear in feature 2 and quadratic in feature 1
- add a third feature which is feature 1 squared



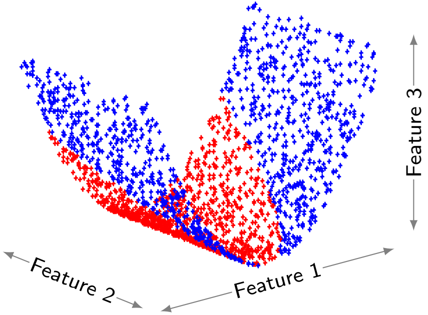
Nonlinear example

- the boundary seems to be linear in feature 2 and quadratic in feature 1
- add a third feature which is feature 1 squared



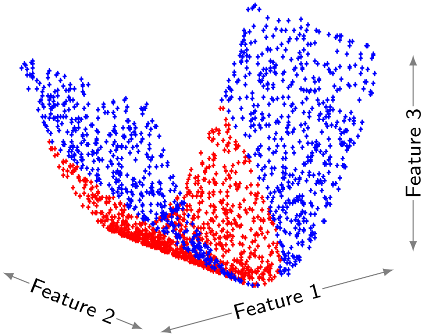
Nonlinear example

- the boundary seems to be linear in feature 2 and quadratic in feature 1
- add a third feature which is feature 1 squared



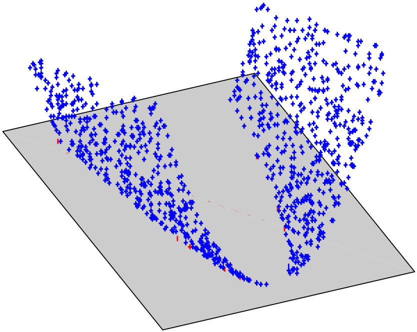
Nonlinear example

- the boundary seems to be linear in feature 2 and quadratic in feature 1
- add a third feature which is feature 1 squared



Nonlinear example

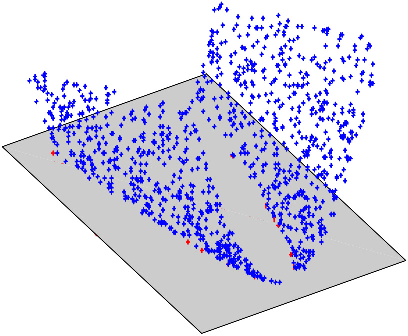
- the boundary seems to be linear in feature 2 and quadratic in feature 1
- add a third feature which is feature 1 squared



- data linearly separable in lifted (feature) space

Nonlinear example

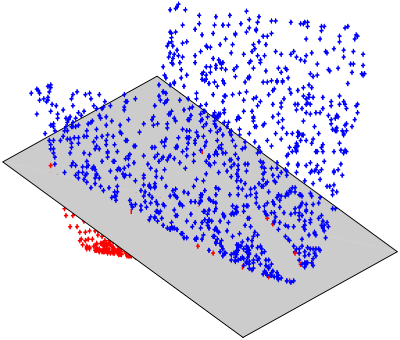
- the boundary seems to be linear in feature 2 and quadratic in feature 1
- add a third feature which is feature 1 squared



- data linearly separable in lifted (feature) space

Nonlinear example

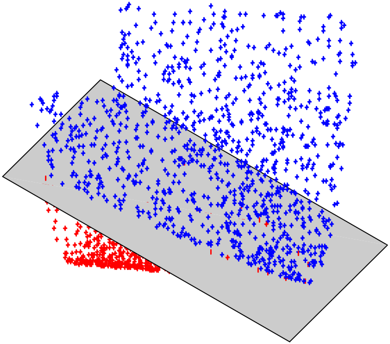
- the boundary seems to be linear in feature 2 and quadratic in feature 1
- add a third feature which is feature 1 squared



- data linearly separable in lifted (feature) space

Nonlinear example

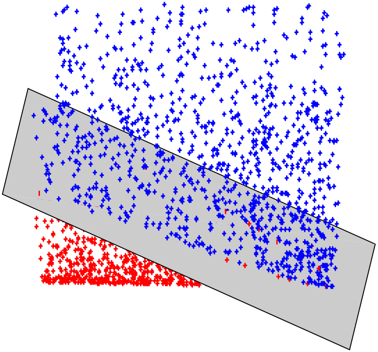
- the boundary seems to be linear in feature 2 and quadratic in feature 1
- add a third feature which is feature 1 squared



- data linearly separable in lifted (feature) space

Nonlinear example

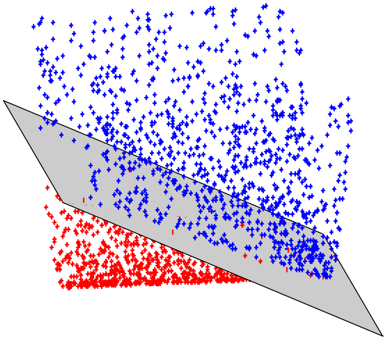
- the boundary seems to be linear in feature 2 and quadratic in feature 1
- add a third feature which is feature 1 squared



- data linearly separable in lifted (feature) space

Nonlinear example

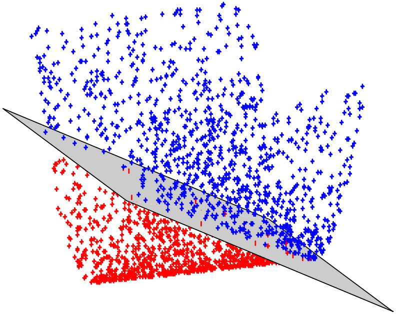
- the boundary seems to be linear in feature 2 and quadratic in feature 1
- add a third feature which is feature 1 squared



- data linearly separable in lifted (feature) space

Nonlinear example

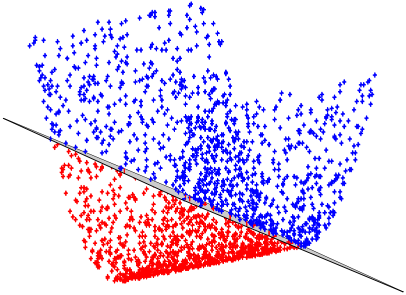
- the boundary seems to be linear in feature 2 and quadratic in feature 1
- add a third feature which is feature 1 squared



- data linearly separable in lifted (feature) space

Nonlinear example

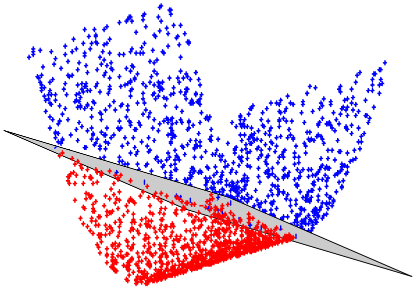
- the boundary seems to be linear in feature 2 and quadratic in feature 1
- add a third feature which is feature 1 squared



- data linearly separable in lifted (feature) space

Nonlinear example

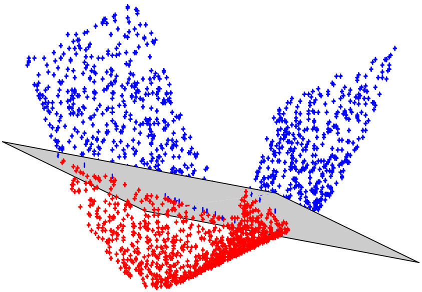
- the boundary seems to be linear in feature 2 and quadratic in feature 1
- add a third feature which is feature 1 squared



- data linearly separable in lifted (feature) space

Nonlinear example

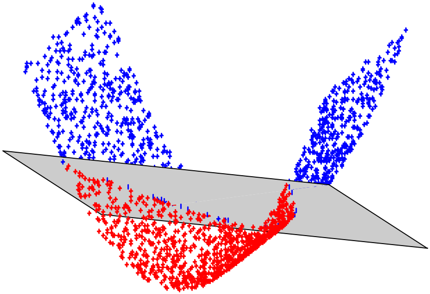
- the boundary seems to be linear in feature 2 and quadratic in feature 1
- add a third feature which is feature 1 squared



- data linearly separable in lifted (feature) space

Nonlinear example

- the boundary seems to be linear in feature 2 and quadratic in feature 1
- add a third feature which is feature 1 squared

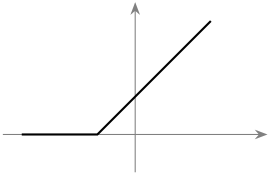


- data linearly separable in lifted (feature) space

Nonlinear models

- create feature map $\phi\colon \mathbb{R}^n \to \mathbb{R}^p$ of training data
- data points $x_i \in \mathbb{R}^n$ replaced by featured data points $\phi(x) \in \mathbb{R}^p$
- use regularization (*e.g.*, Tikhonov) to avoid overfitting
- regularize only a and not the bias term b
- hyperparameters are usually selected using cross validation
- after training a model, we predict the label for a new data point x_i:
  - if $a^T\phi(x_i) + b > 0$, then $y_i = -1$
  - if $a^T\phi(x_i) + b < 0$, then $y_i = 1$
  - if $a^T\phi(x_i) + b = 0$, then either label
- the set $\{x \mid a^T\phi(x) + b = 0\}$ is called the *decision boundary*

Support vector machines



- *SVM* uses the *hinge* loss function

$$l(u) = \max(0, 1 + u)$$

- training problem:

$$\text{minimize} \quad \sum_{i=1}^{N} \max\left(0, 1 + y_i(a^T\phi(x_i) + b)\right)$$

- convex in (a, b)
- zero cost for sample i if $y_i(a^T\phi(x_i) + b) \leq -1$

Dual problem

- consider Tikhonov regularized SVM:

$$\text{minimize} \quad \sum_{i=1}^{N} \max\big(0, 1 + y_i(a^T\phi(x_i) + b)\big) + \frac{\lambda}{2}\|a\|_2^2$$

- derive the dual from reformulation of SVM:

$$\text{minimize} \quad \mathbf{1}^T \max\left(0, 1 + (X_{\phi,Y}a + Yb)\right) + \frac{\lambda}{2}\|a\|_2^2$$

where $\max$ is vector-valued and

$$X_{\phi,Y} = \begin{bmatrix} y_1\phi(x_1)^T \\ \vdots \\ y_N\phi(x_N)^T \end{bmatrix}, \qquad Y = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}$$

Dual problem

- let $L = [X_{\phi,Y}, Y]$ and write problem as

$$\text{minimize} \quad \underbrace{\mathbf{1}^T \max\left(0, 1 + (X_{\phi,Y} a + Y b)\right)}_{f(L(a,b))} + \underbrace{\tfrac{\lambda}{2}\|a\|_2^2}_{g(a,b)}$$

  where
  - $f(w) = \sum_{i=1}^N f_i(w_i)$ and $f_i(w_i) = \max(0, 1 + w_i)$ (hinge loss)
  - $g(a,b) = \frac{\lambda}{2}\|a\|_2^2$, *i.e.*, it does not depend on b

- dual problem:

$$\text{minimize} \quad f^*(\nu) + g^*(-L^T \nu)$$

Conjugate of f

- conjugate of $f_i(w_i) = \max(0, 1 + w_i)$ (hinge loss):

$$f_i^*(\nu_i) = \begin{cases} -\nu_i & 0 \le \nu_i \le 1 \\ +\infty & \text{otherwise} \end{cases}$$

- conjugate of $f(w) = \sum_{i=1}^N f_i(w_i)$ is the sum of individual conjugates:

$$f^*(\nu) = \sum_{i=1}^N f_i^*(\nu_i) = -\mathbf{1}^T \nu + \mathcal{I}_{[0,1]}(\nu)$$

Conjugate of g

- conjugate of $g(a,b) = \frac{\lambda}{2}\|a\|_2^2 = g_1(a) + g_2(b)$ is

$$g^*(\mu_a, \mu_b) = g_1^*(\mu_a) + g_2^*(\mu_b) = \frac{1}{2\lambda}\|\mu_a\|_2^2 + \mathcal{I}_{\{0\}}(\mu_b)$$

- evaluated at $-L^T\nu = -[X_{\phi,Y}, Y]^T\nu$:

$$\begin{aligned} g^*(-L^T\nu) &= g^*\left(-\begin{bmatrix} X_{\phi,Y}^T \\ Y^T \end{bmatrix}\nu\right) \\ &= \frac{1}{2\lambda}\|-X_{\phi,Y}^T\nu\|_2^2 + \mathcal{I}_{\{0\}}(-Y^T\nu) \\ &= \frac{1}{2\lambda}\nu^T X_{\phi,Y} X_{\phi,Y}^T\nu + \mathcal{I}_{\{0\}}(-Y^T\nu) \end{aligned}$$

SVM dual

- the SVM dual is

$$\text{minimize} \quad f^*(\nu) + g^*(-L^T\nu)$$

- inserting the above computed conjugates gives the dual problem

$$\begin{array}{ll} \text{minimize} & -\mathbf{1}^T\nu + \frac{1}{2\lambda}\nu^T \boxed{X_{\phi,Y}X_{\phi,Y}^T}\nu \\ \text{subject to} & 0 \le \nu \le 1 \\ & Y^T\nu = 0 \end{array}$$

- since $Y \in \mathbb{R}^N$, $Y^T\nu = 0$ is a hyperplane constraint
- if no bias term b, then the same dual but with no hyperplane constraint

Recovering primal solution

- meaningless to solve dual if we cannot recover primal
- necessary and sufficient primal-dual optimality conditions

$$0 \in \begin{cases} \partial f^*(\nu) - L(a,b) \\ \partial g^*(-L^T\nu) - (a,b) \end{cases}$$

- from dual solution ν, find (a,b) that satisfies both of the above
- for SVM, second condition is

$$\partial g^*(-L^T\nu) = \begin{bmatrix} \frac{1}{\lambda}(-X_{\phi,Y}^T\nu) \\ \partial \mathcal{I}_{\{0\}}(-Y^T\nu) \end{bmatrix} \ni \begin{bmatrix} a \\ b \end{bmatrix}$$

  which gives optimal $\boxed{a = -\frac{1}{\lambda}X_{\phi,Y}^T\nu}$ (since unique)
- cannot recover b from this condition

Recovering primal solution

- necessary and sufficient primal-dual optimality conditions

$$0 \in \begin{cases} \partial f^*(\nu) - L(a,b) \\ \partial g^*(-L^T\nu) - (a,b) \end{cases}$$

- for SVM, row i of first condition is $0 \in \partial f^*(\nu_i) - L_i(a,b)$, where

$$\partial f_i^*(\nu_i) = \begin{cases} [-\infty, -1] & \nu_i = 0 \\ -1 & 0 < \nu_i < 1 \\ [-1, +\infty] & \nu_i = 1 \end{cases}, \qquad L_i = y_i \begin{bmatrix} \phi(x_i)^T & 1 \end{bmatrix}$$

- pick i such that $\nu_i \in (0,1)$, then $\partial f_i^*(\nu_i) = -1$ is unique and

$$0 = \partial f_i^*(\nu_i) - L_i(a,b) = -1 - y_i(a^T\phi(x) + b)$$

and the optimal b must satisfy $\boxed{b = -y_i - a^T\phi(x_i)}$ for such i

References

- these lecture notes are based to a large extent on the following material:
  - Stanford EE364a class developed by Stephen Boyd
  - Lund course on Optimization for Learning developed by Pontus Giselsson

- the original slides can be downloaded from

  `https://web.stanford.edu/class/ee364a/lectures.html`

  `http://www.control.lth.se/education/engineering-program/frtn50-optimization-for-learning/`